CA Top Secret® for z/OS

Implementation: CICS Guide

r15

Sixth Edition

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 in this edition of the documentation:

- Installing CA Top Secret in CICS (see page 13)—Updated examples based on release information.
- Display CICSPROD Default Bypass and Protect Lists (see page 18)—Added CISP, CIS1, CJSL, CRST, and CPCT to the Bypass and Protect list information.
- Display CICSTEST Default Bypass and Protect Lists (see page 18)——Added CISP, CIS1, CJSL, CRST, and CPCT to the Bypass and Protect list information.
- Modes for LCF Checking (see page 48)—Consolidated information into this section and expanded the explanations in the examples.
- How to Track Execution of Transactions That Bypass Security Checking (see page 59)—Overhauled topic and added CISP, CIS1, CJSL, CRST, and CPCT to the TRANID Bypass list in the example.
- Bypass Transaction Security (see page 63)—Added CISP, CIS1, CJSL, CRST, and CPCT to the Bypass list information.

More information:

Signing on Using CESL (see page 94)
Change a Password or Password Phrase (see page 99)

Contents

Chapter 1: Defining CICS to CA Top Secret

This section contains the following topics:

Introduction

This guide describes:

- How to install the CA Top Secret security product in your CICS system
- How to secure a CICS inter-system environment, daily operations, customization, and diagnostics
- How to select and implement CICS security parameters and/or CA Top Secret suboptions to administer security in your environment

Migration Considerations

Consider the following:

- The DFHSIT parameter XUSER, new for CICS Release 4.1 and above controls non-terminal (background) security.
- You do not have to specify the sysid of the region in the Bypass List to deactivate security. Only SEC=NO is required.
- Installation check messages are not displayed at start up. Use TSEU=INSTALL to see the security parameter settings.
- TSEU=CESF=tttt will log off the designated terminal.
- The EXEC CICS ENABLE, DISABLE, and EXTRACT commands are protected by the SPI resource of EXITPROG.
- The PCTCMDSEC FACILITY suboption is part of the DFHSIT parameter overrides. Based on the Facilities Matrix setting, this parameter honors or overrides the DFHSIT parameter CMDSEC=.
- The PCTRESSEC FACILITY suboption is part of the DFHSIT parameter overrides. Based on the Facilities Matrix setting, this parameter honors or overrides the DFHSIT parameter RESSEC=.
- Consider the following pertaining to the MAXUSER FACILITY suboption:
  - The default setting of 3000 for MAXUSER is high unless you have a very large CICS region (over 2500 users). Therefore, you should adjust your MAXUSER size to match the expected high number of users that might be active in the CICS region.
  - In addition to calculating the number of users for the user pool allocation, MAXUSER is used for a feature called resource caching.

    This feature uses the MAXUSER setting to build the cache box pool.
- MRO securityname is not used for bind and link.
- PCT and PPT entries have been replaced by CSD entries.
- CICS 4.1 and above implements transaction security for background (non-terminal) transactions. You might need to define permits or optionally add to the tran or tranid bypass lists those transactions which are started in this manner.
- The Automatic Terminal Signon (ATS) feature is invoked during any resource validation.
- The real Port-of-Entry (POE) is used for consoles involved in Automatic Terminal Signon (ATS). (The CICS terminal ID was used previously.) If using source protection for your consoles, you might need to add the POE to your console source list(s). The Port-of-Entry name can be obtained from the CONSNAME parameter in the CICS TCT definition. It corresponds to the names defined for consoles in the MVS SYS1.PARMLIB member, CONSOLnn. (See the IBM *CICS System Definition Guide* for more information.)

- Locktime runs in pseudo-conversational mode when you specify PCLOCK=YES facility suboption. The TSLK transaction is used to perform this processing. The TSLK transaction must be defined to the CICS system. The default locktime processing is PCLOCK=NO (conversational mode).
- Certain transaction IDs and program names must be defined.

Installing CA Top Secret in CICS

The CA Top Secret CICS interface requires the CA Common Services for z/OS CAIENF product to be installed and activated. CAIENF CICS installs CA Top Secret intercepts and drives CA Top Secret CICS during security-related events. Without CAIENF, CA Top Secret CICS does not function. For CAIENF to operate properly, establish support for all the active CICS releases at your site by setting the CAIENF parameter file (ENFPARM).

Example: Intercept CICS Startup to Install the CA Security Interface

This example causes ENF to intercept CICS startup in CTS 3.1, CTS 3.2, CTS 4.1, CTS 4.2, CTS 5.1, and CTS 5.2 to install the CA security interface:

```
MODE(CICS,ON)
CICSREL(64,65,66,67,68,69)
```

64

Refers to CICS TS Release 3.1.

65

Refers to CICS TS Release 3.2.

66

Refers to CICS TS Release 4.1.

67

Refers to CICS TS Release 4.2.

68

Refers to CICS TS Release 5.1.

69

Refers to CICS TS Release 5.2

Note: For information about related parameters and ENF operation, see the *CA Common Services for z/OS Getting Started*.

After Installation

After CA Top Secret has been successfully installed:

- Set CA Top Secret control options for CICS security processing in the Facilities Matrix.
- Define the region control ACID for the CICS region and associate it with the appropriate MASTFAC parameter.
- Define CICS as a started task (STC) or a batch job in the CA Top Secret environment.

CICS Installation

To install CA Top Secret in your CICS system:

- Confirm that appropriate CICS components have been installed in your SMP/E environments for CA Top Secret and for CA Common Services for z/OS.
- Ensure that the ENF CICSREL parm includes initiation of support for the release of CICS in question.
- Ensure that the CAILOAD for CA Top Secret has been included in the system link list or in the STEPLIB of the CA Common Services ENF started task JCL.

  Failure to properly install and configure a CICS release in CA software often results in the absence of successful phase initiation messages for phase 0, phase 1, and phase 2 initiation messages at CICS start-up. Assure that these messages indicate successful interface initiation.
- Verify that the SDT has been initialized if you are using Record or Screen Level Protection (RLP/SLP).
- Any program defined in the CSD job displays in the chapter, “CSD PROGRAM and TRANSACTION Sample Entries” and must be in the DFHRPL library. This is normally assured by adding the CA Top Secret CAILOAD to DFHRPL.

  Whenever you apply an upgrade to CA Top Secret, update the affected modules for the programs defined in the CSD and in the appropriate library in the DFHRPL.

  Optional exit programs may be assembled and linked to customize certain CICS security operations. For information, see the section "CA Top Secret CICS Exits”.
- Set CICS security parameters in the CICS tables or define CA Top Secret FACILITY sub-options for controlling CICS security processing in the Facility Matrix table.
- Activate your CICS region. A series of CA Top Secret messages display indicating the phase of initiation for the region; to view the region's security parameters, issue transaction TSEU=INSTALL. A list of these messages appear in the chapter, “CICS Installation Checklist.”

Modify the PLTPI Table for the TSSCPLT Initialization Check Program (Optional)

You can optionally execute the TSSCPLT program during the PLTPI processing phase to ensure that the CICS interface security has been properly initialized in a CICS region.

Typically, initialization failure occurs because of:

- Incorrect or incomplete installation of the CICS interface
- Failure to start the CAIENF started procedure on your system

TSSCPLT verifies that:

- CA Top Secret is installed on the LPAR
- CA Top Secret CICS control blocks are initialized
- CA Top Secret Data Control Module (DCM) has been properly installed into the CAIENF database

If TSSCPLT detects that the CICS interface:

- Has successfully initialized, it issues informational message TSS6160
- Has not properly initialized, it issues message TSS6161 and abends the region with a user abend code, U1800

To check CICS region initialization processing

1. Define a new or modify an existing PLTPI table to include the TSSCPLT program. CAI.CAKOJCL0(TSSCPLT) contains the sample PLTPI table definition:.

   ```
   DFHPLTxx TITLE 'PLTPI-xx PLTPI TABLE'
   DFHPLT TYPE=INITIAL,SUFFIX=xx
   DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
   DFHPLT TYPE=ENTRY,PROGRAM=TSSCPLT
   DFHPLT TYPE=FINAL
   ```

   Notes:

   - The DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM entry is necessary to delimit PLTPI processing done between the first and second phases of PLTPI processing. The TSSCPLT program must execute in the second phase of PLTPI processing.
   - The SUFFIX=xx definition specifies the suffix of the PLTPI table that is created

2. Assemble and link-edit TSSCPLT.

3. Define a CICS RDO program definition for the TSSCPLT program. This was done automatically if you executed the TSSCSD job in the task "Update RDO Definitions". If you skipped this task, use the RDO command:

   ```
   DEFINE PROGRAM(TSSCPLT) GROUP(TOSGRP)DESCRIPTION(CA TSS CICS INITIALIZATION
   VERIFICATION) LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(CICS)RESIDENT(NO)
   USAGE(NORMAL) USELPACOPY(NO) STATUS(ENABLED) CEDF(NO) DATALOCATION(ANY)
   ```

4. Define the PLTPI table module to CICS with a program definition in the CICS RDO file.

5. Specify the table to CICS with the keyword:

   ```
   PLTPI=xx
   ```

   xx

   The suffix of the DFHPLT*xx* table module created in step 1.

   This keyword can be specified in:

   - The DFHSIT table (DFHSIT overrides the CICS execution JCL)
   - The CICS SYSIN file (if used)

For information on DFHPLT tables, see the IBM *CICS Transaction Server for z/OS System Definition Guide.*

Facilities Matrix

Most data centers have multiple CICS regions-each region having its own purpose. These regions are, at a minimum, segregated for test and production usage. Users who sign on to test regions are generally allowed greater freedom in accessing data and issuing transactions than a user who signs on to production regions.

In addition, it might be desirable to have many users access a certain region, such as one dedicated to CA-eMAIL+ or a similar application, while limiting a select group of users to a sensitive region, such as one dedicated to customer inquiry.

To describe your CICS region, you must associate an CA Top Secret facility with the region via an entry in the Facility Matrix Table. Using this table, CA Top Secret allows each region to be associated with a separate facility or for several regions to be associated with the same facility.

The Facilities Matrix contains general and CICS-specific suboptions of the CA Top Secret FACILITY control option. You can configure these suboptions in the Facilities Matrix to customize your CICS security on a facility-by-facility basis. For example, you can tailor access with Bypass Lists, set terminal LOCKTIME thresholds, and control CICS security parameters with these suboptions.

For information on the FACILITY suboptions, see the Control Options Guide. For information on how to configure these, see the chapter, “Implementing Security.”

CICS Default Facilities (CICSPROD and CICSTEST)

The Facilities Matrix contains predefined security attributes for controlling CA Top Secret processing for CICSPROD and CICSTEST. These attributes are actually suboptions of the FACILITY control option. You can use the CICSPROD and CICSTEST default facilities, or you can customize them for your site.

The defaults for the CICSPROD facility are as follows:

```
FACILITY DISPLAY FOR CICSPROD
INITPGM=DFH      ID=C  TYPE=004
ATTRIBUTES=ACTIVE,SHRPRF,ASUBM,NOABEND,MULTIUSER,NOXDEF
ATTRIBUTES=LUMSG,STMSG,SIGN(M),INSTDATA,RNDPW,AUTHINIT
ATTRIBUTES=NOPROMPT,NOAUDIT,NORES,WARNPW,NOTSOC,LCFTRANS
ATTRIBUTES=MSGLC,NOTRACE,NOEODINIT,IJU,NODORMPW,NONPWR
ATTRIBUTES=LUUPD
MODE=WARN  DOWN=GLOBAL  LOGGING=ACCESS,INIT,SMF,MSG,SEC9
UIDACID=8 LOCKTIME=000 DEFACID=*NONE*   KEY=8
FACMATRX=NO       EXTSEC=YES      EJBRPRFX=NO
XJCT=YES XFCT=YES XCMD=YES XDCT=YES XTRAN=YES XDB2=NO  XEJB=NO
XTST=YES XPSB=YES XPCT=YES XPPT=YES XAPPC=NO  XUSER=NO
XHFS=NO  XRES=NO
PCTEXTSEC=OVERRIDE    PCTCMDSEC=OVERRIDE  PCTRESSEC=OVERRIDE
DSNCHECK=NO   LTLOGOFF=NO       RLP=NO   SLP=NO   PCLOCK=NO
MAXUSER=03000  PRFT=003  MAXSIGN=010,RETRY
CICSCACHE=TASKLIFE,NOAUDIT,0512
```

The defaults for the CICSTEST facility are as follows:

```
FACILITY DISPLAY FOR CICSTEST
INITPGM=DFH      ID=K  TYPE=004
ATTRIBUTES=ACTIVE,SHRPRF,ASUBM,NOABEND,MULTIUSER,NOXDEF
ATTRIBUTES=LUMSG,STMSG,SIGN(M),INSTDATA,RNDPW,AUTHINIT
ATTRIBUTES=NOPROMPT,NOAUDIT,NORES,WARNPW,NOTSOC,LCFTRANS
ATTRIBUTES=MSGLC,NOTRACE,NOEODINIT,IJU,NODORMPW,NONPWR
ATTRIBUTES=LUUPD
MODE=WARN  DOWN=GLOBAL  LOGGING=ACCESS,INIT,SMF,MSG,SEC9
UIDACID=8 LOCKTIME=000 DEFACID=*NONE*   KEY=8
FACMATRX=NO       EXTSEC=YES      EJBRPRFX=NO
XJCT=YES XFCT=YES XCMD=YES XDCT=YES XTRAN=YES XDB2=NO  XEJB=NO
XTST=YES XPSB=YES XPCT=YES XPPT=YES XAPPC=NO  XUSER=NO
XHFS=NO  XRES=NO
PCTEXTSEC=OVERRIDE    PCTCMDSEC=OVERRIDE  PCTRESSEC=OVERRIDE
DSNCHECK=NO   LTLOGOFF=NO       RLP=NO   SLP=NO   PCLOCK=NO
MAXUSER=03000  PRFT=003  MAXSIGN=010,RETRY
CICSCACHE=TASKLIFE,NOAUDIT,0512
```

Display CICSPROD Default Bypass and Protect Lists

To display the default Bypass and Protect Lists, enter the following command:

```
TSS MODIFY FACILITY(CICSPROD=BYPLIST)
```

The lists appear.

Important! The ellipsis (....) punctuation is essential and represents internal CICS transactions with hexadecimal unprintable names.

```
FACILITY DISPLAY FOR CICSPROD
BYPASS TABLE DISPLAY FOR FACILITY  CICSPROD
RESOURCE=LOCKTIME BYPASS  NAMES:   TSS
RESOURCE=TRANID   BYPASS  NAMES:   CAQP   CATA   CATD   CATP
 CATR   CAUT   CCIN   CCMF   CDBD   CDBN   CDBO   CDBT
 CDTS   CECS   CEGN   CEHP   CEHS   CESC   CESF   CESN
 CFTS   CGRP   CITS   CLQ2   CLR1   CLR2   CLS3   CLS4
 CMPX   CMTS   CNPX   COVR   CPLT   CPMI   CQPI   CQPO
 CQRY   CRDR   CRMD   CRSQ   CRSR   CRSY   CRTE   CRTR
 CSAC   CSCY   CSFU   CSGM   CSGX   CSHR   CSIR   CSJC
 CSKP   CSLG   CSMI   CSM1   CSM2   CSM3   CSM4   CSM5
 CSNC   CSNE   CSPG   CSPK   CSRK   CSPP   CSPQ   CSPS
 CSRS   CSSC   CSSF   CSSN   CSSX   CSSY   CSTA   CSTB
 CSTE   CSTP   CSTT   CSXM   CSXX   CSZI   CVMI   CVST
 CWTR   CXCU   CXRE   CXRT   TS     8888   9999   ....
 ....   ....   ....   ....   ....   CFTL   CFSL   CKTI
 CKAM   CFCL   CIOD   CIOF   CIOR   CIRR   CJTR   CSHA
 CSHQ   CSOL   CTSD   CWBG   CWXN   CDBF   CEX2   CFQR
 CFQS   CSFR   CSQC   CDBQ   CRMF   CLSG   CFOR   CJMJ
 CLS1   CLS2   CPIH   CPIL   CPIQ   CRTP   CWXU   CPIR
 CPIS   CISC   CISD   CISE   CISR   CISS   CIST   CJGC
 CJPI   CISB   CEPD   CEPM   CISQ   CISU   CISX   CIS4
 CRLR   CISM   CEPF   CPSS   CJSR   CESL   CISP   CIS1
 CJSL   CRST   CPCT   CFCR   CJLR
RESOURCE=TRANID   PROTECT NAMES:   CEDF   TSEU
```

Display CICSTEST Default Bypass and Protect Lists

To display the default Bypass and Protect Lists, issue the following command:

```
TSS MODIFY FACILITY(CICSTEST=BYPLIST
```

The lists appear.

Important! The ellipsis (...) punctuation is essential and represents internal CICS transactions with hexadecimal unprintable names.

```
FACILITY DISPLAY FOR CICSTEST
BYPASS TABLE DISPLAY FOR FACILITY  CICSTEST
RESOURCE=LOCKTIME BYPASS  NAMES:   TSS
RESOURCE=TRANID   BYPASS  NAMES:   CAQP   CATA   CATD   CATP
        CATR   CAUT   CCIN   CCMF   CDBD   CDBN   CDBO   CDBT
        CDTS   CECS   CEGN   CEHP   CEHS   CESC   CESF   CESN
        CFTS   CGRP   CITS   CLQ2   CLR1   CLR2   CLS3   CLS4
        CMPX   CMTS   CNPX   COVR   CPLT   CPMI   CQPI   CQPO
        CQRY   CRDR   CRMD   CRSQ   CRSR   CRSY   CRTE   CRTR
        CSAC   CSCY   CSFU   CSGM   CSGX   CSHR   CSIR   CSJC
        CSKP   CSLG   CSMI   CSM1   CSM2   CSM3   CSM4   CSM5
        CSNC   CSNE   CSPG   CSPK   CSRK   CSPP   CSPQ   CSPS
        CSRS   CSSC   CSSF   CSSN   CSSX   CSSY   CSTA   CSTB
        CSTE   CSTP   CSTT   CSXM   CSXX   CSZI   CVMI   CVST
        CWTR   CXCU   CXRE   CXRT   TS     8888   9999   ....
        ....   ....   ....   ....   ....   CFTL   CFSL   CKTI
        CKAM   CFCL   CIOD   CIOF   CIOR   CIRR   CJTR   CSHA
        CSHQ   CSOL   CTSD   CWBG   CWXN   CDBF   CEX2   CFQR
        CFQS   CSFR   CSQC   CDBQ   CRMF   CLSG   CFOR   CJMJ
        CLS1   CLS2   CPIH   CPIL   CPIQ   CRTP   CWXU   CFIR
        CPIS   CISC   CISD   CISE   CISR   CISS   CIST   CJGC
        CJPI   CISB   CEPD   CEPM   CISQ   CISU   CISX   CIS4
        CRLR   CISM   CEPF   CPSS   CJSR   CESL   CISP   CIS1
        CJSL   CRST   CPCT   CFCR   CJLR
RESOURCE=TRANID   PROTECT NAMES:   CEDF   TSEU
```

Defining a New Facility to the Matrix

In addition to the two CICS default facilities, a total of 222 predefined facilities are provided that you can use to define a new facility of your own. Your security administrator can easily define a facility to the Facilities Matrix by:

- Changing the name of one of the predefined USER facilities.
- Modifying the security attributes of the new facility to tailor security processing for that facility.

For example, if you have a region dedicated to CA-eMAIL+ and you wish to define a unique facility for it, all you need to do is rename one of the available USER facilities, identify it as a CICS-type region, and establish the initiating program (usually DFHSIP) using the FACILITY control option:

```
FACILITY(USER1=NAME=EMAIL)

FACILITY(EMAIL=TYPE=CICS,PGM=DFH)
```

Note: It is recommended that the FACILITY control options be set in the TSS Parameter File at startup; however, alternate entry methods (such as O/S MODIFY and the TSS MODIFY command) can be used.

Other FACILITY control suboptions (for example, MODE, LOCKTIME) can be customized. The effects of such suboptions are described in the *Control Options Guide* as well as later sections in this guide.

Once the new facility has been defined, the region can be associated with the facility.

Associating a CICS Region With a Region ACID

CICS regions are normally associated with a region ACID. In batch, this association occurs naturally through the regions JOB JCL statement and the USER operand as shown in the example below.

```
//CICST1 JOB (acct-parameter),USER=CICST1,...
```

For a started task, the association is administered through the CA Top Secret STC Table as shown in the example below.

```
TSS ADDTO(STC) PROCNAME(CICST1) ACID(CICST1)
```

Associating a CICS Region and a Facility

The association of a CICS region and a facility occurs by adding a MASTFAC parameter to the region ACID as shown in the example below.

```
TSS ADDTO(CICST1) MASTFAC(CICSTEST)
```

CICS started tasks that have no associated region ACID and CICS regions that have no associated MASTFAC will default to the CICSPROD facility. However, CA Top Secret recommends that MASTFAC be explicitly employed so that such associations are fully documented.

Note: This architecture allows the administrator to associate regions one-to-one or many-to-one with facility entries of the Facility Matrix Table.

Defining Separate Facilities for Regions

The advantages of defining separate facilities for each region or group of regions are:

- The TSS command allows a security administrator to specify which facilities a user can access. In other words, he can specify which CICS regions a user can sign on to.
- Operating modes and logging options are specified by facility. This allows one region to be in FAIL mode while another is in WARN mode.
- There are several other control options specified on a facility basis, such as LOCKTIME, which can also prove useful.
- The Limited Command Facility (LCF) allows a security administrator to include or exclude transactions by facility. This allows a user who has access to both CICSPROD and CICSTEST to have access to one set of transactions for CICSPROD and another set of transactions for CICSTEST
- The FACILITY parameter of the PERMIT function allows the security administrator to permit access to one set of resources (like OTRANs, PPTs, FCTs, and so on) for your CICSTEST region and another set of resources for your CICSPROD region.
- The ADMIN function allows a security administrator to establish which facilities a security administrator is responsible for. This provides separate administration for each CICS region.

Define the CICS Region Control ACID

Since a CICS region begins its execution as a batch job or a started task, an CA Top Secret ACID must be associated with each CICS region. This ACID must be able to access the BATCH or STC facility, and must be authorized to all z/OS data sets used within the region, since these data sets are opened by CICS itself. This ACID is referred to as the CICS region control ACID. The ACID is associated with the region, via the USER=acidname parameter in the JCL for the CICS region initiated as BATCH job, or via the CA Top Secret STC table for a region initiated as a started task.

Examples: defining CICS control ACID

This examples defines a region acid and associates a CICS region acid (CICSP1) with the CICSPROD default facility:

```
TSS CREATE(CICSP1) NAME('CICS PRODUCTION REGION')
                   FACILITY(BATCH,STC)
                   PASSWORD(xxxx,0)
                   DEPARTMENT(deptacid)
                   MASTFAC(CICSPROD)
                   NORESCHK
                   NOLCFCHK
                   NODSNCHK
                   NOVOLCHK
                   NOSUBCHK
                   SOURCE(INTRDR)
```

This example defines a region acid and associates a CICS region acid (CICST1) with the CICSTEST default facility:

```
TSS CREATE(CICST1) NAME('CICS TEST REGION')
                   FACILITY(BATCH,STC)
                   PASSWORD(xxxx,0)
                   DEPARTMENT(deptacid)
                   MASTFAC(CICSTEST)
                   NORESCHK
                   NOLCFCHK
                   NODSNCHK
                   NOVOLCHK
                   NOSUBCHK
                   SOURCE(INTRDR)
```

FACILITY(BATCH,STC)

You must specify BATCH as a facility if CICS is submitted as a job or if batch jobs are submitted by CICS. Batch job submission also requires the ASUBM FACILITY suboption. You only need to specify the STC facility if you plan to start CICS as a started task.

SOURCE(INTRDR)

Prevents started tasks for the region ACID except through the internal reader.

MASTFAC(facility)

Must be specified with the CICS region. Users cannot log on unless MASTFAC is added to their user or profile record. For information, see the chapter, “Implementing Security”.

The following bypass attributes limit the involvement of the region ACID in security checking. If you do not bypass the resources and transactions for the region ACID, then the region ACID will require the permission for every resource and transaction available to users of the region.

PASSWORD(xxxx,0)

Defines the region ACID with a password. CA recommends all started task acids be defined in the STC table and OPTIONS(4) be set in the security parameter file so that when the STC is started, there is no password prompt but if someone tries to signon using that acid, the password must be entered.

NODSNCHK

Prevents DSN checking at OPEN time. If you do not specify NODSNCHK, all data set (FCTs), journals (JCTs), extra-partition destinations (DCTs), libraries (STEPLIBs and DFHRPLs) and CICS system files (RDO, DUMP, TEMPSTOR, and so on) must be permitted to the region control ACID. With dynamic FCT and DSN checking, this explicit permission might not be desirable; it is recommended for production regions only.

If DSNCHECK is set in facility, RES must also be specified.

NOLCFCHK

Bypasses LCF checking.

NORESCHK

Bypasses security checking for owned resources, including OTRAN, PPT, and so on.

NOSUBCHK

Allows jobs to be submitted to batch without the ACID authorizations normally required.

NOVOLCHK

Used to prevent volume problems for tape journals against the region ACID.

Notes:

- You can specify the NODSNCHK, NORESCHK, and NOLCFCHK attributes for the region control ACID. If you do not specify these attributes, every resource and/or LCF-protected transaction ID will have to be permitted to the region control ACID.
- CICS issues security check calls during initialization that determine if the CICS region ACID has authority to execute category 1 transactions such as CATA. The CICS region ACID should be permitted to the transactions using the NOLCFCHK bypass attribute or by permitting the category 1 transactions as LCF or OTRAN resources.

OMVS Considerations for CTS 2.2 and Above

Web Initialization with Java for CTS 2.2 and above requires that you add OMVS security parameters to properly initialize the CICS region. Minimally, we recommend the following:

```
tss add(cicsnnt) group(omvsgrp)
                 dfltgrp(omvsgrp)
                 uid(0) home(/)
                 omvspgm(/bin/sh)
```

If OMVS parameters are not present, you experience a SEC6 ABEND during CICS initialization. Although CICS continues to initialize, full initialization of the web interface is curtailed and might not be secure. Ensure that the region ACID has fully defined its OMVS parameters and recycle the region.

OMVS information is also required on the CICS DFLTUSER ACID. It is not necessary to give superuser status to the DFLTUSER:

```
tss add(cicsuser) group(omvsgrp)
                  dfltgrp(omvsgrp)
                  uid(314159)
                  home(/)
                  omvspgm(/bin/sh)
```

Ensure that access permission is granted to the HOME and OMVSPGM directories.

Defining the CICS SIT DFLTUSER ACID

CICS uses the SIT DFLTUSER during region start-up, during session initiation, and during signon processing. During region start-up, CICS signs the DFLTUSER on without an associated terminal. The non-terminal CICS ACEE generated by this signon is then used for processing transactions for which a signon cannot be inferred.

This can include:

- Transactions initiated from PLT
- Transactions started from sessions where no signon has yet occurred
- Transactions initiated from DCT trigger without assigned user

The characteristics of this “default user” session cannot change throughout the life of the CICS region because the non-terminal session is only signed on at region start-up.

DFLTUSER Characteristics

The CICS region start-up will abort if the ACID assigned by the SIT DFLTUSER option does not have the capability to sign onto the region. This means that the DFLTUSER acid must have (minimally) the following characteristics:

- FACILITY must have the same facility as the region ACID MASTFAC
- PASSWORD must be non-expiring NOPW
- NOSUSPEND must be added to avoid VTHRESH processing

Propagated Attributes

The following attributes are propagated to any CICS ACID signon which does not explicitly have the following attributes set:

- OPCLASS
- OPIDENT
- OPPRTY
- OPTIME

If you have applications which require one or more of these attributes, default values should be included in your DFLTUSER ACID.

A user signed on as the SIT DFLTUSER cannot be locked. If an explicit lock command is issued by a user signed on by default, the message TSS6301I SIT DEFAULT USER CANNOT BE LOCKED is issued. If an implicit lock is added to the SIT default user TSS ADD(dfltuser) LTIME(2) the LTIME parameter is ignored. Automatic logoff processing through LTLOGOFF=YES will also not apply to the SIT default user.

Note: Do not confuse the CICS SIT DFLTUSER with the CA Top Secret facility DEFACID. Because the SIT DFLTUSER can be used by anyone in the CICS region, avoid providing unintended users with significant access to resources or to administrative functions within CA Top Secret.

Defining Permission for a Region ACID to Its VTAM APPLID

CICS checks whether a region ACID has permission to access the APPLID supplied in the region SIT. To implement this security, ownership is established first using a command like the one shown below.

```
TSS ADDTO(cicsdept) IBMFAC(DFHAPPL)
```

Permission must then be granted for the SIT APPLID explicitly using a command like the one shown below.

```
TSS PERMIT(cicsp1) IBMFAC(DFHAPPL.applid)
```

Note: This implementation is unnecessary if the region ACID *cicsp1* has the NORESCHK attribute.

Administration Requirements

The following sections explain how to define CICS to CA Top Secret after the installation is complete.

Defining the CA Top Secret MASTFAC Parameter

To associate the CICS region with the appropriate Facilities Matrix entry, you must add the MASTFAC parameter to the CICS region control ACID. If the CA Top Secret MASTFAC parameter is omitted, the CICS region control ACID is automatically associated with the CICSPROD facility.

If you omit the MASTFAC parameter when creating the CICS region control ACID, you can add it to the ACID via the CA Top Secret ADDTO command like this:

```
TSS ADDTO(acid) MASTFAC(facility)
```

Defining CICS

CICS can be defined to CA Top Secret as a started task or a batch job.

As a Started Task

In a previous example, the CICS region ACID (CICSP1) was associated with the STC and BATCH facilities. The next step is to define the actual started task procedure.

CICS can be defined to CA Top Secret as a started task with an entry like this:

```
TSS ADDTO(STC) PROCNAME(PRODPROC)
               ACID(CICSP1)
```

Defining a started task to CA Top Secret results in the association of that STC with a specified ACID.

Note: Be sure to include the CICS JCL in one of the system PROCLIBS.

As a Batch Job

An ACID created with FACILITY(BATCH) allows CICS to execute as a batch job. Therefore, the entries made while adding the CICS region ACID must contain FACILITY(BATCH). For example:

```
TSS ADDTO(CICSP1) FACILITY(BATCH)
```

In addition, your systems programmer must code the USER=*acidnam*e keyword on the batch job statement, then submit the necessary CICS JCL. Using the example above as a reference, your programmer would code USER=CICSP1 in the batch job statement.

CICS Table Changes

This section describes how to define CICS security parameters for the CA Top Secret environment. As part of this process, changes have to be made to your CICS tables before starting up CICS with CA Top Secret. The tables needing required, additional, or optional changes are listed next. Details regarding these changes appear in the following sections.

Required CICS Table Changes

For initial startup, changes may be required in the System Initialization Table (SIT).

Additional CICS Table Entries

For initial startup, additions need to be made to the CSD table for new programs and transactions. Sample CSD table entries can be found in the chapter “CSD PROGRAM and TRANSACTION Sample Entries” and in CAKOJCL0 member TSSCSD.

Optional CICS Table Changes

For initial startup, changes indicating that external security should be involved for the following tables are optional:

- Converting SNT RDM to TSS Commands
- Destination Control Table (DCT)
- TERMINAL Definitions
- Temporary Storage Table (TST)
- TRANSACTION Definitions

Required Table Changes

The following sections detail the changes you need to make to the SIT and PCT.

The System Initialization Table (SIT)

The System Initialization Table (SIT) contains parameter settings for CICS initialization. Included in the SIT are security-related parameters. There are two choices for implementing CA Top Secret security for your CICS region:

- You can decide to use the CICS security parameters coded in the SIT for CICS initialization.
- You can substitute equivalent security suboptions in the CA Top Secret Facilities Matrix for CICS initialization.

Note: If XPARMS are in the DFHSIT, then they are static; if they are in the Facility Matrix Table, they are dynamic.

SIT Security Parameter Settings

SIT security parameter settings recognized by CA Top Secret are listed on the following pages. Any other settings are not recognized.

CMDSEC=

Indicates whether to accept the CMDSEC value.

ASIS

The CMDSEC value is honored for all transactions; corresponds to PCTCMDSEC=HONOR.

ALWAYS

The CMDSEC value is overridden for all transactions and SPI security checking is forced; corresponds to PCTCMDSEC=OVERRIDE.

EJBRPRFX=ejbrole-prefix

Specifies a prefix that is used to qualify the security role defined in an enterprise bean's deployment descriptor. The prefix is applied to the security role when:

- A role is defined to an external security manager.
- CICS calls the external security manager to perform method authorization checks

An application invokes the following method:

isCallerInRole()

You can specify a prefix of up to 16 characters. The prefix must not contain a period (.) character. If you specify a prefix that contains lowercase characters, blanks, or punctuation characters, you must enclose it in apostrophes. If the prefix contains an apostrophe, code two successive apostrophes to represent it.

Note: The EJBROLEPRFX parameter is ignored if security role support is not enabled. To enable security role support you must specify SEC=YES and XEJB=YES.

Mixed case is not supported under CA Top Secret r8 and above or Facility sub option EJBRPRFX. However, you have mixed case support if you specify EJBROLEPRFX in the CICS SIT, and set FACMATRX=NO.

RESSEC=

Indicates whether to accept the RESSEC value.

ASIS

The RESSEC value is honored for all transactions.

ALWAYS

The RESSEC value is overridden for all transactions and resource security checking is forced.

Note: If FACMATRX=YES, RESSEC is set to OVERRIDE.

SEC=

Indicates whether CA Top Secret is active for this region.

YES

It is active for this region; corresponds to EXTSEC=YES.

NO

It is inactive; corresponds to EXTSEC=NO.

SNSCOPE=

Indicates whether a user is restricted from signing on multiple times within the designated scope. Valid values include:

NONE

(Default) No duplicate checking. This value is forced when SNSCOPE=CICS or SNSCOPE=NONE is found in the SIT during region initialization. This alteration is required so that the SIGNMULTI attribute can be enforced.

CICS

Duplicate signons disallowed within CICS region (with exceptions for region acid, DFLTUSER and PLTUSER, as well as for MRO signons). This value, when set, is altered to NONE by CA Top Secret. Enforcement of duplicate signon within a CICS region should be set by using SIGN(S) in the CICS region ACID MASTFAC facility.

MVSIMAGE

Duplicate signons disallowed for CICS regions in the same MVS image. Some anomalies might occur where CA Top Secret successfully signs the user on but the signon is later rejected by CICS due to this setting. So that there is no contradiction between CICS and CA Top Secret enforcement, SIGN(M) should be used on the associated CICS region ACID MASTFAC facility.

SYSPLEX

Duplicate signons disallowed for CICS regions in the same SYSPLEX. Some anomalies might occur where CA Top Secret successfully signs the user on, but the signon is later rejected by CICS due to this setting. So that there is no contradiction between CICS and CA Top Secret enforcement, SIGN(M) should be used on the associated CICS region ACID MASTFAC facility.

XAPPC=

Indicates whether APPC session security can be used.

YES

Uses session security

NO

Session security is not used.

XCMD=

Indicates whether EXEC CICS commands are checked by CA Top Secret.

YES

All SPI commands are checked.

NO

SPI commands are not checked.

SPI commands include both CEMT commands and EXEC CICS SPI commands from an application program.

XDB2=

Indicates whether XDB2 activities are checked.

CTSDB2

The DB2ENTRY AND DB2TRANS resource checks are performed under *one* of the following two conditions:

- If CICS FACILITY FACMATRX=YES and XDB2=YES.
- If CICS FACILITY FACMATRX=NO and CICS SIT XDB2=CTSDB2.

NO

Checking is not performed by CA Top Secret.

XDCT=

Indicates whether transient data entries are checked by CA Top Secret.

YES

Transient data entries for this region are checked.

NO

Transient data entries for this region are not checked.

XEJB=

Specifies whether support of security roles is enabled.

YES

CICS Support for security roles is enabled:

- When an application invokes a method of an enterprise bean, CICS calls the external security manager to verify that the userid associated with the transaction is defined in at least one of the security roles associated with the method.
- When an application invokes the following method:

  ```
  isCallerInRole()
  ```

CICS calls the external security manager to determined whether the userid associated with the transaction is defined in the role specified on the method call.

NO

CICS support for security roles is disabled:

- CICS does not perform enterprise bean method level checks, allowing any userid to invoke any enterprise bean method.
- The following method always returns a value of TRUE:

  ```
  isCallerInRole()
  ```

Note: To enable security role support, you must also specify SEC=YES.

XFCT=

Indicates whether File Control entries for the region are checked by CA Top Secret.

YES

File control entries for this region are checked.

NO

File control entries for this region are not checked.

XHFS = YES | NO

(CTS 3.2 and above) Specifies whether CICS performs security checking for Web Client access to HFS files.

XJCT=

Indicates whether journal entries are checked for this region by CA Top Secret.

YES

Journal control entries for this region are checked.

NO

Journal control entries for this region are not checked.

XPCT=

Indicates whether EXEC-started transactions for this region are checked by CA Top Secret.

YES

Tranids specified on EXEC CICS START, INQ, SET, DISCARD, and COLLECT STATISTICS commands for this region are checked.

NO

Tranids specified on EXEC CICS START, INQ, SET, DISCARD, and COLLECT STATISTICS commands for this region are not checked.

XPPT=

Indicates whether program entries for this region are checked by CA Top Secret.

YES

Program entries for this region are checked.

NO

Program entries for this region are not checked.

XPSB=

Indicates whether PSB entries for this region are checked by CA Top Secret.

YES

Database PSB entries for this region are checked.

NO

Database PSB entries for this region are not checked.

XRES = YES | NO

(CTS 3.2 and above) CICS document templates (DOCTEMPLATE resource definitions).

XTRAN=

Indicates whether attached transaction entries for this region are checked by CA Top Secret.

YES

Transaction entries for this region are checked prior to execution.

NO

Transaction entries for this region are not checked prior to execution.

XTST=

Indicates whether temporary storage entries for this region are checked by CA Top Secret.

YES

Temporary storage keys for this region are checked.

NO

Temporary storage keys for this region are not checked.

XUSER=

Indicates whether surrogate user checking is performed by CA Top Secret.

YES

Performs surrogate user checking, including non-terminal (background) level security.

NO

Does not perform surrogate user checking.

Note: Except for XAPPC and XUSER, XPARMS are in effect only when RESSEC=YES is specified on the transaction or PCTRESSEC=OVERRIDE is in effect.

Update the Signon Transaction Definition (PCT)

The signon transaction should be excluded from CICS SPURGE processing. SPURGE valid transactions are purged from the system by CICS during periods of stress. This is not desirable for signon, since this can lead to abends or overlays, if a signon is purged simultaneously with CA Top Secret returning the user's signon environment.

CESN should be copied from IBM-supplied group to one capable of maintenance. Alter SPURGE attribute to No.

Activating CA Top Secret Security

To use CA Top Secret to secure your region, you must activate it and CAIENF (both CAIENF and CICS must be active). You can use two methods to activate CA Top Secret security in a CICS region:

- Set the SEC security parameter in the DFHSIT to YES (see the section, SIT Security Parameter Settings, for details) or
- Set the FACMATRX and the CA Top Secret EXTSEC suboptions to YES. The facility can be shared by multiple CICS regions. If the FACMATRX suboption is specified, all regions with the facility would have CA Top Secret activated.

The FACMATRX(YES) suboption overrides the DFHSIT security parameter settings and uses the equivalent CA Top Secret FACILITY suboptions to implement security.

Optional CICS Table Changes

Changes to these CICS tables are optional: DCT, FILE, JCT, TERMINAL, TST, and TRANSACTION. This section describes the changes to these tables in detail.

TERMINAL Definitions

The TERMINAL definitions contain terminal ID information. The following security parameters can be defined. See IBM's *CICS Resource Definition Guide* for specific details.

For assembled DFHTCT TYPE=TERM entries for individual terminals; or for CEDA defined TERMINAL definitions in the CSD file.

OPERID

For users defined to CA Top Secret, the OPERID record is accessed from the CA Top Secret Security File via the OPIDENT keyword.

OPERPRI

This value sets the default operator priority for transaction initiated from this terminal.

UCTRAN(YES|TRANID|NO)

Specifies whether transaction text is automatically translated at the terminal:

YES

All data entered is automatically translated to uppercase. (Default.)

TRANID

The TRANID is converted to uppercase, but the terminal buffer is not translated.

NO

Uppercase translation does not take place.

UCTRAN(YES) cannot be used when mixed case passwords are required in a CICS region. Mixed case passwords are appropriate for CTS 3.1 and above with z/OS 1.7 and above only.

TYPETERM Definitions

For assembled DFHTCT TYPE=TERMTYPE entries for terminal groups; or for CEDA defined TYPETERM definitions in the CSD file:

UCTRAN(YES|TRANID|NO)

Specifies capitalization policy default for terminals where the TYPETERM is a model to control whether:

YES

All data entered will be automatically uppercased (default)

TRANID

The TRANID will be uppercased, but the terminal buffer will remain untranslated

NO

No uppercase translation will take place.

UCTRAN(YES) cannot be used when mixed case passwords are required within a CICS region. For details of this terminal attribute, please see appropriate CICS documentation. Mixed case passwords are only appropriate for CTS 3.1 and above, with z/OS 1.7 and above.

SIGNOFF

This security parameter is honored by CA Top Secret.

USERID

The specified userid is signed on by CICS at the time the terminal is installed. The USERID must be defined to CA Top Secret and normal signon restrictions are enforced.

Note: TERMINAL output-only definitions are not protected terminals. ATS is not used for an output-only terminal.

TRANSACTION Definitions

Specify the following operands to indicate whether you want resource checking and SPI security checking done on this transaction.

RESSEC=

Indicates whether CA Top Secret activates security checking for resources.

YES

Activates security checking for resources used by the transaction.

NO

Bypasses security checking for resources used by the transaction.

CMDSEC=

Indicates whether CA Top Secret activates command security checking.

YES

Activates command (SPI) security checking for the transaction.

NO

Bypasses command security checking for the transaction.

Note: CA Top Secret security options can override these values if FACILITY PCTRESSEC=OVERRIDE or PCTCMDSEC=OVERRIDE is set.

DESTINATION CONTROL TABLE INTRAPARTITION Definitions

Trigger level transactions now run under the CICS default userid, not the userid of the signed-on user. Code the USERID=name operand with the userid you want CA Top Secret to use for security checking for the trigger level transaction specified on the TRANSID operand as follows:

```
DFHDCT TYPE=INTRA,DESTFAC=FILE,TRIGLEV=n,TRANSID=yyyy,
       USERID=acidname
```

See the IBM *CICS Resource Definition Guide* for details.

Note: Due to the automatic nature of DCT Signon, acidname must be able to sign on to the region, have permission to the DCT and the transaction yyyy, and have PASSWORD(NOPW,0).

TEMPORARY STORAGE TABLE Definitions

If you want security checking done on your temporary storage queues, you must establish the temporary storage queue security attribute for each explicit or generic queue name you want to secure.

If you use TST tables to define your temporary storage queues, you must reassemble your TST table with the following entry:

```
DFHTST TYPE=SECURITY,DATAID=character-string
```

If you use a TSMODEL to define your temporary storage queue, you must specify the SECURITY attribute for each model you want to secure:

```
SECURITY(YES)
```

For more details, see the *IBM CICS Resource Definition Guide*.

Setting CA Top Secret Security Inactive

There are two ways to deactivate security:

- Specify SEC=NO in the DFHSIT option and FACMATRX=NO

  Use this method to turn security off by region.

- Specify EXTSEC=NO and FACMATRX=YES in the Facilities Matrix suboptions.

  Use this method to turn security off by facility.

CICSplex Support

CICSplex requires running a CMAS address space that is simply a modified CICS region, although it is not usually signed on to directly. Under most circumstances, IBM recommends that this CMAS region run unsecured, although there might be circumstances when signing on to it is required for diagnostic functions.

Internally, this region has its own transactions defined, some of which are prefixed TS, and thereby conflict with the CA Top Secret transactions for administration and debugging. To avoid problems, we recommend that the CMAS region is associated with a unique TYPE=CICS facility. You can then decide whether you want to run this facility with SEC=NO or to secure it.

Setting up CICSPLEX with Security Active

If you do secure it, you must allow the transactions prefixed TS to run unrestricted. This can normally be done by adding the prefix to the TRANID bypass list, (which can already be the case).

Ensure that:

- The transactions are *not* defined to the PROT list.
- The CPSMOBJ, GCPSMOBJ, and CPSMXMP resource classes are defined in CA Top Secret
- The CPSMOBJ(OPERATE.) and CPSMOBJ(MONITOR.) access are owned and permitted

To activate CICSPLEX security

1. Define a facility.

   ```
   FACILITY(USERn=NAME=CPSMFAC,TYPE=CICS)
   ```

2. Create a region ACID:

   ```
   TSS CREATE(acid) TYPE(USER)
                    NAME(name)
                    DEPARTMENT(dept)
                    FACILITY(STC,BATCH)
                    MASTFAC(CPSMFAC)
                    PASSWORD(NOPW,0)
                    NORESCHK NODSNCHK NOVOLCHK
   ```

3. Define the STC proc to the STC table with the region ACID that was created in step 2:

   ```
   TSS ADDTO(STC) PROCNAME(CPSMPROC)
                  ACID(acid)
   ```

4. In SIT or SIT overrides, set SEC(YES).

5. To define the CPSMOBJ GCPSMOBJ CPSMXMP resources to the RDT, issue the following commands:

   ```
   TSS ADDTO(RDT) RESCLASS(CPSMOBJ)
                  RESCODE(xx)
                  ACLST(ALL,UPDATE,CONTROL,READ,NONE)
                  DEFACC(READ)

   TSS ADDTO(RDT) RESCLASS(GCPSMOBJ)
                  RESCODE(xx)
                  ACLST(ALL,UPDATE,CONTROL,READ,NONE)
                  DEFACC(READ)
   ```

```
TSS ADDTO(RDT) RESCLASS(CPSMXMP)
               RESCODE(xx)
               ACLST(ALL,UPDATE,CONTROL,READ,NONE)
               DEFACC(READ)
```

6. If CICSplex administrator authorities are being checked, IBMFAC(BBM.) should be protected and permitted:

```
TSS ADDTO(DEPTACID) IBMFAC(BBM.)
TSS PERMIT(acid) IBMFAC(BBM.)
```

7. Own and permit CPSMOBJ(OPERATE.), CPSMOBJ(MONITOR.), and CPSMOBJ(CONFIG.):

```
TSS ADDTO(dept)     CPSMOBJ(OPERATE.)
TSS PERMIT(acid)    CPSMOBJ(OPERATE.)
TSS ADDTO(dept)     CPSMOBJ(MONITOR.)
TSS PERMIT(acid)    CPSMOBJ(MONITOR.)
TSS ADDTO(dept)     CPSMOBJ(CONFIG.)
TSS PERMIT(acid)    CPSMOBJ(CONFIG.)
TSS ADD(dept)       CPSMOBJ(TOPOLOGY)
TSS PER(acid)       CPSMOBJ(TOPOLOGY)
TSS ADD(dept)       CPSMOBJ(WORKLOAD)
TSS PER(acid)       CPSMOBJ(WORKLOAD)
TSS ADD(dept)       CPSMOBJ(BAS)
TSS PER(acid)       CPSMOBJ(BAS)
TSS ADD(dept)       CPSMOBJ(ANALYSIS)
TSS PER(acid)       CPSMOBJ(ANALYSIS)
```

Updating Access to CPSMOBJ(TOPOLOGY)

The acid may need READ and UPDATE access to CPSMOBJ(TOPOLOGY). To update the access:

- Ensure that the facility definition is uniform between the parmlib initialization value and references in the TSS database.

  Once aligned, valid messages appear.

- Ensure that the region default ACID and PLTPI default ACID have access to facility. (Only for new ACIDS and a new FACILITY.)
- Ensure that the consoles have access to CPSMFAC and are cross authorized to the CPSMACID. (Only for new ACIDS and a new FACILITY.)
- Assign the STC for CAS and CMAS to the CPSMACID.
- Ensure that all the transactions used by PLTPI are authorized.

Authorizing Access to the Temporary Storage Pools

You can control access of temporary storage (TS) servers to the TS pools in the coupling facility. Each TS server can be started as a job or started task. The name of the TS queue pool for a TS server is specified at server startup. Each TS pool can only have one TS server running on each z/OS image in the sysplex. Two security checks are made against the TS server's userid (the userid the job or started task is running under). To ensure that the server passes these checks, execute the following commands:

```
TSS ADD(deptacid) IBMFAC(IXLSTR)
TSS ADD(deptacid) IBMFAC(DFHXQ)

TSS PERM(regionacid) IBMFAC(IXLSTR.DFHXQLS.TSPRODQS) ACC(ALL)
TSS PERM(regionacid) IBMFAC(DFHXQ.TSPRODQS) ACC(ALL)
```

Converting SNT RDM to TSS Commands

The CAKSNMIG program can be used to convert CICS Signon Table information into CA Top Secret commands.

Note: The CAKSNMIG program should only be used if you are converting CICS internal security to CA Top Secret.

CICS encourages administrators to convert existing tables from RDM to RDO. In some of its latest releases, no Signon Table (SNT) can even be assembled. CA encourages administrators to convert existing SNT macros to external security commands using the CAKSNMIG utility, whose JCL is shown in the following:

```
//TRYIT EXEC PGM=CAKSNMIG,PARM=NOCRE
//STEPLIB DD DSN=your.tss.library,DISP=SHR
//DFHSNT DD DSN=yourcics.table.loadlib,DISP=SHR
//ADDTO DD DSN=your.output.addto.lib,DISP=SHR
//TSSCMDS DD DSN=your.output.create.lib,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSUDUMP DD SYSOUT=*
```

The following DD statements must be completed:

STEPLIB

The file containing the CA Top Secret load library.

DFHSNT

The file containing the DFHSNT load module to be converted.

ADDTO

The file containing the TSS ADDTO statements for OPID, OPPRTY, and OPCLASS keywords.

Note: The LRECL for the output statements must be 80, the BLKSIZE is determined by the Device type.

TSSCMDS

The file containing the TSS CREATE statements unless PARM=NOCRE was coded.

Note: The LRECL for the output statements must be 80, the BLKSIZE is determined by the Device type.

SYSPRINT

The listing of output of both files.

Generation Operation

CAKSNMIG takes the load library containing the assembled SNT as input. The program can be executed with and empty PARM or with PARM=NOCRE:

- When the program is executed with no parameter, TSS CREATE commands are routed to the TSSCMDS file and TSS ADDTO commands are routed to the ADDTO file.
- When executed with PARM=NOCRE, only the ADDTO file should contain commands. TSSCMDS file output is suppressed.
- For the PARM field, code PARM=NOCRE indicates that TSS CREATE statements are not generated for the userids.

It is the administrator's responsibility to edit these files for re-submission through TSO/batch SYSTSIN. All messages generated for this program are documented in the *Messages and Codes Guide*.

Sample output from TSSCMDS file is provided in the following.

```
TSS CREATE(MASTER) TYPE(USER)
                   NAME('CICS USER')
                   DEPARTMENT(CICSDEPT)
                   PASSWORD(MASTER,,EXP)

TSS ADDTO(MASTER) OPIDENT(MAS)
                  OPPRTY(255)
                  OPCLASS(1,2,6,18)

TSS CREATE(SAMPLE) TYPE(USER)
                   NAME('CICS USER')
                   DEPARTMENT(CICSDEPT)
                   PASSWORD(SAMPLE,,EXP)

TSS ADDTO(SAMPLE)  OPIDENT(SAM) -
                   OPCLASS(1,10,15,24)
```

Chapter 2: Control Option Requirements

This section contains the following topics:

Setting Security Modes

One of the key issues that a security administrator must resolve during the implementation of CA Top Secret is the selection of a security mode for CICS. CA Top Secret security for CICS can be implemented in such a manner that existing CICS security or CA Top Secret security is in effect.

Modes of Operation

Four modes of operation are supported for a CICS environment; DORMANT, WARN, IMPLEMENT, and FAIL. Modes are assigned at five different levels:

Global

The default for the entire CA Top Secret community. For example:

```
MODE(WARN)
```

Facility

Affects a particular facility within the community. For example:

```
FACILITY(CICS=MODE=IMPL)
```

Profile

Affects a particular group of users attached to the profile. For example:

```
TSS PERMIT(PROF01) MODE(IMPL)
```

User

Affects a particular user within the community.

```
TSS PERMIT(USER01) MODE(FAIL)
```

Resource

Forces a particular resource authorization to be processed in FAIL mode. For example:

```
TSS PERMIT(USER01) TERMINAL(L048T29)
                   ACTION(FAIL)
```

Note: The global level is implemented via the MODE control option, or on a facility level via the MODE= suboption of the FACILITY control option. The profile, user and resource levels are implemented via the PERMIT function of the TSS command.

Modes for Defined Users and Resources

How modes for users and resources defined to CA Top Secret are administered:

DORMANT

No security checking is performed.

WARN

If the user is permitted access, security checking is performed by CA Top Secret only. If the user is not permitted access to the resource, a warning message is issued to the user.

IMPLEMENT

Security checking is performed by CA Top Secret.

FAIL

Security checking is performed by CA Top Secret.

Also note that:

- If ACTION(FAIL) is added to the resource, the mode specified for the user is overridden. This means that any unauthorized access to the specified resource is failed, and authorized access is allowed, regardless of the mode specified for the user. See the *User Guide* for details about the ACTION attribute.
- If an unauthorized access occurs and the DRC indicates the NOVIOL suboption, the security violation is treated as any event, and authorized access overrides CICS security key checking regardless of the mode specified for the user. The violation is flagged, but the user is not failed.
- If the EXIT(ON) control option is specified, the CA Top Secret Installation Exit is activated. Security check return can be altered by this option.
- The USERID on a JOBCARD submitted by a CICS user must be permitted to both the user submitting the job and the region ACID. These cross-authorizations are checked in FAIL mode regardless of the user mode.

Modes for Defined Users and Undefined Resources

How modes for users defined to CA Top Secret and resources *not* defined to CA Top Secret are administered:

DORMANT

No security checking is performed.

WARN

No security checking is performed. If default protection is specified, a warning message is issued to the user.

IMPLEMENT

No security checking is performed. If default protection is specified, security checking is performed by CA Top Secret only. The user fails because the resource is undefined and therefore, not authorized for access.

FAIL

If default protection is specified, security checking is performed by CA Top Secret only. The user fails because the resource is undefined and, therefore, not authorized for access.

In addition to the information contained in the previous table, also note that:

- If ACTION(FAIL) is added to the resource, then the mode specified for the user is overridden. This means that any unauthorized access to the specified resource is failed and authorized access is allowed, regardless of the mode specified for the user. See the *User Guide* for details about the ACTION attribute.
- If an unauthorized access occurs and the DRC indicates the NOVIOL suboption, the security violation is treated as any event, regardless of the mode specified for the user. The violation is flagged, but the user is not failed.
- If the EXIT(ON) control option is specified, the CA Top Secret Installation Exit is activated. Security check return can be altered by this option.

Modes for LCF Checking

The product provides modes of operation for protection of transactions through the Limited Command Facility (LCF). Inclusive LCF lists are defined by the CA Top Secret TRANS function parameter. Exclusive LCF lists are defined by the CA Top Secret XTRANS function parameter.

Note: Transactions that are defined as OTRAN transactions override LCF transactions and are protected by the modes for defined users and resources (see page 47) or modes for defined users and undefined resources (see page 48). For a complete explanation of LCF protection, see the *CA Top Secret User Guide.*

The product protects LCF lists as follows:

- For inclusive LCF lists, the following modes are administered:

  DORMANT mode

  Provides no security.

  WARN, IMPLEMENT, or FAIL mode

  Performs security checking if the user has a TRANS LCF list for the facility, and the transaction ID that is accessed is found in that list.

- For exclusive LCF lists, the following modes are administered:

  DORMANT mode

  Provides no security.

  WARN mode

  Performs security checking if the user specifies an XTRANS LCF list, and the transaction that is accessed is *not* found in the list for the facility.

  IMPLEMENT or FAIL mode

  Fails the user if the user is defined to CA Top Secret and the transaction that is accessed is found in the XTRANS LCF list.

Example: Assign an Inclusive List to a User

This example gives a user an inclusive list (a list of transactions that the user is allowed to use) for facility CICSPROD:

```
TSS ADDTO(acid) TRANSACTIONS(CICSPROD,(PAYT,MAIL,PAYP))
```

Example: Assign an Exclusive List to a User

This example gives a user an exclusive list (a list of transactions that the user is *not* allowed to use) for facility CICSPROD:

```
TSS ADDTO(acid) XTRANS(CICSPROD,(PAYT,MAIL,PAYP))
```

Note: When the NOXDEF suboption is specified on the facility for users defined to CA Top Secret without TRANS or XTRANS lists defined, security checking is performed by CICS only in DORM and WARN modes. Access to the requested transaction is allowed in IMPLEMENT and FAIL modes only.

When the XDEF suboption is specified on the facility for users defined to CA Top Secret without TRANS or XTRANS lists defined, security checking is performed by CICS only in DORM and WARN modes. Access to the requested transaction is allowed. In IMPLEMENT and FAIL modes, CA Top Secret performs security checking, and access to the transaction is denied.

Setting CA Top Secret Control Options

In addition to setting CA Top Secret control options and parameters, there are CICS-specific security parameters that can be set to implement security. These security parameters are set via the suboptions of the FACILITY control option and are discussed in the next section.

Preparing for Mixed Case Passwords

To enable the use of mixed case passwords with CICS, update the CA Top Secret control options and security file.

To prepare for mixed case passwords:

- Use the TSSXTEND utility with the NEWPWBLOCK keyword to copy the security file. For information, see the *Installation Guide*.
- Update your TSS procedure to point at the new security file.
- Open your PARMFILE data set and set the NEWPW sub-option MC (mixed case).
- Provide a separate FACILITY for CICS releases capable of mixed case password entry. This facilitates diagnostic efforts and makes it possible to use the MULTIPW keyword with the facility to segregate mixed case passwords from the ALL-facility password which remains in upper case.

Using Mixed Case Passwords

If you alter an ACID ALL facility password to mixed case and attempt to sign on in an address space that does not accept mixed case passwords, the password will not match.

For information on creating a new CICS-type facility, see the section “Defining Separate Facilities for Regions".

To associate the new facility with the mixed case region assign it as the MASTFAC attribute of the associated region ACID.

Define a MULTIPW password for each ACID which will sign on to the mixed case capable region, for example:

```
TSS ADD(acid) PASSWORD(password,,EXP)
              FACILITY(mixable)
              MULTIPW
```

This:

- Adds the FACILITY(mixable) to the ACID
- Creates a password specifically for that facility
- Leaves the current password untouched
- Alters the display of the "normal" password to the ALL-facility

For example, entering the following commands from a TP monitor which does not automatically convert commands to uppercase:

```
TSS ADD(MULT01) PASS(cics1,,EXP) FAC(CICSTEST) MULTIPW
TSS LIST(MULT01) DATA(PASSWORD)
```

Produces the output:

```
ACCESSORID = MULT01 NAME = MULTIPW PEON USER1
ALL        =
CICSTEST   = EXPIRES = 01/01/80  INTERVAL = 30
TSS0300I  LIST  FUNCTION SUCCESSFUL
```

If an ACID signs only onto monitors which accept mixed case commands, and accept mixed case passwords, it is not necessary to segregate mixed case passwords using MULTIPW. For information on MULTIPW, see the *Command Functions Guide*.

CICS FACILITY Designation Types

There are two designation types that indicate when a suboption takes effect. These designation types are:

Dynamic

Takes place immediately.

After CICS Recycle

Takes effect only after CICS recycle. These options include:

The tables in CA Top Secret Features Suboptions, CICS SIT Facility Override Suboptions, Bypass List Suboptions, and Protect List Suboptions show the designation type for each suboption.

CICS FACILITY Facility Suboption Implementation Types

CICS FACILITY suboptions can be divided into four implementation types:

- CA Top Secret features suboptions whose implementation is controls the implementation of its features
- CICS SIT Facility override suboptions (controlled by FACMATRX)
- Resources in the Bypass List
- Resources in the Protect List

CICS FACILITY CA Top Secret Features Suboptions

The following table details CA Top Secret features suboptions and their corresponding designation types:

CA Top Secret Features Sub-options	Designation Type
CICSCACHE	After CICS recycle
DSNCHECK	Dynamic
LTLOGOFF	Dynamic
MAXSIGN	Dynamic
MAXUSER	After CICS Recycle
RLP	After CICS Recycle
SIGN(M)	Dynamic
SIGN(S)	Dynamic
SLP	Dynamic

CICS SIT Facility Override Suboptions

When FACMATRX=YES in the facility, the FACMATRX suboptions are substituted for options in the CICS SIT; when FACMATRX=NO, the corresponding values specified by the SIT take precedence.

Note: You must set the FACMATRX suboption to YES prior to using any of the associated suboptions listed here.

The following table details CICS SIT Facility Override suboptions and their corresponding designation types:

CA Top Secret Features Suboptions	Designation Type
EJBRPRFX **Note:** The EJBRPRFX suboption overrides EJBROLEPRFX in the SIT. Also, the EJBRPRFX suboption can only accept uppercase values at this time. To implement a mixed case EJBRPRFX, you must set FACMATRX=NO.	After CICS Recycle
EXTSEC **Note:** The EXTSEC suboption overrides the SEC setting in the SIT.	After CICS Recycle
PCLOCK	After CICS Recycle
PCTCMDSEC	Dynamic
PCTRESSEC	Dynamic
XAPPC	Dynamic
XCOMMAND	Dynamic
XDB2 (CICS 5.2 and above)	Dynamic
XDCT	Dynamic
XEJB	After CICS Recycle
XFCT	Dynamic
XHFS	Dynamic
XJCT	Dynamic
XRES	Dynamic
XPCT	Dynamic
XPPT	Dynamic

CA Top Secret Features Suboptions	Designation Type
XPSB	Dynamic
XTRAN	Dynamic
XTST	Dynamic
XUSER	Dynamic

Bypass List Suboptions

The following table details Bypass List suboptions and their corresponding designation types:

Bypass List Suboption	Designation Type
BYPADD(resource)	Dynamic
BYPREM(resource)	Dynamic
BYPLIST	Dynamic

Protect List Suboptions

The following table details Protect List suboptions and their corresponding designation types:

Protect List Suboption	Designation Type
PROTADD(resource)	Dynamic
PROTREM(resource)	Dynamic

Using Suboptions or DFHSIT Parameters

You can choose how to implement security for CICS initialization. You can use the DFHSIT security parameters or the equivalent CA Top Secret FACILITY suboptions to implement security for CICS initialization.

You can set a FACILITY suboption called FACMATRX to indicate whether you are using the DFHSIT security parameters or the CA Top Secret FACILITY suboptions.

- To use the FACILITY suboptions to implement security for CICS initialization, set the FACMATRX suboption to YES. Security is then implemented by the equivalent FACILITY suboptions.
- To use the DFHSIT security parameters to implement security for CICS initialization, set the FACMATRX suboption to NO.

The advantages of using CA Top Secret FACILITY suboptions for security implementation are:

- DFHSIT security parameters can be specified in several places, making it difficult to tell which parameters are actually being used.

By setting FACMATRX=YES, the security administrator can control the security parameters for CICS initialization with CA Top Secret regardless of the security parameter settings in the DFHSIT table. This provides greater control over enforced security checking by CA Top Secret for CICS.

Facility Suboptions

Both the DFHSIT security parameters and their CA Top Secret equivalent FACILITY suboptions are listed next. For a description of how to use the DFHSIT security parameters, see the IBM *CICS/ESA System Definition Guide*. See the *Control Options Guide* for information on how to specify FACILITY suboptions.

DFHSIT Parameters	FACILITY Suboptions
	FACMATRX
SEC=	EXTSEC=
XAPPC=	XAPPC=
XCMD=	XCMD=
XDB2=	XDB2=
XDCT=	XDCT=
XFCT=	XFCT=
XHFS=	XHFS=
XJCT=	XJCT=
XPCT=	XPCT=
XPPT=	XPPT=
XPSB=	XPSB=
XRES=	XRES=
XTRAN=	XTRAN=
XTST=	XTST=
XUSER=	XUSER=
CMDSEC=	PCTCMDSEC=
RESSEC=	PCTRESSEC=
XEJB=	XEJB=
EJBR0LEPRFX=	EJBRPFRX=

XDB2=NO|resource_class

Disables resource checking, or selects and enables resource class checking, for CICS/DB2 keywords:

- DB2CONN
- DB2ENTRY
- DB2TRANS

CICS performs security checking by substituting the SIT specified resource class for the keyword. During initialization, when XDB2 specifies a resource class, and FACMATRX=NO, CICS activates a profile for the specified class. It is the administrator's responsibility to assure that the resource class specified by XDB2 has been defined to CA Top Secret. When XDB2 specifies a valid resource class, the administrator is also expected to provide security for IBMFAC(DFHDB2.) as documented by IBM in the CICS RACF Security Guide.

Selectively Disabling CAIENF/CICS Calls

The Event Notification Facility (CAIENF) automatically calls CA Top Secret when any CICS resource is accessed. CA Top Secret then processes the call based on the FACILITY control option parameters set by your site.

You can eliminate unnecessary overhead by selectively disabling calls for CICS resources that are not protected by CA Top Secret.

To disable CAIENF/CICS calls:

- The facility entry must have FACMATRX set to YES.
- Specify which CA Top Secret CAIENF intercepts you want to disable via the XPARMs; for example, if you do not want FCT checking to take place, specify XFCT=NO.

Specify FACMATRX=NO to disable this process. CA Top Secret then uses the XPARMs specified in the DFHSIT.

CICS Resource Lists

You can construct two types of resource lists:

- The Bypass List
- The Protect List

The following table details the keywords that the Bypass and Protect Lists support:

Resource Keywords	Top Secret Keywords	Notes
SYSID	SYSID	See the chapter, “Security for a Multi-System Environment.”
TRAN	LCF, OTRAN	TRANS is an alias for TRAN. Transactions bypassed might be rejected because of secondary resource checks.
TRANID	LCF, OTRAN	Transactions bypassed in TRANID will also bypass secondary resource checks.
PCT	PCT	CICS started transaction and EXEC CICS commands: COLLECT STATISTICS TRAN DISCARD TRAN INQ TRAN INQ REQID SET TRAN CANCEL TRAN
LOCKTIME	LTIME LTLOGOFF	The CA Top Secret LOCKTIME Bypass List has no effect on OPTIME implementation.

Resource Keywords	Top Secret Keywords	Notes
FCT	FCT	Facility DSNCHECK=NO.
DSNAME	DSNAME	Facility DSNCHECK=YES.
PSB	PSB	For IMS PSB resources defined through ISC.
SPI	SPI	For all SPI resource checking, including CEMT.
CEMT	CEMT	For CEMT verbs such as SET, INQUIRE
DCT	DCT	CICS intra- and extra-partition transient data destinations.
JCT	JCT	CICS system log and journals.
PPT	PPT	CICS program names.
TST	TST	CICS temporary storage destinations
XRES	XRES	CICS Document templates

The Bypass List

The Bypass List lets you avoid security checking by CA Top Secret for the resources you place on this list. Any resource that is not on the Bypass List is checked by default.

- To place a resource on the Bypass List, use the Bypass List BYPADD FACILITY suboption
- To remove a resource, use the BYPREM FACILITY suboption
- To list the resources, use the BYPLIST suboption

Since resource names added to the Bypass List are interpreted as generic prefixes, to perform security checking for a resource that begins with a generic prefix you must put the resource name on the Protect List.

How to Track Execution of Transactions That Bypass Security Checking

Transactions in the TRANID Bypass list bypass transaction security checking. Through logging, you can identify users that have executed transactions in the list without the necessary resource authorization. You can then establish the necessary authorizations for the users and remove the transaction from the bypass list on the CICS facility.

Note: For logging to take place, transactions in the TRANID Bypass list (and their secondary resources) must be owned.

The process is as follows:

1. (If necessary) Add ownership for the transactions (and secondary resources) to enable logging.

   For example, you want to add transaction FILX to the list. FILX accesses file FILEA (a secondary resource), and FACMATRIX=YES, XFCT=YES, and DSNCHECK=NO are set on the CICS facility definition in CA Top Secret. You can own FILX and FILEA by issuing the following command:

   ```
   TSS ADD(dept) OTRAN(FILX)
   TSS ADD(dept) FCT(FILEA)
   ```

2. Add the transactions to the bypass list:

   ```
   TSS MODIFY FACILITY(CICSPROD=BYPADD(TRANID=transaction_name+A))
   ```

3. Use TSSUTIL to generate an audit record violation for ACIDs that do not have an authorization defined for the transactions (or resources that the transactions are using).

   The TSSUTIL report lists the violations. You need to accumulate enough data to determine which ACIDs need to be permitted to the transaction and resources.

4. After you have gathered your data, create the necessary authorizations by permitting the applicable transaction and resources to the user.

5. When all required authorizations are built, remove the transactions from the list:

   ```
   TSS MODIFY FACILITY(CICSPROD=BYPREM(TRANID=transaction_name))
   ```

Example: TRANID Bypass List with Added Transactions

In this example, the CSMI and FILX transactions have been modified with the (+A) extension:

```
TSS9550I FACILITY DISPLAY FOR CICSPROD
TSS9570I BYPASS TABLE DISPLAY FOR FACILITY  CICSPROD
TSS9571I RESOURCE=LOCKTIME BYPASS  NAMES:   TSS
TSS9571I RESOURCE=TRANID   BYPASS  NAMES:   CAQP   CATA   CATD   CATP
TSS9572I        CATR   CAUT   CCIN   CCMF   CDBD   CDBN   CDBO   CDBT
TSS9572I        CDTS   CECS   CEGN   CEHP   CEHS   CESC   CESF   CESN
TSS9572I        CFTS   CGRP   CITS   CLQ2   CLR1   CLR2   CLS3   CLS4
TSS9572I        CMPX   CMTS   CNPX   COVR   CPLT   CPMI   CQPI   CQPO
TSS9572I        CQRY   CRDR   CRMD   CRSQ   CRSR   CRSY   CRTE   CRTR
TSS9572I        CSAC   CSCY   CSFU   CSGM   CSGX   CSHR   CSIR   CSJC
TSS9572I        CSKP   CSLG   CSMI+A CSM1   CSM2   CSM3   CSM4   CSM5
TSS9572I        CSNC   CSNE   CSPG   CSPK   CSRK   CSPP   CSPQ   CSPS
TSS9572I        CSRS   CSSC   CSSF   CSSN   CSSX   CSSY   CSTA   CSTB
TSS9572I        CSTE   CSTP   CSTT   CSXM   CSXX   CSZI   CVMI   CVST
TSS9572I        CWTR   CXCU   CXRE   CXRT   TS     8888   9999   ....
TSS9572I        ....   ....   ....   ....   ....   CFTL   CFSL   CKTI
TSS9572I        CKAM   CFCL   CIOD   CIOF   CIOR   CIRR   CJTR   CSHA
TSS9572I        CSHQ   CSOL   CTSD   CWBG   CWXN   CDBF   CEX2   CFQR
TSS9572I        CFQS   CSFR   CSQC   CDBQ   CRMF   CLSG   CFOR   CJMJ
TSS9572I        CLS1   CLS2   CPIH   CPIL   CPIQ   CRTP   CWXU   CPIR
TSS9572I        CPIS   CISC   CISD   CISE   CISR   CISS   CIST   CJGC
TSS9572I        CJPI   CISB   CEPD   CEPM   CISQ   CISU   CISX   CIS4
TSS9572I        CRLR   CISM   CEPF   CPSS   CJSR   CESL   CISP   CIS1
TSS9572I        CJSL   CRST   CPCT   CFCR   CJLR   FILX+A
TSS9571I RESOURCE=TRANID   PROTECT NAMES:   CEDF   TSEU
```

Example: Logging That Shows Transaction Execution Violations

In this example, CSMI and FILX transaction execution results in logged violations:

```
05/09/07 11:19:59 XE56 C230AOR  CTS230A  K F DFHMIRS EXECUTE NONE *08*-88  +CSMI FILE
05/09/07 11:19:59 XE56 LUGBR06  CTS230A  K F DFHMIRS EXECUTE NONE *08*-88 Q +CSMI
PGMFILE
                       S0006051 A56L810
05/09/07 11:19:59 XE56 LUGBR06  CTS230T  C F PGMFILX EXECUTE NONE *08*-88  +FILX FILEA
                       S0006050 A56L810
05/09/07 11:19:59 XE56 C230AOR  CTS230A  K F DFHMIRS READ   NONE  *08*-88 F +FILEA
                       S0006051
```

The Protect List

The Protect List is used to override generic resource prefixes defined in the Bypass List. If a resource is matched in both the Bypass List and the Protect List, the match in the Protect List controls processing, regardless of the length of the match.

- To place a resource on the Protect List, use the PROTADD FACILITY suboption.
- To remove a resource, use the PROTREM FACILITY suboption.

The following CICS resources can be used with the BYPADD, BYPREM, PROTADD, and PROTREM suboptions.

- CEMT
- DCT
- FCT
- JCT
- LOCKTIME
- PCT
- PPT
- PSB
- SPI
- SYSID
- TRAN
- TST
- TRANID
- XRES

Note: This list is intended for a limited number of resources and should not be used as an alternative for the ALL Record.

Examples: Bypass and Protect lists

This example avoids security checking for transactions beginning with XY:

```
TSS MODIFY FACILITY(CICSTEST=BYPADD(TRANID=XY)
```

You can still check for security on transaction XYZ by entering:

```
TSS MODIFY FACILITY(CICSTEST=PROTADD(TRANID=XYZ)
```

In this example, the PROTADD(TRANID=XYZ) command overrides the BYPADD(TRANID=XY) command.

Bypassing Security for CEMT Commands

Use the CEMT=action parameter to bypass the “action” on both the CEMT Extended Master Terminal Command and on the EXEC CICS “action” for which you want to bypass security checking.

Valid actions are:

- INQUIRE
- PERFORM
- SET
- DISCARD For example, to allow access to all CEMT INQUIRE commands, enter:
  `TSS MODIFY FACILITY(cicsfac=BYPADD(CEMT=INQUIRE))`

Note: To bypass SET you also need to add INQUIRE to the Bypass List because CEMT SET redisplays the items altered in the CEMT SET.

If CEMT=SET is specified, SPOOLWRITE JOB SUBMIT security under CA Top Secret will not work.

Bypassing Security for SPI Commands

To bypass all EXEC CICS INQUIRE commands, except SYSTEM, enter:

```
TSS MODIFY FACILITY(CICSTEST=BYPADD(SPI=INQUIRE))
```

To bypass EXEC CICS INQUIRE SYSTEM also enter:

```
TSS MODIFY FACILITY(CICSTEST=BYPADD(CEMT=INQUIRE))
```

Note: The above command will not bypass the OTRAN or LCF security checks for transaction CEMT, only the SPI security check is bypassed.

Bypass Transaction Security

To bypass transaction security, add an entry to the TRANID or TRAN parameter of the Bypass List. TRAN and TRANS are identical. The TRANID parameter contains transaction name entries that will bypass *all* security checking for the transaction. The default entries are:

```
TSS9550I FACILITY DISPLAY FOR CICSPROD
TSS9570I BYPASS TABLE DISPLAY FOR FACILITY  CICSPROD
TSS9571I RESOURCE=LOCKTIME BYPASS  NAMES:   TSS
TSS9571I RESOURCE=TRANID   BYPASS  NAMES:   CAQP   CATA   CATD   CATP
TSS9572I        CATR   CAUT   CCIN   CCMF   CDBD   CDBN   CDBO   CDBT
TSS9572I        CDTS   CECS   CEGN   CEHP   CEHS   CESC   CESF   CESN
TSS9572I        CFTS   CGRP   CITS   CLQ2   CLR1   CLR2   CLS3   CLS4
TSS9572I        CMPX   CMTS   CNPX   COVR   CPLT   CPMI   CQPI   CQPO
TSS9572I        CQRY   CRDR   CRMD   CRSQ   CRSR   CRSY   CRTE   CRTR
TSS9572I        CSAC   CSCY   CSFU   CSGM   CSGX   CSHR   CSIR   CSJC
TSS9572I        CSKP   CSLG   CSMI   CSM1   CSM2   CSM3   CSM4   CSM5
TSS9572I        CSNC   CSNE   CSPG   CSPK   CSRK   CSPP   CSPQ   CSPS
TSS9572I        CSRS   CSSC   CSSF   CSSN   CSSX   CSSY   CSTA   CSTB
TSS9572I        CSTE   CSTP   CSTT   CSXM   CSXX   CSZI   CVMI   CVST
TSS9572I        CWTR   CXCU   CXRE   CXRT   TS     8888   9999   ....
TSS9572I        ....   ....   ....   ....   ....   CFTL   CFSL   CKTI
TSS9572I        CKAM   CFCL   CIOD   CIOF   CIOR   CIRR   CJTR   CSHA
TSS9572I        CSHQ   CSOL   CTSD   CWBG   CWXN   CDBF   CEX2   CFQR
TSS9572I        CFQS   CSFR   CSQC   CDBQ   CRMF   CLSG   CFOR   CJMJ
TSS9572I        CLS1   CLS2   CPIH   CPIL   CPIQ   CRTP   CWXU   CPIR
TSS9572I        CPIS   CISC   CISD   CISE   CISR   CISS   CIST   CJGC
TSS9572I        CJPI   CISB   CEPD   CEPM   CISQ   CISU   CISX   CIS4
TSS9572I        CRLR   CISM   CEPF   CPSS   CJSR   CESL   CISP   CIS1
TSS9572I        CJSL   CRST   CPCT   CFCR   CJLR
TSS9571I RESOURCE=TRANID   PROTECT NAMES:   CEDF   TSEU
TSS0300I MODIFY   FUNCTION SUCCESSFUL
```

To specify multiple transactions (up to four) on one line for the bypass list, enter the following command:

```
F TSS,FACILITY(cicsfac=BYPADD(TRANID=(trn1,trn2,trn3,trn4)
```

The difference between the Bypass List parameters TRAN and TRANID is that the entries for the TRAN list contain transaction names that will bypass resource OTRAN or LCF security checking only. Entries in the TRANID Bypass List contain transaction names that will bypass *all* types of security checking (OTRAN, LCF, FCT, or any type of resource check, including LOCKTIME, and job submit processing for transient data and spoolwrite).

Important! For CEDF processing, to ensure security checking of transactions and resources being emulated, never place CEDF in the TRANID Bypass List. Consider placing CEDF in the TRAN Bypass List instead.

If an EXEC CICS START TRANSACTION(tran) is issued from a transaction with RESSEC=YES in the PCT and you want to use the bypass list to avoid checks in the started transaction, you must add the started transaction to the PCT and TRANID bypass lists. The PCT bypass allows the start of the transaction, and the TRANID bypass allows access to any resource that the transaction might reference.

Bypassing Terminal Security

The TCT Bypass List contains terminal entries that will bypass CA Top Secret security checking where:

- VTAM= eight-character NETNAME
- TCAM= eight-character terminal ID
- BTAM= four-character terminal ID

For example, to bypass security checking for terminal K06L3544, enter:

```
TSS MODIFY FACILITY(cicsfac=BYPADD(TCT=K06L3544))
```

This command allows any transaction to be run on this terminal without signon entry validation or any resource checking.

Bypassing LOCKTIME Security

The LOCKTIME Bypass List contains terminal entries or transaction IDs that are not checked for lock time by CA Top Secret. When added to the Bypass List, these entries override the LOCKTIME control option settings for that terminal or transaction. You can bypass terminal lock time restrictions where:

- VTAM= eight-character NETNAME
- TCAM= eight-character terminal ID
- BTAM= four-character terminal ID

For example, to bypass LOCKTIME security for terminal K06L3544, enter:

```
TSS MODIFY FACILITY(CICSTEST=BYPADD(LOCKTIME=K06L3544))
```

To bypass LOCKTIME security for transaction PUBL, enter:

```
TSS MODIFY FACILITY(CICSTEST=BYPADD(LOCKTIME=PUBL))
```

Bypassing Security for Specific Resources

You can selectively bypass security checks for specific resources. The following Bypass Lists contain entries that are not checked by CA Top Secret:

DCT

Contains transient data entries.

DSNAME

Contains file control entries (DDNAMES) for data sets. The DSNCHECK= suboption must be set to YES.

FCT

Contains File Control Table entries (DDNAMES). The DSNCHECK= suboption must be set to NO.

JCT

Contains Journal Control Table entries (journal names).

PCT

Contains interval control started transaction identifiers.

PPT

Contains program entries.

PSB

Contains PSB entries.

TRANSACTIONS

Contains transaction identifiers.

TST

Contains Temporary Storage entries (queue names).

XRES

Contains document templates entries.

Additional Suboptions

This section explains how to use additional CA Top Secret FACILITY suboptions.

Limiting User Signon Storage

Use the MAXUSER= suboption to limit the amount of storage allocated by CA Top Secret CICS for session related tokens (SRTs), which are GETMAINed at CICS initialization time. The MAXUSER value is used to calculate the number of SRTs CA Top Secret CICS allocates to maintain a reference point for each signed-on user. If, during the life of the CICS region, the MAXUSER value is exceeded, additional SRTs are dynamically allocated to handle the new signon requests.

Note: The count for MAXUSER also includes MRO/ISC link signons and ATS (Automatic Terminal Signon) events. When setting this value, make sure you include MRO/ISC links and ATS terminal signons with the number of signed on users per CICS region.

For example, to limit the number of users (via User Control Blocks) via the CICS Payroll region to 500, you can use the TSS MODIFY command like this:

```
TSS MODIFY (FACILITY(CICS=MAXUSER=500))
```

The MAXUSER= suboption allocates the ACEEXREF cross reference table. This table:

- Contains one entry for each signed on and potentially signed on user
- Is allocated at startup of the region and does not expand dynamically

Note: After changing the MAXUSER FACILITY suboption, you must recycle your CICS region for the changes to take effect.

Controlling Simultaneous User Signon

Use the MAXSIGN= suboption to restrict the number of signons (or signoffs) that are made simultaneously by users. This suboption lets you set a threshold for the number of user signons that can be made concurrently, and controls the action taken if the threshold is exceeded. The default threshold is 10 (ten users can sign on/sign off concurrently). You can change the default threshold; valid values are 5 to 100, inclusive. For example, to change the threshold so 15 users can sign on concurrently, enter the following command. Note that the parentheses around the value are required.

```
TSS MODIFY FACILITY(CICSPROD=MAXSIGN=(15))
```

The action for MAXSIGN= can be set to KILL or RETRY. When KILL is set and the threshold of the queue is reached, all additional attempts to sign on or off are abended. For example, if you set the threshold for a CICS facility called CICSPAY at 15 and specify KILL as shown below, if 18 users try to sign on to CICSPAY concurrently, the first 15 users in the queue are signed on and the last three users receive a message and their attempt is abended.

```
TSS MODIFY (FACILITY(CICSPAY=MAXSIGN=(15,KILL))
```

When RETRY is set and the threshold of the queue is reached, all additional attempts to sign on are requeued to CICS. For example, if you set the threshold of a CICS facility called CICSPROD at 50 and specify RETRY as shown below, if 60 users try to sign on to CICSPROD concurrently, the first 50 users in the queue are signed on and the last 10 are sent back to CICS to be requeued.

```
TSS MODIFY FACILITY(CICSPROD=MAXSIGN=(50,RETRY))
```

Controlling Concurrent Signons by the Same User

The MASTFAC facility control suboption SIGN(M) might be set to allow concurrent signons by an ACID within a CICS region from multiple nodes or terminals. The CICS administrator should be careful that the SIT parameter SNSCOPE=CICS or NONE is set with this setting.

The MASTFAC facility control suboption SIGN(S) might be set to disallow concurrent signons by an ACID within a CICS region from multiple nodes or terminals. The CICS administrator should be careful that the SIT parameter SNSCOPE=CICS or NONE is set with this setting.

The SIGNMULTI attribute allows an administrator to provide for multiple concurrent signon by an ACID in spite of the facility sub-option SIGN(S). The SIGNMULTI attribute might only be provided through an individual facility. If a user already has access to the facility, it must first be removed:

```
TSS REMOVE(acid) FACILITY(cicsfac)
```

To provide SIGNMULTI access to the facility, use the following command:

```
TSS ADD(acid) FACILITY(cicsfac)
              SIGNMULTI
```

The SIGNMULTI feature will only be honored when SIT parameters SNSCOPE=CICS or SNSCOPE=NONE are set. CA Top Secret will update the SIT SNSCOPE definition as follows, only if SIGN(S) is specified on the FACILITY:

- If SNSCOPE is specified as MVSIMAGE or SYSPLEX, then SNSCOPE is not changed and SIGNMULTI will not work. In this case, TSEU=INSTALL will show SNSCOPE as defined in the SIT.
- If SNSCOPE is specified as CICS or NONE, then SNSCOPE is changed and SIGNMULTI will work. In this case, TSEU=INSTALL will show SNSCOPE=NONE.

SNSCOPE is a CICS SIT parameter which controls the scope in which duplicate signons are allowed at local terminals, or signing on after using the CRTE transaction to connect to another system. At CICS startup, CA Top Secret will honor the values SNSCOPE=MVSIMAGE or SYSPLEX: these values must only be used with facility sub-option SIGN(M); when these values are selected, SIGNMULTI is overridden by CICS. When SNSCOPE=CICS or NONE, CA Top Secret forces the SNSCOPE=NONE: in this case, duplicate signon processing is controlled entirely by CA Top Secret.

The signon scope (SNSCOPE) is enforced with the MVS ENQ macro where there is a limit on the number of outstanding MVS ENQs per address space. If this limit is exceeded, the MVS ENQ is rejected and CICS is unable to detect if the user is already signed on. When this happens, the signon request is rejected with message DFHCE3587. See the *OS/390: MVS Programming: Authorized Assembler Services Guide* for guidance on increasing the MVS ENQ limit.

Securing Data Set Names Instead of FCTs

Use the DSNCHECK(YES|NO) suboption to perform CA Top Secret security checking on the FCT name or the DSN facility name.

- To perform security checking on the FCT name, specify DSNCHECK=NO.
- To perform security checking on the DSNAME name, specify DSNCHECK=YES.

  The RESOURCE FACILITY suboption is required for DSNAME name protection. The RESOURCE suboption brings the user's DSNAME and VOLUME permissions into storage and increases CA Top Secret memory requirements.

To indicate that security checking should be performed on all DSNAME names in the CICS Production 1 region, you can enter a command like this:

```
F TSS,FACILITY(CICSP1=DSNCHECK=YES,RES)
```

If only FCT checking is required, then the command would look like this:

```
F TSS FACILITY(CICSP1=DSNCHECK=NO,NORES)
```

For security checking on all data set names in the CICS Production 2 region, you can enter a command like this:

```
TSS MODIFY FACILITY(CICSP2=DSNCHECK=YES,RES)
```

Note: To provide protection for remote DSNAMEs, you must remove the CSMI transaction from the FACILITY TRANID Bypass List of the remote region. Security checking is performed in the region where the FCT resides.

CICS data set protection (DSNCHECK=YES) does not protect DL1 databases. CICS resources PSB and DBD are used in the CICS environment for this purpose.

Securing Transactions Not Associated with a Terminal

A surrogate user is a user who has the authority to start work on behalf of another user. A surrogate user is authorized to act for that user without knowing the other user's password. There are two ways to enable surrogate user checking:

- Specify XUSER=YES in the DFHSIT
- Specify FACMATRX=YES, then specify XUSER=YES in the Facilities Matrix

If surrogate user checking is employed, it applies to:

- CICS default user
- PLT post-initialization processing
- Preset terminal security
- Started transactions not associated with a terminal
- The userid associated with a transient data destination

If a userid is specified on the EXEC CICS START command, then this user is the one who is associated with the started non-terminal (background) transaction.

If the userid in the START command is not the current user, then the current user must be authorized to the userid specified on the START command.

For example, if the signed on userid is CURRUSER and the following command is issued:

```
EXEC CICS START TRANID('ABC') USERID('STARUSER')
```

You must permit authority for the SURROGAT resource by issuing the following commands:

```
TSS ADD(cicdept) SURROGAT(STARUSER)

TSS PER(CURRUSER) SURROGAT(STARUSER.DFHSTART)
                  ACC(UPDATE)
```

If the signed on user issues the command without USERID:

```
exec cics start tranid('ABC')
```

Then transaction ABC is started under the authority of the initiating user CURRUSER and no surrogate checking is performed.

To activate surrogate checking on background transactions, XUSER=YES and RES must be set in the CICS facility. For example:

```
TSS MODIFY FACILITY(CICSPROD=RES,XUSER=YES,FACMATRX=YES)
```

In the case that FACMATRX=NO, RES must still be set in the facility, but XUSER=YES must be set in the CICS SIT, for example:

```
TSS MODIFY FACILITY(FACMATRX=NO,RES)
```

Selecting CA Top Secret Security for Commands

There are two ways you can use the PCTCMDSEC= suboption:

- To override the CICS TRANSACTION CMDSEC setting and force a security check for *all* commands (the default), or
- To perform selective security checking for *specific* commands by honoring the CICS TRANSACTION CMDSEC parameter setting.

To perform security checking for all commands enter:

```
TSS MODIFY FACILITY(PAYACCT1=PCTCMDSEC=OVERRIDE)
```

To selectively perform security checking for specific commands (and honor the TRANSACTION CMDSEC setting) enter:

```
TSS MODIFY FACILITY(PAYACCT1=PCTCMDSEC=HONOR)
```

Selecting CA Top Secret Security for Resources

You can use the PCTRESSEC= suboption to perform security checking for the program, file, transient data, and temporary storage resources.

There are two ways you can use the PCTRESSEC= suboption:

- To override the CICS TRANSACTION RESSEC setting and force a security check for *all* resources (the default), or
- To perform selective security checking for *specific* transactions by honoring the CICS TRANSACTION RESSEC parameter setting.

To override native CICS transaction RESSEC security checking with CA Top Secret for all resources, enter:

```
TSS MODIFY FACILITY(PAYACCT1=PCTRESSEC=OVERRIDE)
```

To selectively perform security checking for specific transactions (and honor the DFHPCT RESSEC setting) enter:

```
TSS MODIFY FACILITY(PAYACCT1=PCTRESSEC=HONOR)
```

Enabling Record and Screen Level Protection

The data for Record Level Protection (RLP) and Screen Level Protection (SLP) is stored in the Static Data Table (SDT). Before you can enable RLP and SLP you must first initialize the SDT using the SDTBLOCKS parameter of TSSMAINT. You must then extend your old SECFILE into your new SECFILE using TSSXTEND.

The RLP=NO and SLP=NO suboptions are set by default in the CICS facilities. SLP can be dynamically modified during CICS execution. However, RLP cannot be activated dynamically because it requires file exits that are only installed by CA Common Services for z/OS during CICS initialization

RLP can be deactivated dynamically because CA Top Secret reviews the caller's RLP facility setting on every RACROUTE call.

Setting Pseudo-Conversational LOCKTIME Processing

You can use the PCLOCK= suboption to control whether TSS LOCKTIME processing operates conversationally (PCLOCK=NO, default) or pseudo-conversationally (PCLOCK=YES). You cannot set this option dynamically while a CICS region is running and it takes effect only during CICS initialization.

To enable pseudo-conversational locktime, set the PCLOCK facility suboption to YES.

```
TSS MODIFY FACILITY(CICSPROD=PCLOCK=YES)
```

To disable pseudo-conversational locktime, set the PCLOCK facility suboption to NO.

```
TSS MODIFY FACILITY(CICSPROD=PCLOCK=NO)
```

Note: If you are using the TSSPGM02 password prompt exit, be aware that this interface has been modified to support use in pseudo-conversational mode. Installations that use this exit program will need to modify their code. A new flag has been added in the incoming COMMAREA to indicate to the program whether it should issue a SEND or RECEIVE or a conversational prompt. A sample program is supplied in CAI.CAISRCS (member TSSPGM02).

The PCLOCK=NO suboption is set by default in CICS facilities. It cannot be activated dynamically during CICS execution. This option sets LOCKTIME processing to conversational mode.

The PCLOCK=YES subopotion sets CA Top Secret LOCKTIME processing to pseudo-conversational processing. Users might prefer the efficiency of this option.

Note: PCLOCK=NO is preferred in non-TOR MRO regions.

Securing Records Within a CICS File

You can use the RLP= suboption to perform security checking for records within a CICS file (FCT). To enable Record Level Protection (RLP), you must take the following steps:

- Set the XFCT security parameter in the FACILITY suboption to YES.
- Set the RLP parameter in the FACILITY suboption to YES.

For example:

```
TSS MODIFY FACILITY(CICSPROD=XFCT=YES,RLP=YES)
```

Securing Terminal Screen Input

You can use the SLP=suboption to perform security checking for terminal screen input data.

To enable Screen Level Protection (SLP), you must take the following steps:

- Set XTRAN or XPPT security parameter in the FACILITY suboption to YES.
- Set the SLP parameter in the FACILITY suboption to YES.

For example:

```
TSS MODIFY FACILITY(CICSPROD=XTRAN=YES,SLP=YES)
```

EJB Role Based Security

To enable Enterprise Java Bean (EJB) Role Based Security, you must set XEJB to YES in FACILITY suboption or in CICS SIT Table. For example:

```
TSS MODIFY FACILITY(CICSPROD=XEJB=YES)
```

To use EJB Role Prefixing:

- Specify EJBROLEPRFX=*16-byte-value* in the CICS SIT Table .

  or

- Specify EJBRPRFX=*16-byte-value* in the TSS FACILITY suboption.

For example:

```
TSS MODIFY FACILITY(CICSPROD=EJBRPRFX=CICSPROD)
```

Allocating and Usage of CICS Session Cache

CICS session cache is now allocated according to the CICSCACHE facility sub-option. Allowed resources encountered during the life of the cache will be placed into the cache so that later accesses to the same resource will not result in I/O to the security file. This cache was once controlled by OPTIONS(28), the user may now enter the following commands:

```
TSS MODIFY FACILITY(cics_facility=CICSCACHE({TASKLIFE},{NOAUDIT},{512})
                            {SESSLIFE}{ AUDIT }{1024} 2048}{4096}
```

The CICS cache is allocated in the CICS region DSA. Use of persistent large cache size may require alteration in the DSA allocation and in the CICS region size. The default is designed to minimize the effects on storage requirements.

Keywords for this control sub-option are defined as follows:

TASKLIFE

Indicates that the cache will be allocated at the beginning of a CICS task and freed at the end of the task. This is the default.

SESSLIFE

Indicates that the cache will be allocated at signon and will persist without being cleared until the user signs off implicitly or explicitly or until the user is REFRESHed.

NOAUDIT

Indicates that allowed resources will not be audited if they are present in the cache. This is the default.

AUDIT

Indicates that allowed resources will be audited even if they are present in the cache.

512|1024|2048|4096

Indicates the size (in bytes) of the cache to be allocated for users in this CICS facility. This parameter may be altered while a related CICS region is executing, but the change will not take effect until the region is recycled.

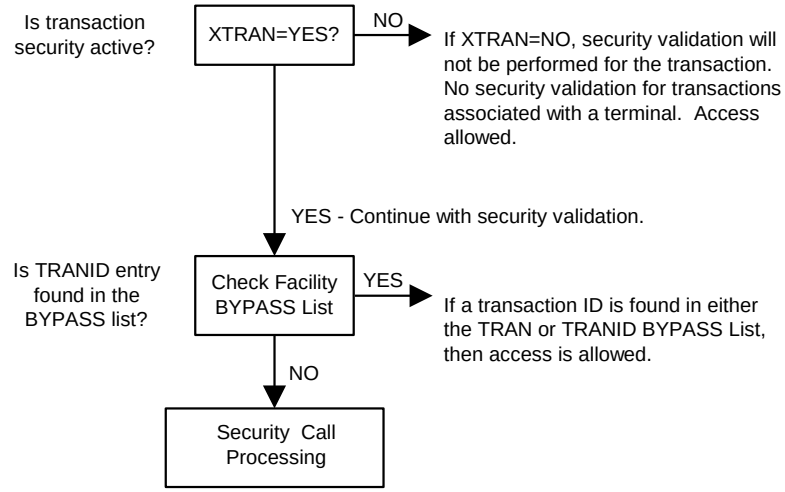
Using SESSLIFE can reduce I/O to the security file by allowing the product to check validity of previously validated resources in cache, rather than consulting the security file. Particularly for complex transactions with many resources access checks, SESSLIFE can improve the performance of CICS. On the other hand, changes to the security file, which normally would have prevented access, may be ignored for sessions, which overlap the changes.

Persistent SESSLIFE large caches for many simultaneous users may require changes to the DSA allocation and to the CICS Region size.

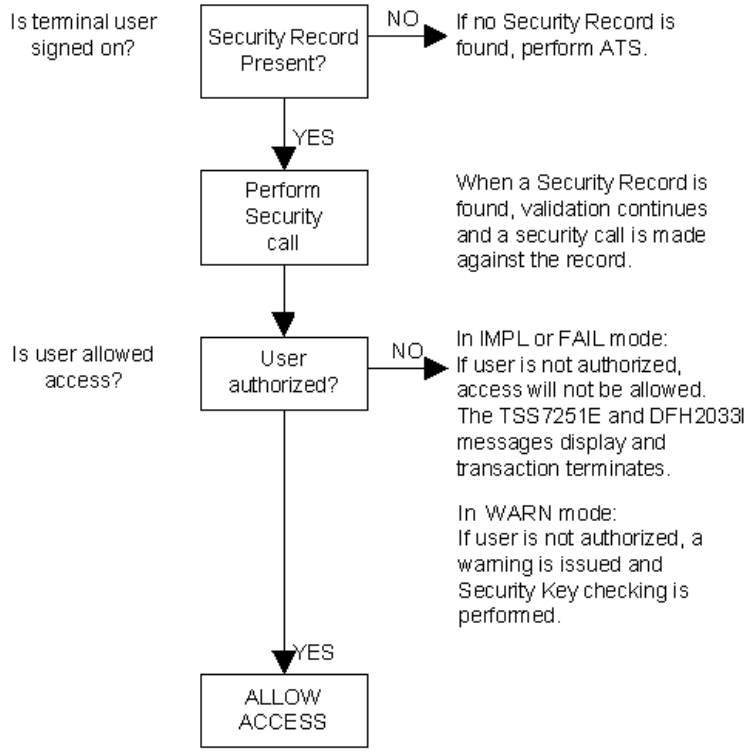
Using NOAUDIT can reduce I/O to SMF or ATF.

Transaction Validation

The following chart illustrates CA Top Secret CICS transaction validation logic.



The following chart illustrates CA Top Secret CICS security call processing:



Chapter 3: Security for a Multi-System Environment

This section contains the following topics:

Introduction

Security requirements for ISC or MRO are similar to the security requirements of a single, stand-alone CICS region. For background information about establishing ISC and MRO regions, see the IBM *Intercommunication Facilities Guide*.

You can define additional levels of security for both MRO and ISC environments. Details on how these levels relate to CA Top Secret security are described in the following sections. These levels are:

- Bind-time security
- Link security
- Attach-time security

Region Violations

In an MRO or ISC environment, region violations reported for CICS transactions result from both resource violations and failed signon attempts in the remote region. A TSSUTIL report of failed initiations can help you determine the cause of region violations for your system. For information about using TSSUTIL, see the *Report and Tracking Guide.*

Using RDO or RDM Parameters

To set up MRO and ISC environments, you must define specific CICS parameters. These parameters can be defined via one of the following:

- Resource Definition Online (RDO)
- Resource Definition Macro (RDM)

The following table shows the CICS parameters that you must define (via RDO or RDM) to set up MRO or ISC in your CICS region.

Definition	Using RDO
CONNECTION	ATTACHSEC
SESSION	USERID

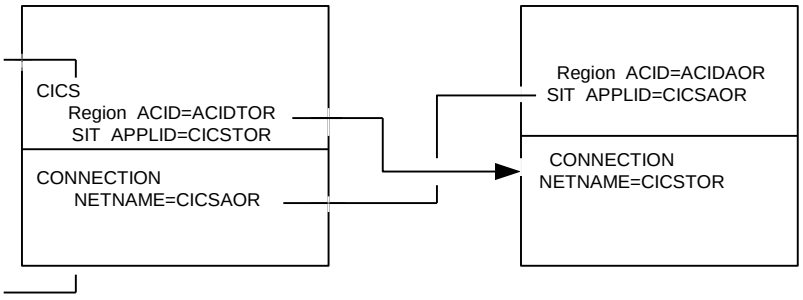
Defining Bind-Time Security

Bind-time security is used to prevent unauthorized remote regions from accessing your CICS region. A security check is performed when a request is made to establish a connection (bind) between two CICS regions. The bind process is accomplished in one of two ways:

- At CICS startup time if IRCSTRT=YES is specified in the DFHSIT.
- If the command, CEMT SET IRC OPEN is issued after CICS has completed its initialization.

For MRO Connections

In CICS, MRO bind-time security checks to see if the local region ACID has permission to bind with the remote APPLID. To determine this, CICS obtains the NETNAME value for each remote CONNECTION as illustrated below.



Before the bind, the local region ACID ACIDTOR is checked for the resource IBMFAC(DFHAPPL.cicsaor) and the remote region ACID is checked for IBMFAC(DFHAPPL.cicstor). In the TOR environment, this permission would be administered using the commands shown below.

```
TSS ADDTO(cicsdept) IBMFAC(DFHAPPL)

TSS PERMIT(acidtor) IBMFAC(DFHAPPL.cicsaor)
```

For the AOR, grant permission as shown below.

```
TSS PERMIT(acidaor) IBMFAC(DFHAPPL.cicstor)
```

For ISC Connections

External bind-time security for ISC is established by specifying BINDSECURITY(YES) in the CICS definition for the link. Each pair of communicating systems must have the same bind password for the link between them to be successful.

A bind password consists of up to 16 hexadecimal digits (0 through F), and can be surrounded by quotes. If you specify less then 16 digits, the bind password is padded on the right with hexadecimal zeros.

Defining ISC External Bindtime Security to CICS

The following figure shows how to define external bind-time security for ISC connections.

```
RDO definition

DEFINE
  CONNECTION(sysidnt)
  GROUP(groupname)
  ACCESSMETHOD(VTAM)
  NETNAME(name)
  PROTOCOL(APPC)
  SINGLESESS(N)
  SECURITYNAME(name)
  BINDSECURITY(YES)
```

For CICS, CA Top Secret lets you define:

```
TSS ADDTO(APPCLU) LINKID(netid.source-applid.target-applid)
                  SESSKEY(pass)
```

netid

The VTAM ACTSTRxx NETID value

source-applid

The local APPLID from the local region SIT

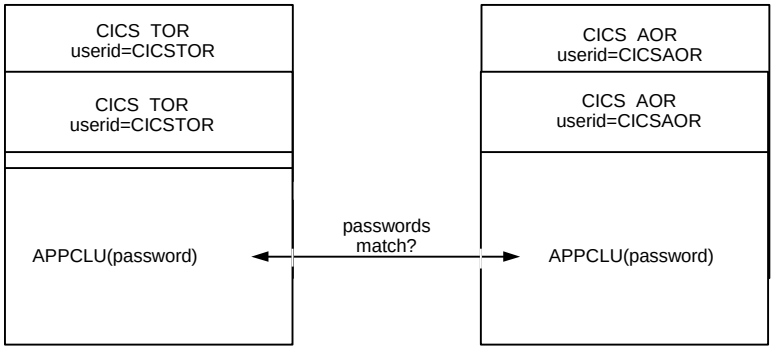
target-applid

The remote APPLID from the remote region SIT

pass

The 16-digit hexadecimal password

An example of how external bind-time security works for ISC connections is shown in the following figure:



Specifying a bind password causes CA Top Secret to perform password checking each time a session is bound. If the two bind passwords do not match, the session is not bound, and the system reacts to a user request for a session with SYSIDERR (an IBM CICS error message).

Defining Link Security

Link security limits a remote system's authorization to attach your transactions and access your resources. Each time a request is made to access a remote resource, a security check is performed against the userid defined in the session definition or the CICS TOR userid, if userid is omitted. Since security calls are being made against the link, the CICS region userid for the link must have permission to these resources, or have the NORESCHK and NOLCFCHK attributes defined to the ACID.

No signon for the link takes place if the requesting system passes a userid that matches the receiving CICS region's userid. Therefore, if you want to apply effective link security, the userid on one side of an MRO link must *not* match the userid on the other side.

It is suggested that MRO region ACIDs be set up with the following attributes:

- NOSUBCHK, NORESCHK, NOLCFCHK
- SOURCE(INTRDR)
- FACILITY(BATCH,STC)
- FACILITY(CICS regions connecting to)

Since these CICS region ACIDs are usually created without a password so that the operator does not have to enter the password when the region is started, the CICS region ACID might be compromised. To prevent a user signing on with the CICS region ACID as a user on a facility to which it is authorized, the SOURCE(INTRDR) restriction is recommended. The NORESCHK and NOLCFCHK attributes are necessary because of LINK SECURITY considerations. NOSUBCHK is necessary for correct handling of job submission.

For MRO and ISC

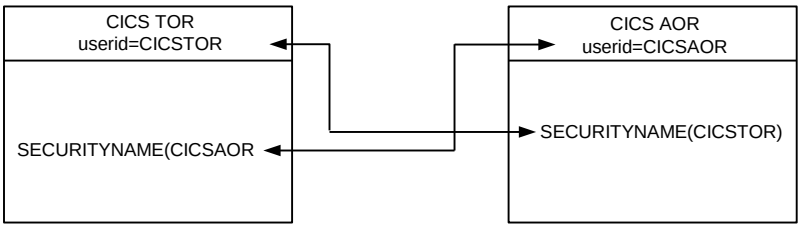
Link security works by signing on to each end of a session (via receive terminals) using the userid specified on the SESSION definition or, if omitted, the default user.

Defining Link Security to CICS

The following figure shows how to define link security for both MRO and ISC connections

```
RDO definition
   DEFINE
   SESSION
   CONNECTION(connect)
   USERID(userid)
```

An example of how link security works in MRO and ISC connections is shown in the following figure.



The SESSION ACID should not be identical across the link or the signon is not performed and the link will fail. The SESSION ACID provides a focal point to decide if individual remote transactions and resources are permitted to the local region. For the purpose of link security, the region's permission for remote resources is represented as the SESSION USERID. This user-acid can be modeled similar to the remote region ACID, but might have more stringent restrictions than the region ACID to prevent specific types of access by all users signed on to the region.

In the previous example, the PAY transaction is being routed to the AOR. Before the transaction is initiated, CA Top Secret issues a link security check against the CICS region ACID specified in the SECURITYNAME definition as CICSTOR. This security check determines if the PAY transaction can run in the AOR region.

The PAY transaction (depending on the mode) is not able to execute in the AOR region if the CICS SESSION ACID USER1:

- Cannot sign on to CICSAOR
- Is not permitted to the PAY transaction in the TOR and AOR environment
- Is not permitted access to PAY resources in the AOR

Link Security Considerations

Link security is checked at the time a remote transaction is about to be executed, or when a remote resource is about to be attached. Link security involves not only the CONNECTION (DFHTCT TYPE=SYSTEM), but also the SESSION definition (DFHTCT TYPE=).

```
RDO definition                RDM definition

DEFINE SESSION (    )        DFHTCT TYPE=  (    )
```

Link security limits the originating region's authorization to attach local transactions and to access local resources. The following CONNECTION considerations are important for link security:

- If the CONNECTION to the originating region does not specify *securityname*, extra security calls are made on every link request.
- Security calls are made to evaluate transactions and resources against the region ACID of the originating region, as well as the user ACID who originated the transaction. To grant permission for these, NORESCHK or NOLCFCHK should be specified on the originating region ACID, or explicit PERMIT commands are required. The originating region ACID must have access to the destination region's MASTFAC facility.
- The region ACIDs for two linked regions should never be the same. If two linked regions share the same region ACID, the *securityname* ACID will never be signed on during link security, causing link security to fail.

The following SESSION considerations are important for link security:

- The value OPERSECURITY(1) should be allowed to default.
- The value OPERRSL(0) should also be allowed to default.

Specifying other values will cause link security to avoid signon for the originating region ACID. Link security will then default to the destination region ACID with often paradoxical results.

Defining Attach-time Security

Attach-time security allows incoming requests to attach to requested transactions. The session must be established. In addition to the link security check, a second check is made on behalf of the signed-on user or the CICS region ACID, depending on the attach-security specification.

The level of attach-time security required for a remote system is specified in the ATTACHSEC parameter (for RDO) or the USERSEC parameter (for RDM), as shown in the following figure.

```
RDO definition                RDM definition
DEFINE                        DFHTCT TYPE=SYSTEM
CONNECTION(sysidnt)              ,SYSIDNT=name
GROUP(groupname)   .                                    .
ATTACHSEC({Local|             ,USERSEC=\{Local|
          Identify|            Identify|
          Verify|              Verify|
          Persistent|          Persistent|
          Mixidpe\}            Mixidpe\}
```

Attach Time Security Levels

There are five levels of attach-time security:

LOCAL

Any requests from the remote system are checked only for Link authority. Set this parameter if CA Top Secret is not securing the remote region. LOCAL is the default.

IDENTIFY

Any requests from the remote system are checked not only for link authority, but also for the user who initiated the request. Set this parameter if CA Top Secret is securing the remote region.

VERIFY

Every attach request requires a user identifier and a user password.

PERSISTENT

Requires a user identifier and user password with the first attach request for a new user. Any subsequent attach requests for the same user only requires a user identifier. The first attach signs the user on, even if the attach is not authorized to attach the transaction. Set this parameter if CA Top Secret is securing the destination region (LU6.2 only).

MIXIDPE

Specifies that the signon level for the remote user is determined by parameters sent with the attach request. The possibilities are: no signon, signon with password, signon without password. Set this parameter if CA Top Secret is securing the destination region ACID (LU6.2 only).

Note: You cannot specify VERIFY, PERSISTENT, or MIXIDPE on MRO links. These are LU6.2 (ISC) only.

Monitoring Type 71 RACF Event Notifications (ENF)

The ENF 71 function under z/OS enables communication between administrators and applications. Beginning with z/OS 1.11 and CICS 4.1, CICS monitors type 71 RACF ENF signals. CA Top Secret immediately sends an ENF signal to CICS when a security administrator makes the following changes:

- Suspends a signed-on remote user or a signed-on user who is not directly using a physical terminal or console
- Adds or removes the profile for a signed-on remote user or a signed-on user who is not directly using a physical terminal or console

  When the profile addition or removal occurs and CICS receives a new attach request for a user ID, CICS performs an implicit signon for the user ID and uses the new profile information. Existing tasks for the user continue with the profile that was valid when the task was attached.
- Deletes a signed-on remote user or a signed-on user who is not directly using a physical terminal or console

Note: CICS is *not* notified when a user ID expires.

The ENF signal *immediately* notifies CICS of the change to the user's security record (overriding any setting specified in the USRDELAY system initialization parameter). CICS can then refresh or remove the user's security record in a remote CICS region.

Example: Monitoring Type 71 RACF ENF Signals

In this example, an administrator issues the following command to suspend USER01 for five days:

```
TSS ADDTO(USER01) SUSPEND FOR(5)
```

Issuing the command immediately notifies CICS of the change.

Local Security Considerations

When ATTACHSEC(LOCAL) is specified for a connection, no individual user information is passed to the remote region; only link security is checked when the transaction processes. This setting should *not* be used for:

- Resources that require information about the individual local user to make security decisions about their use
- Authorized job submission
- TSSCAI calls

Remote Security Considerations

The attach-time parameters IDENTIFY, VERIFY, PERSISTENT, and MIXIDPE provide full remote security. This level of user security processing is the standard CICS security method of propagating the user's security information from one region to another in a CICS MRO or ISC environment. CICS transmits the userid of the signed-on user along with the remote request. When the remote request arrives in the AOR, CICS retrieves the userid and issues a signon request on behalf of the user.

Note the following information:

- Additional security file I/O occurs while processing these remote signon requests.
- If you alter the authority of a signed-on remote user, CICS continues to use the security values acquired at the previous remote signon until one of the following conditions occur:
  - A period of time (specified in the DFHSIT parameter USRDELAY) has elapsed since the previous attach request from this user.
  - Signon occurs with a new GROUP entry.
  - The link to the remote CICS region has been broken.
  - The administrator performs a TSS REFRESH in the remote region address space.
  - The CICS system has been recycled.
  - An issued command suspends a signed-on ACID, adds a profile to a signed-on ACID, removes a profile for a signed-on ACID, or deletes a signed-on ACID, triggering a type 71 RACF ENF signal that notifies CICS of a change to the ACID's security record.
- Some CICS releases use SYSIDNT (as defined in the SIT) to run transactions in connected regions. When this is true, the SYSIDNT must be defined as an ACID and permitted to the facility of the connected region.

  Note: To determine if you must allow for this situation, see the *CICS Interregion Communication Guide* for your appropriate CICS release.

  You should code this ACID with a non-expiring password. The following example shows that no permission is needed just to add the facility with which the SYSID is associated:

```
TSS CREATE('SYSID') NAME('CICS SYSID ACID')
                    FACILITY(CICS)
                    PASSWORD(XXXX,0)
                    DEPARTMENT(deptacid)
```

Chapter 4: Implementing Security

This section contains the following topics:

Day to Day Operations

The day-to-day operations you need to administer your CICS system in a secured environment include the following:

- Overseeing CICS signon and password processing
- Choosing and administering LCF or OTRAN (resource OTRAN) security
- Administering resource level security
- Administering terminal security
- Administering SPI resources
- Protecting job submission

Note: The PassTicket feature for signing on to a host system is available for CICS. See the *User Guide for* details on PassTicket.

Signing On to CICS Under CA Top Secret

Signon/signoff procedures are different for each site, so it is recommended that your security administrator provide the user community with any operating system signon/signoff requirements and the CA Top Secret procedures discussed in this section.

Signing On Using CESN

You can sign on to CA Top Secret CICS through the IBM-supplied CESN transaction. The CESN transaction is used to sign on an up to eight-character alphanumeric userid. This userid should be the same as the CA Top Secret ACID defined for the user.

The CESN signon procedure can be executed through screen prompts or by stringing the commands together.

Signing On By Command String

Use the following syntax to sign on to the CESN transaction. Note that the password is displayed.

```
CESN USERID=name,PS=password,NEWPW=newpassword
```

Make the appropriate entries where:

USERID=

The alphanumeric userid or the defined CA Top Secret user ACID.

Size: Up to eight characters

GROUPID

(Optional) The group name. Before the advent of TSS REFRESH, signing on with a different GROUPID was a way of refreshing the characteristics of an ACID in a related AOR, without having to wait for the USRDELAY timeout. The GROUPID parameter might still be used for this purpose in MRO environments. Validation of the CESN GROUPID is described in the *User Guide*.

Size: Up to eight characters

LANGUAGE

(Optional) Has no security significance.

PS=

The password associated with the user ACID.

Size: Up to eight characters

NEWPS=

The new password replacing your lost, expired, or existing password.

Signing On By Screen Prompt

At most sites, the signon screen is automatically displayed. If it is not, you can sign on to CICS using CESN through screen prompts. See the IBM *CICS-Supplied Transactions* Guide for details on CESN signon procedures.

Make the appropriate entries in each field where:

USERID

The eight-character alphanumeric userid or the defined CA Top Secret user ACID.

Size: Up to eight characters

GROUPID=

(Optional) The group name. Before the advent of TSS REFRESH, signing on with a different GROUPID was a way of refreshing the characteristics of an ACID in a related AOR, without having to wait for the USRDELAY timeout. The GROUPID parameter might still be used for this purpose in MRO environments. Validation of the CESN GROUPID is described in the *User Guide*.

Size: Up to eight characters

LANGUAGE

(Optional) Has no security significance.

PASSWORD

The password associated with the user ACID.

Size: Up to eight characters

NEWPASSWORD

The new password associated with the user ACID which replaces a lost, expired, or existing password.

Size: Up to eight characters

The standard signon procedure using the CESN screen prompts is:

- Enter your CA Top Secret ACID (maximum eight-character alphanumeric) in the USERID: field.
- Enter your selected password (maximum eight-character alphanumeric). The characters in the password field will not display.
- Press Enter.
- When signon is successful, this message is displayed:

```
TSS7000I acidname Last-Used mm/dd/yy hh:mm System=xxxx Facility=xxxxxxxx
TSS7001I Count=xxxxx Mode=xxxx Locktime=xxxxx Name=xxxxxxxxxxxxxxxxxxxx
```

Note: In most cases, your security administrator will set up your password to expire the first time it is entered.

Signing on Using CESL

Use CESL to sign on to CICS using a password phrase as authorization.

With CESL, you can sign on to CICS with a password phrase of 9 to100 characters and a standard password of up to eight characters. If you enter a password that is between 9 and 100 characters, CESL treats the password as a password phrase. In other respects CESL operates the same as the CESN signon transaction.

National Language Support for CTS (CICS)

Primary and secondary language codes are added to an acid with the USERNL1 and USERNL2 fields. CA Top Secret supports the same language codes as CTS (CICS).

The rules for assigning language code are:

- If the user has a supported language code defined in USERNL1 or USERNL2, the code is assigned.
- If the USERNL1 or USERNL2 codes are not defined, the code on the CICS Default User is used.
- A language code entered on the signon screen overrides USERNL1 or USERNL2 on the user.
- The CICS Default Language code assigned to the CICS Default User is the first value in the NATLANG SIT parm.
- If the NATLANG SIT parm is not present it defaults to ENU.

Automatic Terminal Signon Procedure

Automatic Terminal Signon can be used for terminals from which an explicit signon is not possible or desirable. Automatic Terminal Signon is involved whenever a protected transaction is entered from a terminal for which no explicit signon has been performed. When this occurs, CA Top Secret searches its security file for an ACID that matches the terminal name. If the ACID is not found, the transaction is failed, and you will receive message DFH3510, requesting you to sign on. If the ACID is found, then all of the normal security checking associated with this ACID is performed (with the exception of password checking).

If the automatic signon is successful, the ACID is associated with that terminal for that session, just as if an explicit signon had been performed. Processing of the intended transactions are initiated.

The ACID name generated is:

VTAM

Eight-character netname.

TCAM

Eight-character TCAM terminal name.

BTAM

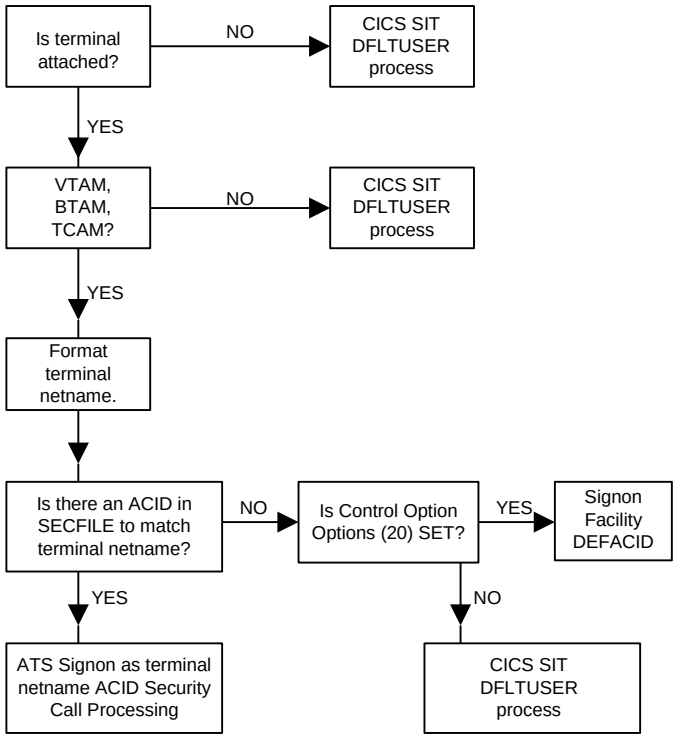
Four-character terminal name.

Your installation selects which terminals are valid for Automatic Terminal Signon by defining an ACID for those terminals. Since these ACIDs are (in CA Top Secret terms) normal user ACIDs, security administration for these ACIDs is no different than other user ACIDs. The ACID should also be given a SOURCE that matches the terminal name, thereby preventing the ACID from being used from any other terminal.

For example, using a VTAM terminal whose netname is K067T018:

```
TSS CREATE(K067T018) NAME('EMAIL SYSTEM GR 1')
                     FACILITY(CICSPROD)
                     DEPARTMENT(CIPCC)
                     PASSWORD(NOPW,0)
                     SOURCE(K067T018)
```

The following illustrates CA Top Secret CICS Automatic Terminal (ATS) processing.



ATS is not performed if:

- Validation is not required for a transaction being entered at a terminal
- XTRAN=NO
- The transaction is in the Bypass List

The following OPTIONS control options maybe set to affect the processing of automatic terminal signons:

OPTIONS(20)

OPTIONS(20) is an anachronism from releases of CICS before CICS 3.2, when the CICS SIT DFLTUSER did not exist. We recommend that the administrator should default to the CICS SIT DFLTUSER, because this involves no security overhead. If the administrator desires a separate CICS DEFACID for audit purposes (for users who cannot type their ID), the DEFACID should be a different ACID from that supplied for the SIT DFLTUSER.

If there is no ACID in the security file that matches the terminal ID, OPTION(20) will attempt to sign on as the MASTFAC facility DEFACID. If the DEFACID is incapable of signing onto the CICS region, CICS will apply the SIT DFLTUSER.

OPTIONS(30)

Update the last-used statistics of an ACID employed for ATS. If this option is turned off, last-used statistics are not updated in the security file.

Signon Initiated Transactions

You can define transactions so that they automatically initiate when you sign on. This helps you to maintain procedures, as well as enables post-signon processing.

For example, with the command shown below, CA Top Secret starts the transaction as soon as the signon messages are cleared (after a few seconds). This transaction runs under the ACID that just signed on, so make sure the ACID has the required signon permissions.

```
TSS ADDTO(user) SITRAN(trans[,facility])
```

CA Top Secret initiates the SITRAN transaction with an EXEC CICS START command. CICS Dynamic Transaction Routing does not act on transactions started in this manner.

Note: If a transaction running attached to a terminal is invoked via EXEC CICS START, the Automatic Terminal Signon (ATS) is executed using the ACID of the user invoking the transaction. The ACID is associated with the terminal until the transaction ends, then the ATS is automatically signed off.

Administering Passwords

Because an extensive variety of password controls is available, you should develop password usage strategies particular to your site.

Here are a few guidelines you can follow for preserving password integrity:

- Memorize your password.
- All written records of your password should be destroyed.
- Do not post your ACID or password near the video terminal, disks, cabinets, bulletin boards, or other areas accessible to unauthorized individuals.
- Do not maintain your password in an unprotected data set where others could view it.
- Do not share your ACID or password with anyone. Personnel requesting the use of another's ACID or password should be directed to the appropriate security administrator.
- Inform your security administrator immediately if you suspect that your ACID or password have been compromised and request a password change.

Note: Keep in mind the CA Top Secret control options that manage password operation: NEWPW, RNDPW, HPBPW, INACTIVE, PTHRESH, and RPW.

When choosing a new CICS password or changing an existing one, at least three of the characters must be different from your previous password.

Change a Password or Password Phrase

You can change your password using a CESN or CESL transaction using screen prompts. Change your password regularly to help ensure system security.

Note: You can enter a password phrase only with the CESL transaction.

Follow these steps:

1. Type your CA Top Secret ACID in the USERID: field.
2. Type your existing password or password phrase.
3. Type your new password or password phrase in the NEWPASSWORD: field and press Enter.

   Note: A password can be a maximum of eight alphanumeric characters. A password phrase can be a maximum of 100 alphanumeric characters.
4. Retype your new password or password phrase and press Enter.
5. (Optional) if the NPWR FACILITY suboption is in effect you are prompted to verify your new password or passphrase again.

   Your password is changed.

Random Password Generation

Use random password generation to have CA Top Secret automatically assign a password for you (except if you are signing on for the first time).

To use a random password

1. Enter your CA Top Secret ACID in the USERID: field.
2. Enter your existing password (maximum eight-character word composed of numbers, letters, and/or national characters).
3. In the NEWPASSWORD: field, type **random.** Press the Enter key.

   The Password Changed messages are displayed.
4. Press the Enter key again.

   The New Password messages are displayed.
5. Memorize the password generated for you (indicated above as xxxxxxxx.) Without this password you are not able to sign on again.
6. Press the Enter key.

   These messages are displayed:

   ```
   TSS7000I acidname Last-Used mm/dd/yy hh:mm System=xxxx Facility=xxxxxxxx
   TSS7001I Count=xxxxx Mode=xxxx Locktime=xxxxx Name=xxxxxxxxxxxxxxxxxxxx
   ```

Note: Random password generation can be made mandatory across the site using the NEWPW and RN control options. When these control options are specified, expired passwords automatically generate random passwords without user request.

Password Expiration

Your CA Top Secret password expires automatically after a set amount of time. Approximately five days before the password expiration date, the following message is displayed each time you sign on to a facility:

```
TSS7003 Password Will Expire Soon on mm/dd/yy
```

When your password expires, this message is displayed:

```
TSS7110I Password Has Expired. New Password Missing.
```

If your password has expired, use the procedure for changing a password to assign a new one.

Lost Passwords

If you forget your password, you cannot access a facility. Do not try to guess your password. Notify the appropriate security administrator immediately to have a new password assigned to you.

Administering Transaction Security

CA Top Secret secures CICS transactions in two ways: using the Limited Command Facility (LCF) or using OTRAN (resource) security.

OTRAN Security

The OTRAN resource name is shared by all CICS, CA-IDMS®, and IMS facilities. Therefore, protecting a transaction via OTRAN for a CICS region also results in transactions of the same name being protected in all CICS, Advantage™ CA-IDMS®, and IMS regions that are also under the control of CA Top Secret.

Note: A transaction protected using OTRAN will ignore LCF security.

To add ownership of a transaction to an ACID, enter:

```
TSS ADDTO(dept) OTRAN(transaction)
```

To permit access for a user-acid to the protected transaction, issue a command like the one shown below.

```
TSS PERMIT(acid) OTRAN(transaction)
```

In CA Top Secret, OTRAN security can also provide password reverification. See the *User Guide* for a detailed discussion on OTRAN security.

LCF Security

If you choose not to protect transactions using OTRAN, they can be protected via LCF. Transactions protected through LCF must be defined by facility. Transactions should be defined inclusively (TRANS) or exclusively (XTRANS), but not both. Essentially, each user can have an inclusive list, which specifies a list of transactions the user is allowed, or an exclusive list, which the user is not allowed to use.

Password reverification can be provided by LCF:

```
TSS ADDTO(acid) TRANSACTIONS(CICSPROD,(PAY9(V))
```

It is recommended that transactions be divided by function or subset and defined as a group within profiles. This way transactions are defined only once per group, instead of once per user.

Using the NOXDEF and XDEF Suboptions

The NOXDEF FACILITY suboption is set by default in CICS facilities to allow all users to access any CICS transaction until access to the transaction is restricted via LCF. To provide default protection of transactions, set the XDEF FACILITY suboption. The XDEF suboption indicates that users must be authorized to use transactions explicitly.

Transactions can be authorized through LCF, or OTRAN. See the *User Guide* for details.

Administering Resource Level Security

For instructions on how to secure resources, see the *User Guide*.

Administering Record Level Protection (RLP)

This section explains how to implement Record Level Protection (RLP). RLP gives you detailed control over which users have access to what data within your system. This access is controlled by defining the records you want to protect to a reserved ACID called the Static Data Table (SDT) Record, and then permitting access to the defined records using the TSS PERMIT command.

Before you can implement RLP, you must first initialize the SDT using the SDTBLOCKS parameter of TSSMAINT. You also need to extend your old Security File into your new Security File by using TSSXTEND.

Protecting Records and Fields

Using RLP, you can give users access to a set of records within a file, instead of all of the records in a file. You can even take this protection one step further by giving users access to a set of fields within a record, instead of all of the fields within a record.

The SDT contains three record elements that are used to implement RLP:

RECORD

Defines the record using its FCT name, and specifies the record's field layout (field name, data type, field positions, length). The field(s) defined are then referenced in the SELECT record. You only need to define the fields that participate in the selection process.

SELECT

Defines the logic, using Boolean expressions, that specifies who gets access to a record based on the contents of one or more fields.

MASKREC

Defines which fields within a record cannot be accessed (optional).

Administering Screen Level Protection (SLP)

To secure terminal screen input data, the following implementation steps must be taken:

- Initialize the SDT, if it has not already been done.
- Decide which transactions or programs you want to protect using SLP. In particular, decide which screens would benefit from limiting the range allowed for certain users in specified fields. Decide if combinations of data could or should be prevented from use by identifiable groups of users.
- Decide the SELECTion criteria to be used to determine access for all programs in the protected application.
- Define the MAPREC fields needed to support the aggregate SELECT criteria.

See the *User Guide* for more details about the SDT and SLP requirements.

Administering Terminal Security

The following sections explain how to administer terminal security.

Using Preset Terminal Security

You can preset terminal security by permanently associating any ACID with a particular terminal. When the preset terminal is connected to CICS, CICS (as an authorized user) signs the ACID on- bypassing password processing. Since no password is required, use of this feature is secured.

To install a terminal definition using preset security, the terminal operator must have access to the name of the preset USERID in the resource class SURROGAT. For example, to install a terminal with a preset USERID of WAREHOUS, you must first define an ACID called WAREHOUS to CA Top Secret with permission to the CICS facility and any appropriate resources. Then, the user defining the terminal must have access to the resource WAREHOUS.DFHINSTAL in RESCLASS(SURROGAT). Sample statements illustrating these two steps appear next.

```
TSS ADDTO(CICSDEPT) SURROGAT(WAREHOUS.DFHINSTAL)
TSS PERMIT(CICSADM) SURROGAT(WAREHOUS.DFHINSTAL) ACCESS(READ)
```

Finally, the CICS facility must run with the RES attribute, so that the permissions in the SURROGAT resource class can be stored when the user signs on.

Restricting Terminal Access

CA Top Secret restricts selected VTAM, TCAM, and BTAM terminals from the use of unauthorized users. By defining a terminal or terminal prefix/node to CA Top Secret, and giving ownership to a user or group of users, only those people given permission to use a terminal can access CICS via that terminal. Any other user ACIDs attempting to use these terminals are logged off after signon.

Securing Sequential Terminals

Full security is enforced for transactions entered from a sequential terminal. To set up security for a sequential terminal, create an ACID with the same name as the sequential terminal, and let the CA Top Secret Automatic Terminal Signon procedure associate the ACID with the terminal.

Note: A CESN signon transaction can be specified. However, this is not recommended since the password would also have to be specified in the data set.

Securing z/OS Console Terminals

Full security is enforced for z/OS console terminals. Since an explicit signon is not appropriate for z/OS consoles, it is recommended that an ACID (or ACIDs, one for each console) that matches the four-character CICS terminal ID be created. This allows the CA Top Secret Automatic Terminal Signon procedure to associate the ACID with the z/OS console terminal. It is recommended that an inclusive transaction list containing CEMT (and/or CSMT) be ADDed to this ACID for each CICS facility, as shown below:

```
TSS ADDTO(xxxx) TRANSACTION(fac,(CEMT))
```

This prevents a z/OS operator from entering sensitive transactions.

Terminal Locking Security

CA Top Secret provides TSS commands and a control option that allows the security administrator and individual users to control when inactive or unattended terminals are locked. The standard locktime procedure is:

- When you signon to a terminal, CA Top Secret begins to monitor LOCKTIME thresholds. If you do not signon and are assigned to the SIT DFLTUSER, no LOCKTIME monitoring will occur.
- When the LOCKTIME threshold is expired, at the next action key (Enter, Clear, PF key, and so on) the terminal screen is cleared and you are prompted for your password.
- You can enter your password, CESF, or press the Clear key.
- Use the LTLOGOFF FACILITY suboption to further enhance LOCKTIME processing. When you set LTLOGOFF=YES, if the LOCKTIME expires again before your enter your password, the terminal is signed off security and logged off.
- Use LTLOGOFF=SIGNOFF to sign off the terminal's user without disconnecting the terminal from CICS.

Use these methods to control terminal lock time:

- Use the LTIME parameter with the TSS ADD command to allow the security administrator to set terminal lock times for individual users.
- The TSS LOCK/UNLOCK commands allow users to lock and unlock their terminals.
- The LOCKTIME suboption of the FACILITY control option allows CA Top Secret security administrators to set lock times for all terminals connected to a specific facility.

LOCKTIME processing occurs only on the CICS region where the user is physically signed on. For example, in an MRO environment, AOR processing is not affected. Terminals and ACIDs can be exempted from LOCKTIME processing by placing them in the LOCKTIME Bypass List.

Using OPTIME Security

To support CICS, an OPTIME field matches the TIMEOUT field described in IBM CICS documentation. The syntax follows.

```
TSS ADDTO(acid) OPTIME(hhmm)
```

OPTIME controls the period of time allowed before CICS considers a terminal user to be "timed-out." The action taken by CICS depends on a CICS parameter, SIGNOFF, which is specified in the TYPETERM definition. See the IBM documentation for more information about the use and setting of these fields.

CA Top Secret provides the OPTIME field to support IBM CICS TIMEOUT functionality. When the user signs on, CA Top Secret places the OPTIME value into the user's TCTTE. CICS then scans the idle time for each terminal. When OPTIME/TIMEOUT is exceeded, CICS will take action based on the TYPETERM SIGNOFF parameter associated with the user's terminal.

Adding a non-zero OPTIME to DFLTUSER gives a default value for OPTIME to any user who does not have a non-zero OPTIME already assigned.

Administering Transient Data Security

You can designate an ACID to be associated with transactions initiated by an intra-partition transient data queue with a trigger level. The value might come from one of the following sources:

- If the userid is specified on the TYPE=INITIAL or TYPE=INTRA macro, it is signed on and used for security.
- If the destination is associated with a terminal (DESTFAC=TERMINAL), the userid is derived from the terminal. If no one is signed on and Automatic Terminal Signon is not in effect, the userid will result in the CICS DFLTUSER being used.
- If the QUEUE specifies DESTFAC=SYSTEM then the link userid on the connection definition is used.

Administering Job Submission

Transactions in CICS can be submitted through one of two mechanisms:

- The SPOOLOPEN, SPOOLWRITE, SPOOLCLOSE mechanism
- Extra-partition destination DCT

Two acids are involved in job submission: the submission originating user (the user signed on to CICS who initiates the job submission request); and the region acid (the user who manages the interface to JES on behalf of the originating user).

When either of the two mechanisms is used, CA Top Secret analyzes the JOB card USER parameter to determine if the originating user and the region acid are cross-authorized to submit the job. Such permission must be granted to the originating user explicitly as follows:

```
TSS PERMIT(subacid) ACID(jobacid)
```

The region acid can have explicit permission granted through a similar explicit command, or the administrator can grant global submission to the region acid on behalf of any acid through the following:

```
TSS ADDTO(regacid) NOSUBCHK
```

Granting global job submission to the region ACID is an administrative advantage, except for the loss of explicit control of job submission by the region acid. The advantage of granting explicit permission is precisely in maintaining this extra level of control. CA Top Secret leaves The choice of strategy is left up to the individual administrator.

When using these methods of job submission, the USER and PASSWORD parameters on the JOB statement might be omitted. If present, the USER and PASSWORD parameters are checked as written; if omitted, CA Top Secret will add the USER and PASSWORD from the currently active acid at the time that the EXEC WRITEQ or EXEC SPOOLWRITE commands (respectively) were issued. If no active user can be determined, the region acid is used as the active acid.

Note: If you do not wish to allow the region acid to be used as a default USER for job submission, then CICS must be run as a started task and the region acid should not be allowed to the BATCH facility.

The associated DDNAME for the DCT must be in the form:

```
//ddname  DD SYSOUT=(class,INTRDR)
```

For additional information on defining DCT extra-partition destinations, consult the appropriate IBM documentation. In this guide, additional information on DCT security is found in the section, Bypassing Security for Specific Resources, in the chapter, “Control Option Requirements,” and the section, Administering Resource Level Security, in the chapter, “Implementing Security.”

Bypassing SPOOLWRITE Job Submission Protection

In CICS, there is an SPI check to determine if the user has permission to open the SPOOL data set. To bypass this check, issue the following command:

```
F TSS,FACILITY(cicsprod=BYPADD(SPI=JESSPOOL))
```

If you do not want to place the user ID on the job card that is being submitted through SPOOLWRITE, issue the following command:

```
F TSS,FACILITY(cicsprod=BYPADD(SPI=SPOOLSUB))
```

If you do not wish to grant facility-wide access to job submission, these SPI resources might also be protected and controlled through TSS ADD and TSS PERMIT commands.

For example, the following command protects the use of the SPOOL data set:

```
TSS ADDTO(deptacid) SPI(JESSPOOL)
```

The command shown next restricts an individual from job submission except on FRIDAYs:

```
TSS PERMIT(user) SPI(JESSPOOL) FRI
```

Implementing RLP

There are four processes to Implementing RLP.

Task-Gather Information

Before you can define record elements, there are several preliminary steps you must perform. These steps are important, since the information you gather here will determine how smoothly RLP is implemented.

- Determine which of your applications would benefit from RLP.
- Meet with the programmers to gather information about the application (like FCT name, field names, positions, data types, length of field, and selection criteria).
- Become familiar with the application.
- Plan the details needed to implement RLP for this application For example, you might decide on a selection criteria that limits the user to viewing only the records within their departmental scope.
- Determine who is the administrator(s) for implementing RLP and give them MISC3(SDT) authority.

Task-Enter Definitions

All definitions are entered using the TSS ADDTO(SDT) command. The process is:

- Define the fields in the RECORD definitions needed for all SELECT statements. You do not need to define fields to CA Top Secret that are not needed by a SELECT statement. A sample RECORD definition is:

  ```
  TSS ADDTO(SDT) RECORD(pfile)
                 RECDATA(dept,char,10,4)
  ```

  CA Top Secret must know the layout of the record the user wants to access with such information as:

  - What is the name of the record (FCT name)?
  - What are the fields of the record that are used (referenced) in the selection process.
  - What is the format of the data in the fields?
  - What sizes are the fields?

  Note: If you are employing multiple fields within one record for your SELECT logic, you must do a separate ADD for each field you want to validate. You can define up to 10 fields for one record.

- Define the SELECT expressions to the SDT that you are using on the PERMIT command. A sample definition is:

  ```
  TSS ADDTO(SDT) SELECT(dp1000)
                 SELDATA('IF dept GE “1000” AND dept LE “1099”)
  ```

  After you have defined the layout of the record, you must define the following as part of the SELECT record:

  - What field of the record do you want CA Top Secret to validate?
  - What type of comparison should be made?
  - To what is the field being compared?

- Define any MASK records to the SDT. MASK records are optional, and identify which fields within a record cannot be accessed. A sample definition is:

  ```
  TSS ADDTO(SDT) MASKREC(mdept)
                 MASKDATA(pay,packed,30,4,0000)
  ```

- Check your work by listing the SDT records you just created. To list all records, use the command:

  ```
  TSS LIST(SDT) RECORD(ALL)
  TSS LIST(SDT) SELECT(ALL)
  TSS LIST(SDT) MASKREC(ALL)
  ```

- To correct any errors, first use the TSS REMOVE(SDT) command to remove any field you wish to modify, then use the TSS ADDTO(SDT) command to add the field you want to replace.

- When you are satisfied that everything is correct, refresh the SDT in-core tables using the command:
  ```
  TSS MODIFY(SDTTABLE)
  ```

Task-Permit Access to the Defined Records

When your definitions are complete, you are ready to permit access to the defined records.

1. First you must revoke any existing PERMITs that a user might have for these FCTs.
2. Then, re-PERMIT the FCTs using the SELECT and/or MASKREC clauses. A sample PERMIT command is shown next.

```
TSS PERMIT(jane) FCT(pfile)
                 ACCESS(READ)
                 SELECT(dp1000)
                 MASKREC(mdept)
```

Task-Enable Protection

After your definitions and permissions are complete, you must enable RLP for the facility. (The definitions and permissions will not take effect until RLP is enabled.) To enable RLP in the CICS region, enter:

```
TSS MODIFY FACILITY(cicsprod=RLP=YES)
```

Note: The CICS region must be restarted for the RLP to take affect.

Special Considerations

- For access level of BROWSE, only the records the user is allowed are returned. (Any records the user is not allowed are automatically bypassed). No violations or logging occurs for records *not* allowed by the RLP selection process.
- Even if XFCT=YES, and RLP=YES, the FCT in question MUST be owned and permitted to the user with the SELECT clause, before RLP will have any affect. For example:

```
TSS PERMIT(userid) FCT(FILEA)
                   ACCESS(READ)
                   SELECT(ISFILEA)
```

- For access level of DELETE to function under the RLP selection criteria, you must have the FCT defined (to CICS) with a journalling option or recovery enabled. For example, under CICS CEDA transaction:

```
CEDA def File(FILEB)
            .
            .
        RECOVERY PARAMETERS
        Recovery    : ALL
        Fwdrecovlog: 01
```

  Regardless of the FCT having journalling or not, normal access checks will still occur for DELETE access.

  If you are using the MASK feature of RLP, be aware that although masking is limited to READ and BROWSE file operations, the application should not WRITE(CREATE) a record from the data buffer that might contain the masking values.
- Make certain that the data types specified in the MASKDATA or RECDATA definitions match the data types of those contained on the actual file record.
- When RLP=YES, NOTAUTH conditions might have EIBRESP2=0000 instead of the expected EIBRESP2=101.

Administering CICS Command Security

CA Top Secret provides the SPI resource for added security checking.

Securing CEMT Commands

To obtain the security features in the following sections, you must ensure that the transaction CEMT has the PCT/RDO parameter RESSEC=NO. It is not necessary to separately secure the CEMT transaction through LCF or OTRAN resource checks. Instead, CEMT is secured in CA Top Secret mainly through a special SPI (Set, Perform, Inquire) resource class. Individual SPI resources are constructed from CEMT "keywords" to control the "action" in a CEMT command.

The table, *SPI Access Levels for CEMT*, shows the CA Top Secret ACCESS level required to execute "action" verbs in the CEMT syntax shown below.

```
CEMT action.keyword [(resource-name)] [keyword-operand value]
```

The table, *SPI Resource Keywords,* shows the correspondence between CEMT keywords and CA Top Secret SPI resource names. Because some actions in CEMT generate displays of individual resources, and allow the alteration of those resources displayed on the screen, CA Top Secret performs individual resource checks for certain resources, which are summarized in the table, *CEMT Secondary Resource Checks*.

The following table lists valid SPI access levels for CEMT commands:

CEMT Action	SPI Access Level
INQUIRE	INQUIRE
PERFORM	PERFORM
SET	SET
DISCARD	DISCARD

CEMT commands have keywords relating to a specific set of actions. The next section describes how CA Top Secret secures each keyword and their associated action.

Securing INQUIRE and SET Commands

The following table lists the CEMT command keywords and their associated SPI resource names:

Command Keyword	SPI Keyword
'Blanks' (default)	SPI(SYSTEM)
ATOMSERVICE	SPI(ATOMSERV)
AUTINSTMODEL	SPI(AUTINSTM)
AUTOINSTALL	SPI(AUTOINST)

Command Keyword	SPI Keyword
AUXTRACE	SPI(TRACEDES)
BEAN	SPI(BEAN)
BRFACILITY	SPI(BRFACILI)
BUNDLE	SPI(BUNDLE)
CAPTURESPEC	SPEC(CAPTURES)
CFDTPOOL	SPI(CFDTPOOL)
CLASSCACHE	SPI(CLASSCAC)
CONNECTION	SPI(CONNECTI)
CORBASERVER	SPI(CORBASER)
DB2CONN	SPI(DB2CONN)
DB2ENTRY	SPI(DB2ENTRY)
DB2TRAN	SPI(DB2TRAN)
DELETSHIPPED	SPI(DELETSHI)
DELTSHIPPED	SPI(DELTSHIP)
DISPATCHER	SPI(DISPATCH)
DJAR	SPI(DJAR)
DLIDATABASE	SPI(DLIDATAB)
DOCTEMPLATE	SPI(DOCTEMPL)
DSA	SPI(SYSTEM)
DSNAME	SPI(DSNAME)
DUMP	SPI(DUMP)
DUMPDS	SPI(DUMPDS)
EPADAPTER	SPI(EPADAPTE)
ENQ	SPI(UOWENQ)
ENQMODEL	SPI(ENQMODEL)
EVENTBINDING	SPI(EVENTBIN)
EVENTPROCESS	SPI(EVENTPRO)
EXCI	SPI(EXCI)
FECONNECTION	SPI(FEPIRESO)
FENODE	SPI(FEPIRESO)

Command Keyword	SPI Keyword
FEPOOL	SPI(FEPIRESO)
FEPROPSET	SPI(FEPIRESO)
FETARGET	SPI(FEPIRESO)
FILE	SPI(FILE)
GTFTRACE	SPI(TRACEDES)
HOST	SPI(HOST)
INTTRACE	SPI(TRACEDES)
IPCONN	SPI(IPCONN)
IRBATCH	SPI(IRBATCH)
IRC	SPI(IRC)
JMODEL	SPI(JMODEL)
JOURNALNAME/JOURNALNUM **Note:** JOURNALNAME is used for CTS 1.2 and above; JOURNALNUM is used for CICS 4.1 and CTS 1.1.	SPI(JOURNAL)
JVM **Note:** For CICS CTS 2.3 and above.	SPI(JVM)
JVMPOOL	SPI(JVMPOOL)
JVMSERVER	SPI(JVMSERVE)
LIBRARY	SPI(LIBRARY)
LINE	SPI(LINE)
LSRPOOL	SPI(LSRPOOL)
MAPSET	SPI(MAPSET)
MODENAME	SPI(MODENAME)
MONITOR	SPI(MONITOR)
MQCONN	SPI(MQCONN)
MQINI	SPI(MQINI)
NETNAME	SPI(TERMINAL)
PARTNER	SPI(PARTNER)
PARTITIONSET	SPI(PARTITIO)
PIPELINE	SPI(PIPELINE)
PITRACE	SPI(PITRACE)

Command Keyword	SPI Keyword
PROCESSTYPE	SPI(PROCESST)
PROFILE	SPI(PROFILE)
PROGRAM	SPI(PROGRAM)
REQUESTMODEL	SPI(REQUESTM)
RRMS	SPI(RRMS)
SESSIONS	SPI(SESSIONS)
STATISTICS	SPI(STATISTI)
STORAGE	SPI(STORAGE)
STREAMNAME	SPI(STREAMNA)
SUBPOOL	SPI(SUBPOOL)
SYSDUMPCODE	SPI(SYSDUMPC)
SYSTEM	SPI(SYSTEM)
TASK	SPI(TASK)
TCLASS	SPI(TCLASS)
TCPIP	SPI(TCPIP)
TCPIPSERVICE	SPI(TCPIPSER)
TDQUEUE	SPI(TDQUEUE)
TEMPSTORAGE	SPI(TEMPSTOR)
TERMINAL	SPI(TERMINAL)
TRANSACTION	SPI(TRANSACT)
TRDUMPCODE	SPI(TRANDUMP)
TSPOOL	SPI(TSPOOL)
TSQNAME	SPI(TSQNAME)
TSQUEUE	SPI(TSQUEUE)
TYPETERM	SPI(TYPETERM)
UOW	SPI(UOW)
UOWDSNFAIL	SPI(UOWDSNFA)
UOWENQ	SPI(UOWENQ)
UOWLINK	SPI(UOWLINK)
URIMAP	SPI(URIMAP)

Command Keyword	SPI Keyword
VOLUME	SPI(VOLUME)
VTAM	SPI(VTAM)
WEB	SPI(WEB)
WEBSERVICE	SPI(WEBSERVI)
WORKREQUEST	SPI(WORKREQU)
XMLTRANSFORM	SPI(XMLTRANS)

Examples: Securing CICS

In In this example, the user only has permission to execute the CEMT INQUIRE SYSTEM or CEMT INQUIRE commands, since SYSTEM is the default if no function is specified:

```
TSS ADDTO(deptacid) SPI(SYSTEM)

TSS PERMIT(acidname) SPI(SYSTEM)
                     ACCESS(INQUIRE)
```

In In this example, the user only has permission to execute CEMT INQUIRE DUMP commands:

```
TSS ADDTO(deptacid) SPI(DUMPDS)

TSS PERMIT(acidname) SPI(DUMPDS)
                     ACCESS(INQUIRE)
```

In In this example, the user only has permission to execute CEMT INQUIRE AUTOINSTALL commands:

```
TSS ADDTO(deptacid) SPI(AUTOINST)

TSS PERMIT(acidname) SPI(AUTOINST)
                     ACCESS(INQUIRE)
```

Note: Although authorization to SPI resources can be specified for up to 44 characters, ownership of the resource is limited to eight characters.

In In this example, the user only has permission to execute CEMT SET VTAM OPEN commands:

```
TSS ADDTO(deptacid) SPI(VTAM)

TSS PERMIT(acidname) SPI(VTAM) ACCESS(SET)
```

Secondary Resource Checks

The following table indicates that certain CEMT keywords require secondary resource checks. When secondary checks are used:

- SPI resource access ensures that the user is permitted to display or alter a particular type of CICS resource.
- Individual resource access allows display or alteration of the individual resources displayed at the user's terminal.

Like CEMT INQUIRE, the CEMT SET action is also used to provide a display of affected resources (after the SET operands are implemented). For this reason, individual resources described in the table, *CEMT Secondary Resource Checks*, will often need *both* INQUIRE and SET access to invoke alteration through CEMT. You should also note that:

- SET access does not imply INQUIRE access.
- When the CEMT SET action is applied to these resources, both SET and INQUIRE access is required through CA Top Secret.
- Whether the CEMT SET or INQUIRE action is used to initiate a resource display for the keywords in this table, both SET and INQUIRE access through CA Top Secret is required to alter the individual CICS resource.
- When an individual resource is permitted only INQUIRE access, the resource can be displayed but not altered, whether SPI access to INQUIRE or SET the CICS resource class has been granted.

The following table shows the relationship between a CEMT keywrod and a secondary resource type:

CEMT Keyword	Secondary Resource Type
DB2ENTRY*	DB2ENTRY
DB2TRAN*	DB2TRAN
DSNAME	DATASET
FILE	FCT
JOURNAL	JCT
PROGRAM	PPT
QUEUE	DCT
TRANSACTIONS	OTRAN or LCF
VOLUMES	VOLUMES
Note: * CTS 1.2 and above only	

Notes

- DSNAME access checking by CA Top Secret requires the FACILITY control option DSNCHECK=YES. This is set via the command:

  ```
  TSS MODIFY FACILITY(facility=DSNCHECK=YES)
  ```

  When this control option is in effect, CA Top Secret checks DATASET, but not FCT resources for FILE or DATASET keywords in INQUIRE or SET actions through CEMT.

- FCT access checking by CA Top Secret requires the FACILITY control option DSNCHECK=NO (the default). This is set via the command:

  ```
  TSS MODIFY FACILITY(facility=DSNCHECK=NO)
  ```

  When this control option is in effect, CA Top Secret checks the FCT but not DATASET resources when FILE or DATASET keywords with INQUIRE or SET actions through CEMT.

Examples: securing CEMT secondary resources

Using the following commands, the user only has permission to execute CEMT INQUIRE TRAN(CS*) commands.

```
TSS ADDTO(deptacid) SPI(TRANSACT)

TSS ADDTO(deptacid) OTRAN(CS)

TSS PERMIT(acidname) SPI(TRANSACT)
                     ACCESS(INQUIRE)

TSS PERMIT(acidname) OTRAN(CS)
                     ACCESS(INQUIRE)
```

Note: The OTRAN permission in the above example does not allow the ACID to use the transactions.

Although authorization to SPI resources can be specified for up to 44 characters, ownership of the resource is limited to eight characters.

Securing PERFORM Commands

The PERFORM action of the CEMT command has related keywords. This section describes how CA Top Secret secures each keyword for the CEMT PERFORM action.

The following table lists the CEMT command keywords and their SPI equivalents for the CEMT PERFORM action:

Command Keyword	SPI Keyword
CLASSCACHE	SPI(CLASSCAC)
DELETESHIPPED	SPI(DELETESH)
DUMP	SPI(DUMP)
RECONNECT	SPI(RECONNEC)
RESET	SPI(RESET)
SECURITY	SPI(SECURITY)
SHUTDOWN	SPI(SHUTDOWN)
SNAP	SPI(SNAP)
STATISTICS	SPI(STATISTI)

Example: securing CEMT PERFORM commands

In In this example, the user only has permission to execute CEMT PERFORM SHUTDOWN commands:

```
TSS ADDTO(deptacid) SPI(SHUTDOWN)

TSS PERMIT(acidname) SPI(SHUTDOWN)
                     ACCESS(PERFORM)
```

Securing ADD and REMOVE Commands

In CICS 4.1, you can secure CEMT SET VOLUME( ) ADD and REMOVE commands for VOLUMEs only. The following table lists valid access levels:

Command Keyword	SPI Keyword
VOLUME	SPI(VOLUME)

Examples for securing CEMT ADD and REMOVE commands appear in the following. Using these commands the user has permission to execute CEMT ADD and REMOVE commands for VOLUMEs only.

```
TSS ADDTO(deptacid) SPI(VOLUME)

TSS PERMIT(deptacid) SPI(VOLUME)
                     ACCESS(SET)

TSS REMOVE(acidname) SPI(VOLUME)

TSS REVOKE(acidname) SPI(VOLUME)
                     ACCESS(SET)
```

Securing EXEC CICS Commands

You can secure EXEC CICS commands via the CA Top Secret SPI resource. The syntax for the IBM EXEC CICS command is:

```
EXEC CICS function option(argument)
```

function

Corresponds to the CA Top Secret access level.

option

The equivalent to the CA Top Secret SPI resource.

argument

The data element being examined or modified.

For example:

```
EXEC CICS SET FILE(PAYROLL) OPEN
```

To secure EXEC CICS commands:

- TSS ADD the SPI resource to a department or division ACID.
- TSS PERMIT the SPI resource to the user ACID and include the appropriate access level.

For example:

```
TSS ADDTO(divacid) SPI(FILE)

TSS PERMIT(acid) SPI(FILE)
                 ACCESS(SET)
```

The same SPI keyword is used for both CEMT and EXEC CICS restrictions. Once ownership is established, protection is available for both CEMT and EXEC CICS commands.

INQUIRE and SET Commands

CA Top Secret provides the SPI resource for securing EXEC CICS INQUIRE and SET commands.

The following table lists the EXEC CICS command options and their SPI equivalents for the EXEC CICS INQUIRE and SET commands:

Command Option	SPI Keyword
ASSOCIATE **Note:** You can only use INQUIRE.	SPI(ASSOCIAT)
ATOMSERVICE	SPI(ATOMSERV)
AUTINSTMODEL	SPI(AUTINSTM)
AUTOINSTALL	SPI(AUTOINST)
BUNDLE **Note:** This option is used for CTS 4.1.0 and above.	SPI(BUNDLE)
BUNDLEPart **Note:** This option is used for CTS 4.1.0 and above. You can only use INQUIRE.	SPI(BUNDLEPA)
CAPTURESPEC **Note:** This option is used for CTS 4.1.0 and above. You can only use INQUIRE.	SPI(CAPTURES)
CONNECTION	SPI(CONNECTI)
DB2ENTRY	SPI(DB2ENTRY)
DB2TRAN	SPI(DB2TRAN)
DELETESHIPPED	SPI(DELETESH)
DELTSHIPPED	SPI(DELTSHIP)
DOCTEMPLATE	SPI(DOCTEMPL)
DSNAME	SPI(DSNAME)
DUMPDS	SPI(DUMPDS)
EPADAPTER	SPI(EPADAPTE)
EVENTBINDING **Note:** This option is used for CTS 4.1.0 and above.	SPI(EVENTBIN)

Command Option	SPI Keyword
EVENTPROCESS **Note:** This option is used for CTS 4.1.0 and above.	SPI(EVENTPRO)
EXITPROGRAM	SPI(EXITPROG)
FILE	SPI(FILE)
HOST	SPI(HOST)
IPCONN	SPI(IPCONN)
IPFACILITY **Note:** This option is used for CTS 3.2 and above. You can only use INQUIRE.	SPI(IPFACILI)
IRC	SPI(IRC)
JMODEL	SPI(JMODEL)
JOURNALNAME/JOURNALNUM **Note:** This option is used for CTS 1.2 and above; JOURNALNUM is used for CICS 4.1 and CTS 1.1.	SPI(JOURNAL)
JVM **Note:** This option is used for CICS CTS 2.3 and above. You can only use INQUIRE.	SPI(JVM)
JVMPOOL	SPI(JVMPOOL)
JVMPROFILE **Note:** This option is for CICS CTS 2.3 and above. You can only use INQUIRE.	SPI(JVMPROFI)
JVMSERVER **Note:** This option is used for CTS 4.1.0 and above.	SPI(JVMSERVE)
LIBRARY	SPI(LIBRARY)
MODENAME	SPI(MODENAME)
MONITOR	SPI(MONITOR)
MQCONN **Note:** This option is used for CTS 4.1.0 and above.	SPI(MQCONN)
MQINI **Note:** This option is used for CTS 4.1.0 and above. You can only use INQUIRE.	SPI(MQINI)

Command Option	SPI Keyword
MVSTCB **Note:** This option is for CICS CTS 2.3 and above. You can only use INQUIRE.	SPI(MVSTCB)
NETNAME	SPI(TERMINAL)
PARTNER	SPI(PARTNER)
PIPELINE	SPI(PIPELINE)
PROFILE	SPI(PROFILE)
PROGRAM	SPI(PROGRAM)
REQID	SPI(REQID)
STATISTICS	SPI(STATISTI)
STORAGE	SPI(STORAGE)
STREAMNAME	SPI(STREAMNA)
SYSDUMPCODE	SPI(SYSDUMPC)
SYSTEM	SPI(SYSTEM)
TASK	SPI(TASK)
TCLASS	SPI(TCLASS)
TDQUEUE	SPI(TDQUEUE)
TEMPSTORAGE	SPI(TEMPSTOR)
TERMINAL	SPI(TERMINAL)
TSMODEL	SPI(TSMODEL)
TSPOOL	SPI(DB2CONN)
TRACEDEST	SPI(TRACEDES)
TRACEFLAG	SPI(TRACEFLA)
TRACETYPE	SPI(TRACETYP)
TRANCLASS	SPI(TCLASS)
TRANDUMPCODE	SPI(TRANDUMP)
TRANSACTION	SPI(TRANSACT)
TSQUEUE	SPI(TSQUEUE)
UOW	SPI(UOW)
UOWDSNFAIL	SPI(UOWDSNFA)

Command Option	SPI Keyword
UOWENQ	SPI(UOWENQ)
UOWLINK	SPI(UOWLINK)
URIMAP	SPI(URIMAP)
VOLUME	SPI(VOLUME)
VTAM	SPI(VTAM)
WEB	SPI(WEB)
WEBSERVICE	SPI(WEBSERVI)
XMLTRANSFORM **Note:** This option is used for CTS 4.1.0 and above.	SPI(XMLTRANS)

Secondary Resource Checks

Some EXEC CICS commands result in two CA Top Secret security checks:

- To see if the user is authorized to execute the EXEC CICS command.
- To see if the user is authorized to execute the EXEC CICS command for the specified resource.

The following table contains EXEC CICS keywords, the resource types called in the secondary CA Top Secret security check, and the associated access levels:

EXEC CICS Keyword	Secondary Resource Type	Access Level
DATASET	FCT	INQUIRE, SET
DB2ENTRY	DB2ENTRY	INQUIRE,SET
DB2TRAN	DB2TRAN	INQUIRE,SET
FILE	FCT	INQUIRE, SET
PROGRAM	PPT	INQUIRE, SET
TRANSACTIONS	OTRAN	INQUIRE, SET

Examples: securing EXEC CICS INQUIRE and SET commands

In In this example, the user only has permission to execute EXEC CICS INQUIRE PROGRAM(TSSCAI) commands:

```
TSS ADDTO(deptacid) SPI(PROGRAM)

TSS PERMIT(acidname) SPI(PROGRAM)
                     ACCESS(INQUIRE)

TSS ADDTO(deptacid) PPT(TSSCAI)

TSS PERMIT(acidname) PPT(TSSCAI)
                     ACCESS(INQUIRE)
```

Note: If the program is owned, then ACCESS(EXEC) is required on the PERMIT statement.

Note: For acidname to actually call TSSCAI from a transaction, ACCESS(EXEC) would need to be added to the above PERMIT.

In In this example, the user only has permission to execute EXEC CICS SET TRANSACTION(TSS) commands:

```
TSS ADDTO(deptacid) SPI(TRANSACT)

TSS PERMIT(acidname) SPI(TRANSACT)
                     ACCESS(SET)

TSS ADDTO(deptacid) OTRAN(TSS)

TSS PERMIT(acidname) OTRAN(TSS)
                     ACCESS(SET)
```

Note: Although authorization to SPI resources can be specified for up to 44 characters, ownership of the resource is limited to eight characters.

Securing ENABLE, DISABLE, EXTRACT, and COLLECT STATISTICS Commands

You can secure the ENABLE, DISABLE, EXTRACT, and COLLECT STATISTICS EXEC CICS commands via the CA Top Secret SPI resource. The syntax for the IBM EXEC CICS commands is:

```
EXEC CICS function option(argument)
```

ENABLE, DISABLE, EXTRACT, and COLLECT STATISTICS are command functions.

CA Top Secret protects EXEC CICS commands by providing equivalent SPI access levels for EXEC CICS function options. CA Top Secret secures EXEC CICS functions via two commands:

- TSS ADD the SPI resource to a department or division ACID.
- TSS PERMIT the user ACID and include the appropriate SPI resource access level.

The following table lists valid SPI access levels for EXEC CICS commands:

Command Function	SPI Access Level
ENABLE	SET
DISABLE	SET
EXTRACT	INQUIRE
COLLECT STATISTICS	COLLECT

EXEC CICS ENABLE, DISABLE, EXTRACT, and COLLECT STATISTICS commands have related functions. The next section describes how CA Top Secret secures each function and their associated commands.

Securing Functions

The following table lists the EXEC CICS command functions and their SPI equivalents for the EXEC CICS ENABLE, DISABLE, EXTRACT, and COLLECT STATISTICS commands:

Command Function	SPI Keyword
ENABLE	SPI(EXITPROG)
DISABLE	SPI(EXITPROG)
EXTRACT	SPI(EXITPROG)
COLLECT STATISTICS	SPI(EXITPROG)

Examples: securing EXEC CICS ENABLE, DISABLE, EXTRACT, COLLECT STA EXEC CICS ENABLE

In In this example, the user only has permission to execute the EXEC CICS ENABLE commands:

```
TSS ADDTO(deptacid) SPI(EXITPROG)

TSS PERMIT(acidname) SPI(EXITPROG)
                     ACCESS(SET)
```

In In this example, the user only has permission to execute the EXEC CICS DISABLE commands:

```
TSS ADDTO(deptacid) SPI(EXITPROG)

TSS PERMIT(acidname) SPI(EXITPROG)
                     ACCESS(SET)
```

In In this example, the user only has permission to execute the EXEC CICS EXTRACT commands:

```
TSS ADDTO(deptacid) SPI(EXITPROG)

TSS PERMIT(acidname) SPI(EXITPROG)
                     ACCESS(INQUIRE)
```

In In this example, the user only has permission to execute the EXEC CICS COLLECT STATISTICS commands:

```
TSS ADDTO(deptacid) SPI(EXITPROG)

TSS PERMIT(acidname) SPI(EXITPROG)
                     ACCESS(COLLECT)
```

Secure CICS SPOOLOPEN Commands

You can secure EXEC CICS SPOOLOPEN commands via the CA Top Secret SPI resource. The syntax for the EXEC CICS command is:

```
EXEC CICS function option(argument)
```

JESSPOOL is the **function** of the EXEC CICS command.

CA Top Secret provides equivalent SPI access levels to secure EXEC CICS SPOOLOPEN commands.

This table lists the EXEC CICS command function and the SPI equivalent for the EXEC CICS SPOOLOPEN commands.

Command Function	SPI Keyword
SPOOLOPEN	SPI(JESSPOOL)

This table lists valid SPI access levels for EXEC CICS SPOOLOPEN commands.

Command Options	SPI Access Level
INPUT	SET
OUTPUT	SET

Examples: securing EXEC CICS SPOOLOPEN commands

In this example, the user only has permission to execute the EXEC CICS SPOOLOPEN INPUT commands:

```
TSS ADDTO(deptacid) SPI(JESSPOOL)

TSS PERMIT(acidname) SPI(JESSPOOL)
                     ACCESS(SET)
```

In this example, the user only has permission to execute the EXEC CICS SPOOLOPEN INPUT commands:

```
TSS ADDTO(deptacid) SPI(JESSPOOL)

TSS PERMIT(acidname) SPI(JESSPOOL)
                     ACCESS(SET)
```

SPOOLOPEN USERID Commands

To have CA Top Secret spool protection and protect the userid in a particular CICS facility, define them as ABSTRACT resources as shown in the following examples.

Using the commands shown next, the user only has permission to execute the EXEC CICS SPOOLOPEN INPUT USERID commands.

```
TSS ADDTO(deptacid) ABSTRACT(ext writer name)

TSS PERMIT(acidname) ABSTRACT(ext writer name)
```

EXEC CICS SPOOLOPEN OUTPUT USERID(userid)

In this example, the user only has permission to execute the EXEC CICS SPOOLOPEN OUTPUT USERID commands.

```
TSS ADDTO(deptacid) ABSTRACT(userid)

TSS PERMIT(acidname) ABSTRACT(userid)
```

QUERY SECURITY Command

The EXEC CICS QUERY SECURITY command and its functions are fully supported by CICS. See the IBM C*ICS Application Programmers* Reference and *CICS/ESA CICS-RACF Security Guide* for more information.

Note: The QUERY SECURITY command as provided by IBM allows a limited number of access levels to be checked which do not always correspond to all access levels supported by CA Top Secret. However, the CA Top Secret application interface supports all access levels.

Securing the CSD Command

CICS TS 4.1 supports a new CSD command that you may use to directly update the CICS/TS CSD file without the use of the CEDA transaction. You can secure the CSD command using the CA Top Secret SPI resource.

The general syntax for the IBM EXEC CICS command is:

```
EXEC CICS CSD function option(argument)
```

CA Top Secret protects the EXEC CICS CSD command by providing equivalent SPI access levels for the EXEC CICS CSD function options. CA Top Secret secures the EXEC CICS CSD command using two commands:

- TSS ADD the CSD SPI resource to a department or division ACID.
- TSS PERMIT the user ACID and include the appropriate CSD SPI access level.

The equivalent CA Top Secret commands for the IBM EXEC CICS CSD commands shown are as follows:

```
TSS ADDTO(deptacid) SPI(CSD)
TSS PERMIT(acid) SPI(CSD) ACCESS(accesslevel)
```

Granting a user an SPI(CSD) access level lets them use all of the functions requiring that level of access.

CSD Command Access Levels

The CSD command functions and their Access Levels are as follows:

CSD Function	Access Level
INSTALL	CREATE
DISCONN	INQUIRE
ENDBRGROUP	INQUIRE
ENDBRLIST	INQUIRE
ENDBRRSTCE	INQUIRE
GETNEXTGROUP	INQUIRE
GETNEXTLIST	INQUIRE
GETNEXTRSRCE	INQUIRE
INQUIREGROUP	INQUIRE

CSD Function	Access Level
INQUIRELIST	INQUIRE
INQUIRERSRC	INQUIRE
STARTBRGROUP	INQUIRE
STARTBRLIST	INQUIRE
STARTBRRSRCE	INQUIRE
ADD	SET
ALTER	SET
APPEND	SET
COPY	SET
DEFINE	SET
DELETE	SET
LOCK	SET
REMOVE	SET
RENAME	SET
UNLOCK	SET
USERDEFINE	SET

Securing DL/I PSBs and DBDs

IMS and DL/I regions can be connected via ISC to CICS. For more information, see your IBM documentation.

CA Top Secret invokes the External Security Manager and makes checks against the PSB at scheduling time. If the installation is able to schedule the PSB, it returns the DBD names. CA Top Secret checks to determine who has access to the specific DBD, and you must have the appropriate authorization. Access control from CICS where the DBD resides.

If you are not authorized to access the DBD, a DHA4 abend will occur.

Using Resource Caching

CA Top Secret maintains a cache of permitted resources that a user has already been allowed to access. Checking for subsequent access during the life of the cache is allowed if the resource is present in the cache with a matching access level. Since the resource has already been vetted before entering the cache, subsequent accesses need not access the security file to revalidate. By making use of the cache, CA Top Secret reduces security file I/O.

The following facility control option is used in CICS-type facilities to control the resource cache:

```
CICSCACHE({TASKLIFE|SESSLIFE},{NOAUDIT|AUDIT},{512|1024|2048|4096})
```

This control option is normally set in the parameter file on your CICS facilities. However, the first two parameters may be changed dynamically while CICS is running, for testing and diagnostic purposes.

Resource Cache Operation

Each terminal is allocated its own resource cache buffer. The size of the buffer is determined at start-up from the CICSCACHE facility option. Complex transactions that access many secured resources may require larger buffers. If the buffer becomes full during the life of the cache, it will be cleared to accommodate new entries. Earlier cached entries will be lost. In order to maintain the advantages of caching, the cache size selected must be appropriate to the applications in the region with that facility. When TASKLIFE caching is used, the cache is cleared at the start of each transaction. When SESSLIFE caching is used, the cache is cleared when the user signs off. SESSLIFE is maintained as far as possible during operation until new entries require clearance.

Cached resources are only checked for

- Resource class
- Resource name
- Access level

Additional restrictions are not checked for cached resources. For example:

- PRIVPGM
- LIB
- DAYS
- TIME
- CALENDAR
- TIMEREC

However, RLP restrictions will be checked on every check, if they were present when they entered the cache.

- Advantages of TASKLIFE caching
  - Cache size is smaller than for session life cache
  - If changes occur in a user's permission to a resource, they will be rechecked at each transaction
  - If restrictions are present on the permission, they will be rechecked at each transaction
- Advantages of SESSLIFE caching
  - Repeated access to a set of resources will remain in cache during the life of the end-user session, preventing the need to recheck
  - Significant reduction in Security File I/O, during run-time applications, especially with repeated complex transactions

- Disadvantages of TASKLIFE caching
  - Increased access to the security file may lead to performance degradation
- Disadvantages to SESSLIFE caching
  - Some restrictions on permissions will not be honored on subsequent accesses, after a resource has been cached.
  - Memory required for session cache is larger than for tasklife
  - Audit requirements may be compromised
- When CICSCACHE is set to AUDIT, then resource classes with the AUDIT attribute, individual resources in the AUDIT record, and users with the AUDIT attribute will be treated differently in cache than they would otherwise.
  - Audited resources will automatically be rechecked with each access regardless of caching.
  - Audited accesses will be logged in the ATF or SMF, according to your installation control option settings
  - When RDT or AUDIT record no longer require auditing, processing of cache returns to normal immediately
  - When a user is signed on, the AUDIT requirement is maintained until the user signs on again or is refreshed by the administrator

When CICSCACHE defaults to NOAUDIT, auditing occurs only on the first access to the resource during the life of the cache. Clearly, if session life caching is in place with NOAUDIT, it may be possible to miss auditable events during the life of the check.

Resource Cache Processing

When a resource validation is required, CA Top Secret scans the terminal's resource cache for the requested resource before asking the host CA Top Secret system to perform the validation. If the requested resource name does not match one of the resource cache entries, or if AUDIT is set and the cache entry is marked for audit, the host is asked to perform a normal resource validation. The result of the host resource validation determines whether the resource is added to the resource cache.

- If the user is allowed access to the resource, CA Top Secret adds the current resource as a new entry to the cache buffer.
- If the resource cache is too full to accept another allowed resource, the least frequently accessed entry in the cache is dropped to make room for the new entry.
- If the user is flagged for AUDIT, if the resource name is in the AUDIT record, or if the resource class is flagged for AUDIT, and CICSCACHE has been set for AUDIT, then the AUDIT flag is set for that resource in the cache, to remind the security product that a full resource check is required on the next access.
- If the result of the host resource validation is to deny access, the resource cache is not updated and normal violation processing takes place.
- If a requested resource is found in the cache, CA Top Secret assumes that access is allowed, but security processing (password verification) is still performed. However, certain permission restrictions will not be checked, as described above.
- If host resource validation is done in WARN mode and a user is not allowed to access a resource, the resource is not added to the cache if such validation would otherwise fail when running in FAIL mode.

How to Set CICSCACHE

CICSCACHE should normally be set in the TSS PARMFILE. The defaults for this facility option are:

```
CICSCACHE(TASKLIFE,NOAUDIT,512)
```

This is adequate for most purposes and provides the maximum resource security with minimum-security file access for most small applications with a small number of secured resources. Should you wish to adjust the cache size, this must be done before starting the CICS region. Although the task size in the facility display will show your altered value, the new cache value will not be implemented until the region is restarted. The options TASKLIFE/SESSLIFE and NOAUDIT/AUDIT may be altered dynamically and take effect immediately.

Note: OPTIONS(28) no longer has any effect on session life and will be overridden by the associated facility CICSCACHE values.

If you need to troubleshoot a problem with support, and you use SESSLIFE caching, be sure to let support know that this is the case. For the purposes of troubleshooting, you may wish to alter CICSCACHE to AUDIT, and to flag your test user with the AUDIT flag, in order to obtain detailed information about the activities of a problem transaction.

Tuning the Session Cache

To help you tune the resource cache, CA Top Secret provides functions to display cache utilization on a global system level and on an individual terminal level. You should compare the number of times that overflows occurred to the number of validations requested. Some cache overflow is reasonable, since a small set of users might access many different resources in a CICS system. However, if no overflows are noted, then the cache size is probably too large for the kind of resource validation activity taking place in the CICS system.

Note: If the cache size is too small, excessive host resource validation is performed. Making the cache too large can result in excessive operating system paging.

Displaying the Global Cache Status

To display the global cache utilization status, execute the following transaction from any CICS terminal: TSEU=MAXT. Executing this transaction displays the following information:

```
Maximum users = 3000
  Total number session-related tokens = 750
  Allocated session-related tokens = 5
  Max concurrent sign-on/off requests = 10
  Current no. of users signed on = 3
  Active user-related storage = 1360 bytes
  Total user-related storage = 83K  bytes
  Active user-related cache storage = 6144 bytes
  Total user-related cache storage = 1500K bytes

  CICS Maximum task value = 60
  Current no. of tasks = 10
  Active task-related storage = 40K  bytes
  Total task-related storage = 240K bytes
  Total number task-related tokens = 60

  Times resource found in cache = 30
  Times resource not found in cache = 16
  Times cache overflowed = 0
  Number entries added to cache = 16
  Number entries excluded from cache = 0
  Session cache box size = 2048 bytes
Command complete
```

The cache-related information displayed on this screen is displayed below.

Times resource found in cache

Number of times cache resource scan was successful.

Times resource not found in cache

Number of times cache resource scan was unsuccessful.

Times cache overflowed

Number of times a cache entry had to be deleted to make room for a new entry.

Number entries added to cache

Total number of new entries added to the resource cache.

Number entries excluded from cache

Number of times a resource was allowed to be accessed, but still was not added to the cache.

Resource cache box size

Current size of each resource cache buffer. This is the value of the cache-size in CICSCACHE when the CICS region was initialized.

Displaying Terminal Cache Status

To display the cache information for your current terminal, use the following transaction:

```
TSEU=TERM=*
```

This display shows very little unless SESSLIFE is in force, since the transaction TSEU will be the only transaction active in your session, and TSEU is in the bypass list.

If you are attempting to view a session at another terminal, use the following transaction:

```
TSEU=TERM=termid
```

termid

A 4-character CICS terminal id.

Unless your TSEU transaction is timed just right, it is unlikely that you will have anything worth review when TASKLIFE is in force.

The following represents as session with multiple transactions while SESSLIFE is in force:

```
ANALYSIS OF TERMINAL G005
Address of security anchor is (7F4FD318)
The USER on this terminal is CICSPEON
The user is defined and is running in FAIL mode
The security record is located in HIGH PRIVATE

The CICS operator identifier is USA
The CICS operator priority is 000
The attended bit is ON

------------------------ User SESSLIFE CACHE -----------------------
Lookups  38        Hits  30        Inserts  8         Overflows  0
% Cache used  11.474
--------------------------  CACHE Detail  --------------------------
Resource      Resource    Number         Resource
  Class       Access      Hits           Name
LCF....%       READ         5           AAAA
PPT...Q%       READ         5           AAAAAAAADFHCICST
LCF....%       READ        10           XXXX
PPT...Q%       READ        10           XXXXXXXXDFHCICST
LCF....%       READ         0           CEMT
PPT...Q%       READ         0           DFHEMTP DFHCICST
PPT...Q%       READ         0           DFHEITMT
PPT...Q%       READ         0           DFHEMTD
Command complete
```

Global Terminal Cache Utilization

Inserts

Number of times a new entry was added to the cache.

Overflows

Number of times entries were deleted.

Detailed Resource Level Information

Resource class

Class name for the current resource.

Resource access

Requested access level.

Number hits

Number of times the current resource entry was found in the cache.

Resource name

Name of the current resource.

Chapter 5: Programmable Interfaces

This section contains the following topics:

Issuing TSS Commands Under CICS

The TSS transaction is defined to execute the program TSSCICS when you install the CSD definitions for CA Top Secret. Normally this transaction takes its input from the terminal input-output area (TIOA), by entering a command from a clear unformatted screen, for example:

```
TSS WHOAMI
```

However, TSSCICS can be invoked from CICS commands programmatically, in which case it may take its command input from the program COMMAREA or from temporary storage queues. Invocation of TSSCICS in these programmatic ways is explained in the following sections.

Note: In CICS, if TSSCICS is invoked from a non-TSS transaction, and PF3 is depressed, execution of TSSCICS will terminate and the command output may be prematurely terminated. This can cause unpredictable results in the calling transaction. It is the programmer's responsibility to assure that PF3 is not used during the execution of TSSCICS as a called program.

Sample Program Calling TSSCICS via COMMAREA

This sample program calls TSSCICS via COMMAREA:

```
TITLE   'ISSUE TSS COMMAND VIA COMMAREA'
COMMAND         CSECT
                EXEC CICS LINK PROGRAM('TSSCICS') COMMAREA(COMM) LENGTH(256)
                EXEC CICS RETURN
*
* DATA FOLLOWS
*
COMM            DC      CL256'TSS LIST(userid) DATA(ALL)'
                END
```

Sample Program Calling TSSCICS via TEMPORARY STORAGE and TERMID

When executing a TSS command as part of a CICS transaction (executed from a valid terminal), you can pass the command to the program TSSCICS in a temporary storage queue and receive output data back using the same queue.

Consider following considerations command behaviors:

- When you execute TSSCICS from a terminal, the temporary storage queue should be the concatenation of “TSSA” with the 4-character terminal ID. In the following sample program, the terminal ID is obtained using the CICS EXEC ASSIGN FACILITY() command.
- As a precaution to prevent previously stored commands or output from interfering with your communication with TSSCICS, delete the queue prior to writing.
- Delete the queue after your output has been read.
- The length of the output buffer (in the sample program TSSLNGTH) must be lowered to 80 if the terminal is a sequential terminal.
- When larger buffer sizes are used, it is the user's responsibility to deblock the output from multiple-line buffers.
- Use of TSSCICS with large volumes of output (for example, LIST commands for large ACIDs) should be avoided. Clients are cautioned to limit commands to those with minimal outputs. For complex inquiries, TSSCAI can be of greater utility than TSSCICS.

Example: calling TSSCICS via TEMPORARY STORAGE and TERMID

This example calls TSSCICS via TEMPORARY STORAGE and TERMID. This program is designed for command outputs which span, at most, a single 3270-2 screen (1920 bytes). We recommend that you use TERMID as an ID for transactions expected to operate from a terminal:

```
           TITLE 'ISSUE TSS COMMAND USING TEMPORARY STORAGE & TERMID'
OUT        EQU  8
PGMCTSS    DFHEIENT CODEREG=(3),DATAREG=(5)

MAINLINE   DS   0H
           EXEC CICS ASSIGN                                          X
                FACILITY(NET)
           MVC  WQUEUEID,=CL4'TSSA'
           MVC  WQUEUEID+4(4),NET
           EXEC CICS DELETEQ TS QUEUE(WQUEUEID) RESP(RESP)
           EXEC CICS WRITEQ TS                                  X
                QUEUE(WQUEUEID)                                 X
                FROM(MSG)                                       X
                LENGTH(256)                                             X
                RESP(RESP)
           EXEC CICS LINK PROGRAM('TSSCICS')
           EXEC CICS READQ TS QUEUE(WQUEUEID)                   X
                SET(OUT) LENGTH(TSSLNGTH) RESP(RESP)
            EXIT EXEC CICS DELETEQ TS QUEUE(WQUEUEID) RESP(RESP)
           EXEC CICS RETURN
MSG        DC   CL256'TSS REPL(userid) PASS(pass,30,EXP)
RESP       DS   F
TSSLNGTH   DC   H'+1920'
WQUEUEID   DS   CL8
NET        DS   CL4
           END
```

Sample Program Calling TSSCICS via TEMPORARY STORAGE and TASK NUMBER

When executing a TSS command as part of a TSSCICS transaction (without a terminal), you can pass the command to the program TSSCICS in a temporary storage queue and receive output data back using the same queue.

Consider the following command behaviors:

- When you execute TSSCICS without a terminal, the temporary storage queue should be the concatenation of “TSSA” with the four-character task number. In the sample program below, the task number is obtained using the CICS EIB field EIBTASKN.
- As a precaution to prevent previously stored commands or output from interfering with your communication with TSSCICS, delete the queue prior to writing.
- Delete the queue after your output has been read.
- The length of the output buffer (in the sample program TSSLNGTH) must be lowered to 80 if the terminal is a sequential terminal.
- When larger buffer sizes are used, it is the user's responsibility to deblock the output from multiple-line buffers.
- Use of TSSCICS with large volumes of output (for example, LIST commands for large ACIDs) should be avoided. Clients are cautioned to limit commands to those with minimal outputs. For complex inquiries, TSSCAI can be of greater utility than TSSCICS.

Example: calling TSSCICS via TEMPORARY STORAGE and TASK NUMBER

This example program calls TSSCICS via TEMPORARY STORAGE and TASK NUMBER. This program is designed for command outputs which span, at most, a single 3270-2 screen (1920 bytes). We recommend that you use Task Number for transactions that operate from PLTI or from EXEC START commands, and that are not associated with a terminal:

```
            TITLE 'ISSUE TSS COMMAND USING TEMPORARY STORAGE & TASK #'
OUT         EQU   8
PGMCTSS     DFHEIENT  CODEREG=(3),DATAREG=(5)

MAINLINE    DS    0H
            MVC   WQUEUEID,=CL4'TSSA'
            MVC   WQUEUEID+4(4), EIBTASKN
            EXEC  CICS DELETEQ TS QUEUE(WQUEUEID) RESP(RESP)
            EXEC  CICS WRITEQ TS                                  X
                  QUEUE(WQUEUEID)                                       X
                  FROM(MSG)                                             X
                  LENGTH(256)                                     X
                  RESP(RESP)
            EXEC  CICS LINK PROGRAM('TSSCICS')
            EXEC  CICS READQ TS QUEUE(WQUEUEID)                   X
                  SET(OUT) LENGTH(TSSLNGTH) RESP(RESP)
            EXIT  EXEC  CICS DELETEQ TS QUEUE(WQUEUEID) RESP(RESP)
            EXEC  CICS RETURN
MSG         DC    CL256'TSS LIST(userid) DATA(ALL)'
RESP        DS    F
TSSLNGTH    DC    H'+1920'
WQUEUEID    DS    CL8
DFHEISTG    DSECT
            END
```

Application Interface

The CA Top Secret Application Interface is a CICS application program that performs security checking and other CA Top Secret services. This program allows CA Top Secret to provide security for installation-defined resources that are not protected by CA Top Secret.

Invoking the Application Interface

The Application Interface program for CICS, TSSCAI, resides in and is distributed in the CA Top Secret load library. The TSSCAI program must reside in a DFHRPL program library, and must be defined to the DFHPPT definition macro. The Application Interface can be invoked by Command- or Macro-Level programs, written in any of the following programming languages:

- COBOL
- PL/I
- Assembler

Examples of the Assembler coding required to invoke the Application Interface are located in Coding Samples. Samples of PL/I, Assembler, and COBOL coding are located on the CA Top Secret distribution tape under CAI.CAKOSRC0.

Note: The TSSCAI program does NOT check the CICS facility bypass lists.

Writing Requirements

Follow these guidelines when writing the CICS Application Interface:

- Use the CA Top Secret Application Interface via an EXEC CICS LINK (or a DFHPC TYPE=LINK) statement.
- The name of the CA Top Secret Application Interface program that you are linking to is TSSCAI.
- TSSCAI and TSSCAIN must have CSD entries like the one shown below (the example shows that the program is written in the Assembler language).
- Note the effect of OPTIONS(22) and OPTIONS(64) control options on the operation of TSSCAI.

  You must add a CSD entry for modules TSSCAI and TSSCAIN with EXECKEY(CICS) when issuing TSSCAI calls.

```
DEFINE PROGRAM(TSSCAI) GROUP(TOPSGRP)
       DESCRIPTION(CA Top Secret Application Interface Stub)
       LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(CICS)
       RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO)
       STATUS(ENABLED) CEDF(NO) DATALOTION(BELOW)
DEFINE PROGRAM(TSSCAIN) GROUP(TOPSGRP)
       DESCRIPTION(CA Top Secret Application Interface Stub)
       LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(CICS)
       RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO)
       STATUS(ENABLED) CEDF(NO) DATALOTION(BELOW)
```

- You must pass the Application Interface a parameter list. The parameter list is passed by a temporary storage queue or by Command-Level COMMAREA.
- The length of the COMMAREA or the temporary storage queues that contain the Application Interface parameter list must be:
  - For Release V4L1-320 bytes
  - For Release V4L2-370 bytes
  - For Release V4L3-370 or 1138 bytes (The 1138 value refers to the FACLIST, RESLIST, and FLDXTR calls; all other calls use 370 bytes.)
  - For Release V4L4-370 or 1138 bytes (The 1138 value refers to the FACLIST, RESLIST, and FLDXTR calls; all other calls use 370 bytes.)
  - For Release V5L1-370 bytes, 1138 bytes, or larger. (The 1138 value refers to the FACLIST, RESLIST, and FLDXTR calls; all other calls use 370 bytes.).

    V5L1 requires that you fill in TSSLRTN with the length of TSSRTN, that is MVC TSSLRTN,=AL4(L'TSSRTN)
- If your release of CICS supports concurrency, you can specify CONCURRENCY(THREADSAFE) in the RDO entry for TSSCAI and TSSCAIN.

Installation-Defined Resources

Installation-defined resource classes (such as FIELD, UR1, UR2, and ABSTRACT) that are also pre-defined in the Resource Descriptor Table (RDT) require the use of the Application Interface to be protected by CA Top Secret. These resource types allow an individual site to extend security to resources, such as database fields, that CA Top Secret does not usually protect.

An installation can also dynamically define any resource that it wishes to protect. For more details, see the TSS ADD(RDT) command function in the *Command Functions Guide*.

Transaction Checking

An application can perform a transaction or panel check by specifying a class name of LCF and a resource name consisting of the transaction or panel name. No other fields are required for a transaction check. See the chapter, “Implementing Security,” for information on administering transaction security.

Note: OTRAN only provides security checking for owned transactions while LCF checks for both owned and unowned transactions.

Coding Samples

Use the coding examples provided in this section as customization samples.

Test TSSCAI Using Temporary Storage Record

This section contains a customization sample.

```
          TITLE 'TESTCAI 1' --- TSS CICS APPLIATION INTERFACE
*********************************************************************
* NAME     - TESTCAI1 0
* FUNCTION - COMMAND LEVEL ASSEMBLER CODE ..... 0
*            TEST TSSCAI USING TEMPORARY STORAGE RECORD. 0
* CALLS    - THE CICS APPLICATION INTERFACE PROGRAM 0
*********************************************************************
         EJECT
DFHIESTG DSECT
TSSQID   DS    0CL8       TEMPORARY STORAGE QUEUE NAME
TSSQPREF DS    CL4        QUEUE ID PREFIX IS ALWAYS 'TSSA'
TSSQTERM DS    CL4        QUEUE NAME SUFFIX IS TERMINAL NAME
TSSCREC  DS    2CL185     PARAMETER LIST FOR TSSCAI
TSSITEM  DS    H
         EJECT
R2       EQU   2
TESTCAI1 CSECT
* TELL CICS TO IGNORE A QUEUE NOT FOUND CONDITION.
         EXEC CICS IGNORE  CONDITION QIDERR
*
* PURGE THE QUEUE OF ANY OLD REQUESTS.
*
         MVC   TSSQPREF,=CL4'TSSA'
         MVC   TSSQTERM,=EIBTRMID
*
         EXEC  CICS DELETEQ TS QUEUE(TSSQID)
* RESET THE HANDLE.
*
         EXEC  CICS HANDLE CONDITION QIDERR
         EXEC  CICS IGNORE CONDITION LENGERR
*
* BUILD THE TSSCPL PARAMETER LIST.
         MVC   TSSQPREF,=CL4'TSSA'
         MVC   TSSQTERM,EIBTRMID
         LA    R2,TSSCREC          R2 @ OF PARAMETER LIST
         USING TSSCPL,R2           ESTABLISH ADDRESSABILITY
         MVC   TSSHEAD,=CL8'TCPLV3L0'
         MVC   TSSCLASS,=CL8'FIELD '
         MVC   TSSRNAME,=CL8'TSSFIELD'
         MVC   TSSPPGM,=CL8' '
         XC    TSSACC,TSSACC
*
* WRITE THE REQUEST RECORD TO TEMPORARY STRORAGE.
*
         EXEC  CICS WRITEQ TS QUEUE(TSSQID)                    X
               FROM(TSSCREC) LENGTH(TSSLNGTH) MAIN
```

```
*
* INVOKE THE TSS APPLICATION INTERFACE TO PROCESS THE REQUEST
*
         EXEC  CICS LINK PROGRAM('TSSCAI')
*
* READ THE REQUEST RECORD BACK FROM TEMPORARY STORAGE.
         EXEC  CICS READQ TS QUEUE(TSSQID)                       X
               INTO(TSSCREC) LENGTH(TSSLNGTH)
*
* PURGE THE REQUEST QUEUE.
         EXEC  CICS DELETEQ TS QUEUE(TSSQID)
*
* RETURN TO CICS
*
         EXEC  CICS RETURN

* WORKING STORAGE.
*
TSSLNGTH DC    H'+370'
         #TSSCPL               CICS PARAMETER LIST
         END
```

Test TSSCAI Using CICS COMMAREA

This section contains a customization sample.

```
           TITLE 'TESTCAI2 --- CICS APPLICATION INTERFACE'
*****************************************************************
*
* NAME      - TESTCAI2
*
* FUNCTION  - COMMAND LEVEL ASSEMBLER CODE ......
*             TEST TSSCAI USING CICS COMMAREA.
*
* CALLS     - THE CICS APPLICATION INTERFACE PROGRAM
*
*****************************************************************
*
          EJECT
DFHEISTG  DSECT
TSSCREC   DS     CL185               PARAMETER LIST LENGTH
          DS     CL185               PARAMETER LIST LENGTH
          EJECT
R2        EQU    2                   BASE REG FOR PARAMETER LIST
TESTCAI2  CSECT
*
* BUILD THE TSSCPL PARAMETER LIST.
*
          LA     R2,TSSCREC          R2 @ PARAMETER LIST
          USING  TSSCPL,R2           ESTABLISH ADDRESSABILITY
          MVC    TSSHEAD,=CL8'TCPLV3L0'
          MVC    TSSCLASS,=CL8'FIELD '
          MVC    TSSCLASS,=CL8'TSSFIELD'
          MVC    TSSPPGM,=CL8' '
          XC     TSSACC,TSSACC
* INVOKE THE TSS APPLICATION INTERFACE TO PROCESS THE REQUEST.
*
          EXEC CICS LINK PROGRAM('TSSCAI') COMMAREA(TSSCPL) LENGTH(370)
*
* RETURN TO CICS
*
          EXEC CICS RETURN
*
* WORKING STORAGE.
*
*         #TSSCPL
          END
```

CA Top Secret CICS Exits

CA Top Secret provides the user exits:

- TSSPGM01-A message exit that lets you suppress or change the text of messages.
- TSSPGM02-A message exit that lets you suppress or change the text of the locktime prompt.

The TSSPGM01 Exit

The TSSPGM01 exit is enabled by defining the PROGRAM=TSSPGM01 to your CICS environment. CA Top Secret CICS invokes the exit by issuing:

```
EXEC     CICS LINK
         PROGRAM
         COMMAREA
         LENGTH
         RESP
```

TSSPGM01 can be invoked before CA Top Secret issues any CA Top Secret CICS messages (except password prompts). The exit program must be written in Command-Level Assembler. The COMM area layout is:

```
WPARMLIST     DS      0H          PARAMETER LIST FOR EXIT
WMESSAGE      DS      XL800       MESSAGE AREA
WMSGLRC       DS      X           RETURN CODE
$TEXIT        EQU     X'00'       EXIT MODULE
$TWRTTD       EQU     X'01'       WRITE MESSAGE TO TD QUEUE
$TWRITE       EQU     X'02'       WRITE MESSAGE TO TERMINAL
$TABEND       EQU     X'FF'       ABEND TASK
WMSGALN       EQU     *-WPARMLST  PARAMETER LIST LENGTH
```

Note: If running CICS/ESA, this exit must run AMODE(31).

WMESSAGE

Contains the message to be written to the user's terminal. The message is in a BMS Send TEXT format.

WMSGLRC

Contains a return code the user will enter in the TSSPGM01 exit program.

$TEXIT

Indicates that CA Top Secret messages are not written to the user's terminal.

$TWRTTD

Writes the CA Top Secret message to the CSML Transient Data Queue.

$TWRITE

Writes the CA Top Secret message to the user's terminal.

$TABEND

Is the ABEND transaction (ABEND Code TAZ7).

Note: A sample exit program resides in CAI.CAKOSRC0, and the JCL to assemble and link it resides in CAI.CAKOJCL0.

The TSSPGM02 Exit

The TSSPGM02 exit is enabled by defining the PROGRAM=TSSPGM02 to your CICS environment. CA Top Secret CICS invokes the exit by issuing:

```
EXEC     CICS LINK
         PROGRAM
         COMMAREA
         LENGTH
         RESP
```

TSSPGM02 is invoked for password prompts that CA Top Secret CICS does not support. The exit program must be written in Command-Level Assembler. The COMM area layout is:

```
WPARMLST DSECT                 PARAMETER LIST.
WMGAREA  DS    0XL79           MESSAGE AREA
WMSGLN   DS    H               MESSAGE LENGTH.
WPFLAG2  DS    X               FLAGS
WPPHRASE EQU   X'80'           PASSWORD PHRASE ACTIVE
         DS    X               RESERVED.
WMESSAGE DS    CL75            MESSAGE AREA.
WPFLAG   DS    0XL1            EXIT REQUEST FLAG
WPSEND   EQU   X'FF'           PSEUDO-CONVERSIONAL SEND
WPRECV   EQU   X'FE'           PSEUDO-CONVERSIONAL RECEIVE
WPPWAREA DS    0XL8            PASSWORD AREA.
WPSWD    DS    CL8             PASSWORD.
WPSWDPHR DS    CL100           PASSWORD PHRASE
WPLSTLN  EQU   *-WPARMLST      PARAMETER LIST LENGTH
```

Note: If running CICS/ESA, this exit must run AMODE(31).

WMGAREA

Contains the password prompt message.

WPPWAREA

The field that the TSSPGM02 Exit program places the user's password in for reverification.

WPSWDPHR

The field that the TSSPGM02 Exit program places the user's password phrase in for reverification.

Note: If using password phrases, add the password phrase to the WPSWDPHR field, not the WPSWD field.

Passwords

If you are using the TSSPGM02 Password Prompt Exit, be aware that this interface has been modified to support use in pseudo-conversational mode. Installations that use this exit program will need to modify their code. A new flag has been added in the incoming COMMAREA to indicate to the program whether it should issue a SEND or RECEIVE or a conversational prompt. A sample exit program (TSSPGM02) resides in CAI.AAKOSRC0. A sample CICS map and the JCL to assemble and link it reside in CAI.SAMPJCL (TSSMAP2 and TSSJCLX2).

If you use mixed case passwords, and you set PCLOCK=YES in your CICS facility, alter the password input field TSSPW in map TSSMAP2 to CASE=MIXED in the DFHMDF macro.

Sample Program Definitions

Program definitions for the TSSPGM01 and TSSPGM02 message exits should look like the following examples:

```
DEFINE PROGNAM(TSSPGM01) GROUP(TOPGRP)
       DESCRIPTION(-SECURITY/CICS USER EXIT 01)
       LANGUAGE(ASSEMBLER) EXECKEY(CICS)
       RESIDENT(NO) USAGE(NORM) USELPACOPY(NO)
       STATUS(ENABLED) CEDF(NO) DATALOTION (ANY)

DEFINE PROGRAM(TSSPGM02) GROUP(TOPSGRP)
       DESCRIPTION(CA-SECURITY/CICS USER EXIT 02)
       LANGUAGE(ASSEMBLER) EXECKEY(CICS)
       RESIDENT(NO) USAGE(NORM) USELPACOPY(NO)
       STATUS(ENABLED) CEDF(NO) DATALOCATION(ANY)

DEFINE MAPSET(TSSMAP2) GROUP(TOPSGRP)
       DESCRIPTION(CA-SECURITY/CICS EXIT MAP 02)
       RESIDENT(NO) USAGE(NORM) USELPACOPY(NO)
       STATUS(ENABLED)
```

Note: EXECKEY must be CICS, otherwise errors will result.

Chapter 6: CA Top Secret Supplied Transactions

This section contains the following topics:

LOCKTIME Logoff Feature Support (TSLA, TSLM, TSLK)

The TSLA and TSLM transactions and their associated programs are required if you are specifying the LTLOGOFF FACILITY suboption.

TSLA Transaction

TSLA is a CA Top Secret CICS transaction used to support the LOCKTIME logoff feature set via the LTLOGOFF FACILITY suboption. LTLOGOFF controls whether CA Top Secret causes user logoff after the second LOCKTIME interval expires. This transaction is used to perform CESF LOGOFF and can be issued from EXEC CICS START.

TSLM Transaction

TSLM is a CA Top Secret CICS transaction used with the TSLA transaction (described previously) to support LOCKTIME processing.

TSLK Transaction

TSLK is a CA Top Secret CICS transaction used with the TSLA transaction (described previously) to support pseudo-conversational LOCKTIME processing.

The Environmental Utility (TSEU)

TSEU is a CA Top Secret CICS user-executed transaction utility that provides security-related information. Anyone designated to perform administrative and troubleshooting procedures for your installation will use this utility. The security-related features include:

TSEU=INSTALL

Analyzes the installation specifications of a CICS region.

TSEU=WHOSON

Indicates who is signed on to a particular region. The display only shows users that are signed on through explicit signon, automatic terminal signon, or preset terminal security.

TSEU=TRANS=(trans)

Gives information about a specified CICS transaction, where *trans* is the four-character CICS transaction ID specified in the PCT.

TSEU=CESF=termid

Will reset the ATS flag or sign off the user at the designated terminal.

TSEU=CESF=tttt

Will initiate a signoff for the user at the designated terminal.

TSEU=TERM=(term | *)

Gives information about a specified terminal, where term is the four-character CICS terminal ID specified in the TCT or "*" that indicates the current terminal. The information varies depending on whether a user is signed on.

Note: If you are using mixed mode terminals (terminals that accept lowercase and uppercase input), you must set UCTRAN=NO on the profile for TSEU.

TSEU=MAXT=INQ

Inquires about the maximum number and actions available for concurrent signon/signoff requests that are set.

TSEU=NEWC=(program)

Refreshes the running copy of a TSS CICS module, allowing emergency maintenance to be applied to a single CICS region without recycling that specific region.

TSEU=TRACE=

(INQ|ON|OFF) Inquires on or controls the status of the CA Top Secret CICS diagnostic tracing facility.

Executing TSEU

This section contains a detailed description of the information returned by each of the transactions.

TSEU=INSTALL

Details the following information about a region:

- Whether SEC=YES or SEC=NO is coded.
- Where the CA Top Secret modules are located.
- Whether the Application Interface is installed.
- Whether the TSS command is installed.
- The name of the region control userid.
- The name of the MASTFAC.
- The settings of the XPARMs, DSNCHECK, LTLOGOFF, PCTRESSEC, and PCTCMDSEC.

TSEU=WHOSON

Indicates who is signed on to a particular region.

TSEU=TRANS=(trans)

Describes the following information about a specific CICS transaction:

- Whether the transaction is local or remote.
- The priority at which it is defined.
- Whether the transaction was defined in the PCT or was loaded dynamically from the RDO file.
- Whether the transaction was generated with external security.
- Whether the PCTEXTSEC FACILITY suboption has been set to HONOR or OVERRIDE.
- Whether the transaction resides in the Bypass List.

TSEU=CESF=termid

Resets the ATS flag to allow the terminal to ATS again.

To reset the ATS flag, issue the following command (where L809 is the terminal id):

The terminal that issues the action will receive the following message: “CESF has been scheduled for terminal L809.”

Resetting the ATS flag also allows non-ATS users to be signed off manually avoiding operator already signed on messages.

TSEU=TERM=(term | *)

Gives the following if a user is *not* signed on to a terminal:

- Whether there is a Security Record at the terminal.
- The three-character operator ID.
- If the attend bit is on or off.

TSEU=TERM=(term)

Gives the following information if a user *is* signed on to a terminal:

- Gives the name of the ACID.
- Whether the user is defined or undefined, and the security MODE.
- The three-character CICS operator ID.
- If the attend bit is on or off.

TSEU=MAXT=INQ

Has the ability to *inquire* about the maximum number and actions available for concurrent signon/signoff requests that are set.

A sample screen for TSEU=MAXT=INQ appears next:

```
Maximum users = 3000
Total number session-related tokens = 750
Allocated session-related tokens = 4
Max concurrent sign-on/off requests = 10
Current no. of users signed on = 2
Active user-related storage = 1248 bytes
Total user-related storage = 83K  bytes
Active user-related cache storage = 1024 bytes
Total user-related cache storage = 375K bytes

CICS Maximum task value = 60
Current no. of tasks = 8
Active task-related storage = 32K  bytes
Total task-related storage = 240K bytes
Total number task-related tokens = 60

Times resource found in cache = 0
Times resource not found in cache = 2
Times cache overflowed = 0
Number entries added to cache = 2
Number entries excluded from cache = 0
Session cache box size = 512 bytes
Command complete
```

TSEU=NEWC=(program)

Refreshes the running copy of a CA-Top Secret CICS module, which is not callable from normal CICS application programming. This function allows emergency maintenance to be applied to a single CICS region without recycling that specific region.

The following CA Top Secret CICS modules are refreshed through the CICS command TSEU=NEWCOPY: CAKSLMT, CAKSMSGH, CAKSPCH1, CAKSSIGN, CAKSRINT, CAKSPVAL, CAKSRVAL, CAKSGLUE, CAKSPWH, CAKSCMIN, CAKSHASH, CAKSALOC, CAKSTRPX, CAKSXCMD, CAKSATS, CAKSEXIT, CAKSSHUT, CAKSPSPM, CAKSCBXM, CAKS#SEC, TSSCLMT, TSSCRVAL, TSCRTYxx, TSCCTYxx, TSCRACxx, TSSCXFM.

The following CA Top Secret CICS modules are refreshed through the CICS command CEMT SET PROGRAM(name) NEWCOPY: CAKSEXSN, CAKSCHEK, CAKSINST, CAKSMAXT, CAKSNEWC, CAKSTERM, CAKSTRAC, CAKSTRAN, CAKSWHOS, CAKSWRIT, CAKSSCAN, CAKSLOCK, TSSCAI, TSSCAIN, TSSCICS, TSSCICSN, TSSCPTSS. However, any TSS or CAKS program with a CSD definition can be refreshed using CEMT NEWCOPY.

TSEU=TRACE=(INQ|OFF|ON)

Inquires on or controls the status of the CA Top Secret CICS diagnostic tracing facility. The Trace Facility writes diagnostic trace records into the CICS main trace table.

INQ

Displays the current status (ON|OFF) of the CA Top Secret CICS diagnostic tracing facility.

OFF

Turns off the CA Top Secret CICS diagnostic trace.

ON

Turns on the CA Top Secret CICS diagnostic trace. Note that the CICS auxiliary trace must be controlled independently through CICS transactions CETR or CEMT.

Note: TRACE adds to the overhead experienced by CICS. Only run this option under the direction of CA Top Secret technical support.

You can control the amount of trace data created by specifying one or more of the following keywords when you activate the trace.

Level=1|2|3

Controls the amount of trace data created:

1

(Default) Each trace record contains the trace point ID, the name of the calling program, and the offset into the program. In addition, any specific trace data provided by the calling program appears in the trace record.

2

In addition to the data provided by Level 1, each trace record contains the contents of general registers R0 to R15 at the time the trace call was made.

3

In addition to the data provided by Level 2, each trace record contains a dump of two key control blocks. The control blocks from which you can select are: TRT, PGE, WSB, PGA, and SRT. The TRT and PGE control blocks are traced by default.

To identify which control blocks are to be traced, specify *block*=Y at the end of the command. Replace block with the name of the control block to be traced.

MODule=name

Specify a module name to trace only those calls made by the specified module.

EVent=xxyy

Specify an ENF event ID to trace only those calls made for the specified event.

TRAns=name

Specify a transaction ID to trace only those calls made by the specified transaction.

TERm=name

Specify a terminal ID to trace only those calls made by the transactions running on the specified terminal.

ENtry=name

Specify an entry ID so that only the specified trace point creates trace records.

DUMP=traceid

This function is to be used at the request of support and produces an XPI dump on the first entry to that traceid.

The following example shows how to trace only a program named CAKSROUT with detail at level 2.

```
TSEU=TRACE=ON,LE=2,MOD=CAKSROUT
```

The next example shows how to trace only CEMT transactions with detail at level 3, which will include a dump of WSB and SRT on each trace record. Control block tracing is only available with LE=3 tracing.

```
TSEU=TRACE=ON,LE=3,TRANS=CEMT,WSB=Y,SRT=Y
```

Chapter 7: Using the CA Top Secret Administration Panels

This section contains the following topics:

Installing Administration Menus

The TSS command can be used under CICS to perform all security administration online. The syntax of the TSS command is identical to that used on all other facilities. A full set of CA Top Secret security administration panels are provided for CICS.

Prerequisites

CA-C Runtime r3.1, CAILIB, CAICICS, and the CA Top Secret CAILIB must be contained in the DFHRPL.

You must update both the CA Top Secret and the CA-C Runtime PROGRAM and TRANSACTION definitions.

Panel Installation

If you installed transactions and programs through CAI.CAKOJCL0(TSSCSD), the administration panels are already defined. For manual table entries, see the appendix, "CSD PROGRAM and TRANSACTION Sample Entries."

Accessing the Administration Menu

The Administration Menus have been split into two transactions:

- PTSM-Administration Menu for Modify Commands (Control Options)
- PTSS-Administration Menu for Other Commands

This new granularity allows room for a less cluttered design. To initiate either of these transactions, sign on (CESN) as an administrator with the correct access to successfully issue the commands. When you issue either of these transactions, you can navigate the menus to generate commands.

PTSS Transaction

The PTSS transaction invokes the CA Top Secret administration panels.

After entering the CICS PTSS transaction, the main menu displays.

The next panel indicates how to set up a TSS WHOAMI command to use the option, Display TSS command text:

```
P000001          Security Administration Main Menu        CA TOP SECRET

===>  1

Concerning Me...                Resource Administration

1  Who am I?                    21  Assign/remove resource ownership

2  Lock my terminal             22  Permit/revoke resource access

3  Unlock my terminal           23  Display resource access/ownership

                                24  Certificate Management processing



ACID Administration             System Administration

11  Create ACID                 32  Modify security tables

12  Change ACID attributes

13  Assign administrative authority
14  Display

Options for TSS Administration Session



   ( _ ) List after successful command

   ( _ ) Clear after successful command

   ( X ) Display TSS command text
```

The following are the results of the command as displayed by the PTSS process:

TSS03031 ACIDNAME(CICSPEON) TYPE(USER) MODE(FAIL)

TSS03031 FACILITY(CICSPROD) TERMINAL(A55TU048) LOCKTIME(000)

TSS03031 SYSTEMID(XE57) LOG(ALL)

TSS03001 WHOAMI FUNCTION SUCCESSFUL

By pressing the CLEAR key, the user is returned to the blank P0000001 screen. To exit PTSS, press PF3.

Using the TSS Command Function Panels

Navigation of the Administration Panels is strictly hierarchical. PF3 ends the current panel and returns you to the menu that brought you to the current panel. PF3 issued from the main menu ends the transaction. In some cases it may be necessary to return all the way to the Administration Panel Main Menu in order to activate a selected command. To move forward in the menu hierarchy, you may choose one of several numeric menu choices, displayed on each menu screen.

Because commands have been divided into PTSM and PTSS, it is currently not possible to start a MODIFY command from PTSS, or a non-Control Option command from PTSM.

If using a panel that requires mixed case, enter:

```
EXEC CICS SET TERMINAL NOUCTRAN
```

To reset the terminal back, enter:

```
EXEC CICS SET TERMINAL UCTRAN
```

Appendix A: CSD PROGRAM and TRANSACTION Sample Entries

This section contains the following topics:

Sample Entries for the CA Top Secret Component

The following example shows how to define PROFILE, TRANSACTION, and PROGRAM entries for the CA Top Secret component. The examples can be found in member TSSCSD in data set CAI.CAKOJCL0.

```
DEFINE PROFILE(TOPSPROF) GROUP(TOPSGRP)
         DESCRIPTION(CA Top Secret CICS Interface)
         SCRNSIZE(DEFAULT) UCTRAN(YES)
         PRINTERCOMP(NO) JOURNAL(NO) MSGJRNL(NO)
         MSGINTEG(NO) ONEWTE(NO)
         CHAINCONTROL(NO) DVSUPRT(ALL)
         INBFMH(EODS) RAQ(NO) LOGREC(NO)
         NEPCLASS(0) RTIMOUT(NO)
DEFINE TRANSACTION(PTSM) GROUP(TOPSGRP)
         DESCRIPTION(CA-SECURITY/CICS Administration Menu)
         PROGRAM(TSSCPTSM) TWASIZE(8)
         PROFILE(TOPSPROF) STATUS(ENABLED) DYNAMIC(NO)
         PRIORITY(1) DTIMOUT(NO)
         RESTART(NO) SPURGE(NO)
         TASKDATALOC(BELOW) TASKDATAKEY(USER)
         TPURGE(NO) DUMP(YES) TRACE(YES)
         RESSEC(NO) CMDSEC(NO)
DEFINE TRANSACTION(PTSS) GROUP(TOPSGRP)
         DESCRIPTION(CA-SECURITY/CICS Administration Menu)
         PROGRAM(TSSCPTSS) TWASIZE(8)
         PROFILE(TOPSPROF) STATUS(ENABLED) DYNAMIC(NO)
         PRIORITY(1) DTIMOUT(NO)
         RESTART(NO) SPURGE(NO)
         TASKDATALOC(BELOW) TASKDATAKEY(USER)
         TPURGE(NO) DUMP(YES) TRACE(YES)
         RESSEC(NO) CMDSEC(NO)
DEFINE TRANSACTION(TSS) GROUP(TOPSGRP)
         DESCRIPTION(CA Top Secret Administration Interface)
         PROGRAM(TSSCICS) TWASIZE(0)
         PROFILE(TOPSPROF) STATUS(ENABLED) DYNAMIC(NO)
         PRIORITY(1) DTIMOUT(NO)
         RESTART(NO) SPURGE(NO)
         TASKDATALOC(BELOW) TASKDATAKEY(USER)
         TPURGE(NO) DUMP(YES) TRACE(YES)
         RESSEC(NO) CMDSEC(NO)
DEFINE PROGRAM(TSSCAI) GROUP(TOPSGRP)
         DESCRIPTION(CA Top Secret Application Interface Stub)
         LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(CICS)
         RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO)
         STATUS(ENABLED) CEDF(NO) DATALOCATION(BELOW)
```

```
DEFINE PROGRAM(TSSCAIN) GROUP(TOPSGRP)
         DESCRIPTION(CA Top Secret Application Interface)
         LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(CICS)
         RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO)
         STATUS(ENABLED) CEDF(NO) DATALOCATION(BELOW)
DEFINE PROGRAM(TSSCICS) GROUP(TOPSGRP)
         DESCRIPTION(CA Top Secret Administration Interface Stub)
         LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(USER)
         RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO)
         STATUS(ENABLED) CEDF(NO) DATALOCATION(BELOW)
DEFINE PROGRAM(TSSCICSN) GROUP(TOPSGRP)
         DESCRIPTION(CA Top Secret Administration Interface)
         LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(CICS)
         RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO)
         STATUS(ENABLED) CEDF(NO) DATALOCATION(BELOW)
DEFINE PROGRAM(TSSCPTSM) GROUP(TOPSGRP)
         DESCRIPTION(CA Top Secret Administration Menu Interface)
         LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(USER)
         RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO)
         STATUS(ENABLED) CEDF(NO) DATALOCATION(BELOW)
DEFINE PROGRAM(TSSCPTSS) GROUP(TOPSGRP)
         DESCRIPTION(CA Top Secret Administration Menu Interface)
         LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(USER)
         RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO)
         STATUS(ENABLED) CEDF(NO) DATALOCATION(BELOW)
DEFINE PROGRAM(TSSCPLT) GROUP(TOPSGRP)
          DESCRIPTION(CA-TOP SECRET CICS INITIALIZATION VERIFICATION)
          LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(CICS)
          RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO)
          STATUS(ENABLED) CEDF(NO) DATALOCATION(ANY)
```

To run the TSS command above the line, use the following definition.

```
DEFINE TRANSACTION(TSS) GROUP(TOPSGRP)
         DESCRIPTION(CA Top Secret Administration Interface)
         PROGRAM(TSSCICSN) TWASIZE(0)
         PROFILE(TOPSPROF) STATUS(ENABLED) DYNAMIC(NO)
         PRIORITY(1) DTIMOUT(NO)
         RESTART(NO) SPURGE(NO)
         TASKDATALOC(ANY) TASKDATAKEY(USER)
         TPURGE(NO) DUMP(YES) TRACE(YES)
         RESSEC(NO) CMDSEC(NO)
```

PROFILE Entries for the CICS Component

The following example shows how to define PROFILE entries for the CICS component.

```
DEFINE PROFILE(CAKSPROF) GROUP(CAKSGRP)
         DESCRIPTION(CA-SECURITY/CICS Interface)
         SCRNSIZE(DEFAULT) UCTRAN(YES)
         PRINTERCOMP(NO) JOURNAL(NO) MSGJRNL(NO)
         MSGINTEG(NO) ONEWTE(NO)
         CHAINCONTROL(NO) DVSUPRT(ALL)
         INBFMH(EODS) RAQ(NO) LOGREC(NO)
         NEPCLASS(0) RTIMOUT(NO)
```

TRANSACTION Entries for the CICS Component

This example defines TRANSACTION entries for the CICS component:

```
DEFINE TRANSACTION(TSEU) GROUP(CAKSGRP)
         DESCRIPTION(CA-SECURITY/CICS Environment Utility)
         PROGRAM(CAKSCHEK) TWASIZE(0)
         PROFILE(CAKSPROF) STATUS(ENABLED) DYNAMIC(NO)
         PRIORITY(1) DTIMOUT(NO)
         RESTART(NO) SPURGE(NO)
         TASKDATALOC(ANY) TASKDATAKEY(USER)
         TPURGE(NO) DUMP(YES) TRACE(YES)
         RESSEC(NO) CMDSEC(NO)
DEFINE TRANSACTION(TSLM) GROUP(CAKSGRP)
         DESCRIPTION(CA-SECURITY/CICS Locktime (LTLOGOFF) Monitor)
         PROGRAM(CAKSSCAN) TWASIZE(0)
         PROFILE(CAKSPROF) STATUS(ENABLED) DYNAMIC(NO)
         PRIORITY(1) DTIMOUT(NO)
         RESTART(NO) SPURGE(NO)
         TASKDATALOC(ANY) TASKDATAKEY(USER)
         TPURGE(NO) DUMP(YES) TRACE(YES)
         RESSEC(NO) CMDSEC(NO)
DEFINE TRANSACTION(TSLA) GROUP(CAKSGRP)
         DESCRIPTION(CA-SECURITY/CICS Locktime (LTLOGOFF) Action)
         PROGRAM(CAKSLOCK) TWASIZE(0)
         PROFILE(CAKSPROF) STATUS(ENABLED) DYNAMIC(NO)
         PRIORITY(1) DTIMOUT(NO)
         RESTART(NO) SPURGE(NO)
         TASKDATALOC(ANY) TASKDATAKEY(USER)
         TPURGE(NO) DUMP(YES) TRACE(YES)
         RESSEC(NO) CMDSEC(NO)
DEFINE TRANSACTION(TSLK) GROUP(CAKSGRP)
         DESCRIPTION(CA-SECURITY/CICS Locktime LTLOGOFF Action)
         PROGRAM(CAKSLOCK) TWASIZE(0)
         PROFILE(CAKSPROF) STATUS(ENABLED) DYNAMIC(NO)
         PRIORITY(1) DTIMOUT(NO)
         RESTART(NO) SPURGE(NO)
         TASKDATALOC(ANY) TASKDATAKEY(USER)
         TPURGE(NO) DUMP(YES) TRACE(YES)
         RESSEC(NO) CMDSEC(NO)
DEFINE TRANSACTION(TSUX) GROUP(CAKSGRP)
         DESCRIPTION(CA-SECURITY/CICS Environment Utility)
         PROGRAM(CAKSEXSN) TWASIZE(0)
         PROFILE(CAKSPROF) STATUS(ENABLED) DYNAMIC(NO)
         PRIORITY(1) DTIMOUT(NO)
         RESTART(NO) SPURGE(NO)
         TASKDATALOC(ANY) TASKDATAKEY(USER)
         TPURGE(NO) DUMP(YES) TRACE(YES)
         RESSEC(NO) CMDSEC(NO)
```

PROGRAM Entries for the CICS Component

The following shows how to define PROGRAM entries for the CICS component.

```
DEFINE PROGRAM(CAKSEXSN) GROUP(CAKSGRP)
         DESCRIPTION(CA-SECURITY/CICS Environment Install checks)
         LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(CICS)
         RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO)
         STATUS(ENABLED) CEDF(NO) DATALOCATION(ANY)
DEFINE PROGRAM(CAKSCHEK) GROUP(CAKSGRP)
         DESCRIPTION(CA-SECURITY/CICS Environment Utility Driver)
         LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(CICS)
         RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO)
         STATUS(ENABLED) CEDF(NO) DATALOCATION(ANY)
DEFINE PROGRAM(CAKSINST) GROUP(CAKSGRP)
         DESCRIPTION(CA-SECURITY/CICS Environment Install checks)
         LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(CICS)
         RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO)
         STATUS(ENABLED) CEDF(NO) DATALOCATION(ANY)
DEFINE PROGRAM(CAKSMAXT) GROUP(CAKSGRP)
         DESCRIPTION(CA-SECURITY/CICS Environment Storage usage)
         LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(CICS)
         RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO)
         STATUS(ENABLED) CEDF(NO) DATALOCATION(ANY)
DEFINE PROGRAM(CAKSNEWC) GROUP(CAKSGRP)
         DESCRIPTION(CA-SECURITY/CICS Environment Refresh maintenance)
         LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(CICS)
         RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO)
         STATUS(ENABLED) CEDF(NO) DATALOCATION(ANY)
DEFINE PROGRAM(CAKSTERM) GROUP(CAKSGRP)
         DESCRIPTION(CA-SECURITY/CICS Environment Terminal analysis)
         LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(CICS)
         RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO)
         STATUS(ENABLED) CEDF(NO) DATALOCATION(ANY)
DEFINE PROGRAM(CAKSTRAC) GROUP(CAKSGRP)
         DESCRIPTION(CA-SECURITY/CICS Environment Trace interface)
         LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(CICS)
         RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO)
         STATUS(ENABLED) CEDF(NO) DATALOCATION(ANY)
DEFINE PROGRAM(CAKSTRAN) GROUP(CAKSGRP)
         DESCRIPTION(CA-SECURITY/CICS Environment Transaction analysis)
         LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(CICS)
         RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO)
         STATUS(ENABLED) CEDF(NO) DATALOCATION(ANY)
DEFINE PROGRAM(CAKSWHOS) GROUP(CAKSGRP)
         DESCRIPTION(CA-SECURITY/CICS Environment Signed on users)
         LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(CICS)
         RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO)
         STATUS(ENABLED) CEDF(NO) DATALOCATION(ANY)
```

```
DEFINE PROGRAM(CAKSWRIT) GROUP(CAKSGRP)
         DESCRIPTION(CA-SECURITY/CICS Environment Transaction analysis)
         LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(CICS)
         RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO)
         STATUS(ENABLED) CEDF(NO) DATALOCATION(ANY)
DEFINE PROGRAM(CAKSSCAN) GROUP(CAKSGRP)
         DESCRIPTION(CA-SECURITY/CICS Locktime Logoff Monitor)
         LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(CICS)
         RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO)
         STATUS(ENABLED) CEDF(NO) DATALOCATION(ANY)
DEFINE PROGRAM(CAKSLOCK) GROUP(CAKSGRP)
         DESCRIPTION(CA-SECURITY/CICS Locktime Logoff Action Interface)
         LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(CICS)
         RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO)
         STATUS(ENABLED) CEDF(NO) DATALOCATION(ANY)
```

Appendix B: CICS Installation Checklist

Use this checklist when you are installing and implementing CA Top Secret security using CICS.

- CAIENF Considerations

  The CA Top Secret CICS interface requires the CA Common Services for z/OS CAIENF (Event Notification Facility) service to be installed and activated. CAIENF/CICS performs CA Top Secret CICS intercepts and drives CA Top Secret CICS during security-related events. Without CAIENF, CA Top Secret CICS does not function.

  To ensure that CAIENF and CA Top Secret are installed correctly, you should review the following:

  - Confirm that the TSSENFDC JCL to define CA Top Secret to the CAIENF database was run.
  - List the CAIENF database by running the CAS9DB ENF utility with the LISTDB() DETAIL control statement. Output should show KO50DCM2, CAS9DCM0, and CAS9DCM2.
  - Verify that the CICS MODE(ON) and the CICSREL(xx) options have been coded in the CAIENF startup parameters. All releases of CICS that will use CA Top Secret need to be specified, such as CICSREL(63).
  - Ensure that CA Top Secret and CAILIB are in the linklist. If CAILIB is not in the linklist, then it must be pointed to in the CICS startup procedure with a CENFLIB DD card.

- Administrative Considerations:
  - Define a region control ACID for the CICS region and associate it with the appropriate MASTFAC parameter. For example:

    ```
    TSS CREATE(cicsnn) NAME('new CICS region acid')
                       FACILITY(BATCH,STC)
                       PASSWORD(NOPW,0)
                       MASTFAC(cicsprod)
                       NOVOLCHK
                       NODSNCHK
                       NORESCHK
                       NOSUBCHK
                       NOLCFCHK

    TSS ADDTO(cicsnn) MASTFAC(cicsprod)
    ```

    If you execute CICS as an STC, the procedure must be added to the STC table and assigned an ACID. For example:

    ```
    TSS ADDTO(STC) PROCNAME(yourproc) ACID(cicsnn)
    ```

  - Define a default user acid. Be careful when assigning attributes, specifically the bypass attributes (NORESCHK, NODSNCHK, NOLCFCHK, and so on) because these can cause a security exposure.
- System Initialization Table (SIT)

  You must specify SEC=YES in the SIT or the Facilities Matrix. For CA Top Secret resource protection allow the following to default or code YES for: XTRAN, XPCT, XFCT, XPPT, XTST, XPSB, XJCT, and XDCT.
- Convert SNT RDM to TSS Commands

  Submit CAKSNMIG Utility. Edit the output as appropriate. Submit edited output through batch TSO. See the “Defining CICS to CA Top Secret” chapter for more information.
- CA Top Secret Supplied Transactions and Programs

  The following CA Top Secret transactions are defined via updates to your CICS TRANSACTION and PROGRAM definitions:
  - TSS command
  - TSEU User Executed Transaction Utility
  - TSS Application Interface
  - TSSTRACK Utility

  Note: Update the CICS TRANSACTION and PROGRAM definition statements to use these CA Top Secret-supplied transactions.

  All CA Top Secret programs listed in the PROGRAM statements must reside in the CICS DFHRPL library.

Whenever an upgrade to CA Top Secret is applied, be sure to update the affected modules for the programs defined in the CSD in the appropriate library in the DFHRPL.

- TSSTRACK Utility

  Allocate the Audit Tracking file to the CICS region.

- ISC/MRO Considerations

  Review to the chapter, “Security for a Multi-System Environment,” prior to implementing security under an MRO and/or ISC environment.

- Starting Your CICS Region

  - Verify that your CICS region ACIDs have the NORESCHK or NOLCFCHK attributes or have been PERMITted to the appropriate transactions.
  - At this point CA Top Secret and CAIENF are installed and active.
  - The following messages are displayed in succession:

```
15.04.24 STC06025 TSS6093I - TSS/CICS Initialization Phase 0 started.
15.04.25 STC06025 TSS6094I - TSS/CICS Initialization Phase 0 complete.
15.05.33 STC06025 +DFHXS0206 - CA-ENF Installing the CICS interface.
15.05.38 STC06025 TSS6000I - TSS/CICS Initialization Phase 1 started.
15.05.38 STC06025 TSS6099I - TSS/CICS Initialization Phase 1 complete.
15.05.38 STC06025 TSS6002I - TSS/CICS Initialization Phase 2 started.
15.05.40 STC06025 TSS6095I - TSS/CICS Signon Manager Subtask is active.
15.05.41 STC06025 TSS6096I - TSS/CICS Attaching 005 Signon Server.
15.05.42 STC06025 TSS6088I - TSS/CICS Core Manager Subtask is active.
15.05.42 STC06025 TSS6088I - TSS/CICS Core Manager Subtask is active.
15.05.46 STC06025 TSS6089I - TSS/CICS Program Manager Subtask is active.
15.05.46 STC06025 TSS6003I - TSS/CICS Initialization Phase 2 complete.
15.05.46 STC06025 TSS6007I - TSS/CICS Security Activated.
```

  - Once the TSS6007I Security Activated message is displayed, the CA Top Secret CICS interface is installed and active in the region.
  - If you do not receive the TSS6093I - TSS/CICS Initialization Phase 0 started message, the product has failed to install, probably because CAKSCINT is not available, the DCM is not installed, or CICSREL parm was not updated.

- CDDE BMS=STANDARD or GREATER

  The CA Top Secret CICS interface must be run with BMS=STANDARD (or GREATER) since the interface uses the BMS SEND TEXT command.

Index

E

F

G

H

I

L

M

N

O

P

Q

R

S

T

U

W