CA-Top Secret®

Implementation: CICS Guide

3.0

VSE

SP1

Second Edition, September 2000

Contents

About This Guide

Purpose

This guide describes how to install the CA-Top Secret security product in your CICS system. The information is arranged in two parts:

- The first part discusses how to install CA-Top Secret when running CICS Release 2.3.
- The second part discusses how to install CA-Top Secret when running CICS Transaction Server (TS) 1.1 and above.

Both parts explain how to secure a CICS intersystem environment, daily operations, customization, and diagnostics.

In addition, both parts of this guide explain how to select and implement CICS security parameters and/or CA-Top Secret suboptions to administer security in your environment. Finally, the CA-Top Secret Application Interface is discussed.

This guide assumes the reader is familiar with CICS concepts and the following CA-Top Secret guides:

User Guide
Planning Guide
Installation Guide
Control Options Guide

The intended audience for the *Implementation: CICS Guide* is systems programmers involved in the installation of CA-Top Secret, and security administrators responsible for implementation.

Organization

PART 1: Running CA-Top Secret With CICS Release 2.3

Chapter	Description
1	Explains how to define CICS to CA-Top Secret, including the Facilities Matrix and administration requirements. Also explains how to install CA-Top Secret in CICS.
2	Describes the implementation control options and CICS FACILITY suboptions.
3	Describes CA-Top Secret security for a CICS intersystem environment.
4	Details how to implement security in your CICS system.
5	Describes the Application Interface and CICS exits.
6	Details CA-Top Secret-supplied transactions.
7	Provides a CICS installation checklist.
8	Contains PPT and PCT sample entries.

PART 2: Running CA-Top Secret With CICS Transaction Server

Chapter	Description
9	Explains how to define CICS to CA-Top Secret, including the Facilities Matrix and administration requirements. Also explains how to install CA-Top Secret in CICS.
10	Describes the implementation control options and CICS FACILITY suboptions.
11	Describes CA-Top Secret security for a CICS intersystem environment.
12	Details how to implement security in your CICS system.
13	Describes the Application Interface and CICS exits.
14	Details CA-Top Secret-supplied transactions.
15	Provides a CICS installation checklist.
16	Contains sample CSD table entries.
Index	Provides an efficient way to locate specific material.

CA-Top Secret Publications

The following publications are supplied with CA-Top Secret:

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

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

Related Publications

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

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

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

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

Command Notation

Enter the following exactly as they appear in command descriptions:

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

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

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

The following table indicates the appropriate command format.

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

Chapter 1. Defining CICS Release 2.3 to CA-Top Secret

This chapter explains how to install CA-Top Secret in your CICS environment. All the tasks necessary to get your CICS system up and running in a secured environment are described. Later chapters give additional information for modifying your security environment.

1.1 Pre-Installation Considerations

The CA-Top Secret CICS interface requires the CA-CIS 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. Refer to the *CA-CIS Installation guide* for detailed information.

1.2 Installing CA-Top Secret in CICS

To install CA-Top Secret in your CICS system, you need to:

- Make sure that CA-Top Secret Release 3.0 and CAIENF are installed and active.

  Note: Ensure that CA-Top Secret and CA-CIS libraries are accessed in LIBDEFs in CICS partition.

- Define CICS to CA-Top Secret.
- Set CICS security parameters in CICS tables.
- Define CA-Top Secret FACILITY suboptions for controlling CICS security processing in the Facility Matrix Table.

- Activate your CICS region. A series of CA-Top Secret messages are displayed indicating the phase of initiation and the security parameters for the region. A list of these messages appear in Chapter 7.

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 batch job in the CA-Top Secret security environment.

1.3 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 may 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. For these reasons, CA-Top Secret allows each CICS region to be associated with a unique facility, several CICS regions to be associated with a common facility, or any combination thereof.

To tell CA-Top Secret about your CICS region, you must associate a 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.**

CA-Top Secret defines a facility as a VSE subsystem that a user must have access to in order to enter the system. For example, Batch and CICS are defined to CA-Top Secret as facilities that must be accessed by users and jobs. All facilities that interface with CA-Top Secret must be defined in the Facilities Matrix.

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 facility-defined CICS regions 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 a list and definitions of the CA-Top Secret FACILITY suboptions, refer to the *Control Options Guide.* For an explanation of how to configure these suboptions for your CICS system, refer to Chapter 4, "Implementing Security".

1.3.1 Providing CICS Default Facilities

CA-Top Secret provides two CICS default facilities—CICSPROD and CICSTEST—that are already defined in the Facilities Matrix. This means that the security attributes that control CA-Top Secret processing for CICSPROD and CICSTEST are predefined. These attributes, listed in the following figures, are actually suboptions of the FACILITY control option. You can use the CICSPROD and CICSTEST default facilities as they are, or you can customize them for your site.

CICSPROD Facility: The defaults for the CICSPROD facility are:

```
INITPGM=DFH          ID=C  TYPE=04
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,NODORMPW,NONPWR,NOIMSXTND
MODE=FAIL  LOGGING=INIT,MSG,SEC9,SMF
UIDACID=8  LOCKTIME=000  DEFACID=*NONE*  KEY=8
FACMATRX=NO         EXTSEC=YES
XJCT=YES  XFCT=YES  XCMD=YES XDCT=YES  XTRAN=YES
XTST=YES  XPSB=YES  XPCT=YES XPPT=YES
PCTEXTSEC=OVERRIDE    PCTCMDSEC=OVERRIDE    PCTRESSEC=OVERRIDE
DSNCHECK=NO   LTLOGOFF=NO
MAXSIGN=10,RETRY                MAXUSER=3000
```

Figure 1-1: CICSPROD Default Attributes

To display the default Bypass List parameters issue:

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

The default parameters appear below.

```
BYPASS: RESOURCE=TRANID    NAMES:   CAQP    CATA    CATD
        CATP      CATR     CAUT     CCMF    CDBD    CDBN
        CDBO      CDBT     CDTS     CECS    CEGN    CEHP
        CEHS      CESF     CESN     CFTS    CGRP    CITS
        CLS1      CLS2     CLS3     CLS4    CMPX    CMSG
        CMTS      COVR     CPLT     CPMI    CSNE    CNPX
        CSM1      CSM2     CSM3     CSM4    CSM5    CSNC
        CSPG      CSPK     CSRK     CSPP    CSPQ    CSPS
        CSRS      CSSC     CSSF     CSSN    CSSX    CSSY
        CSTA      CSTB     CSTE     CSTP    CSTT    CSXM
        CSXX      CSZI     CVMI     CVST    CWTR    CXCU
        CXRT      TS       8888     9999
```

CICSTEST Facility: The defaults for the CICSTEST facility are:

```
INITPGM=DFH          ID=K  TYPE=04
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,NODORMPW,NONPWR,NOIMSXTND
MODE=FAIL  LOGGING=INIT,MSG,SEC9,SMF
UIDACID=8  LOCKTIME=000  DEFACID=*NONE*  KEY=8
FACMATRX=NO         EXTSEC=YES
XJCT=YES  XFCT=YES  XCMD=YES XDCT=YES  XTRAN=YES
XTST=YES  XPSB=YES  XPCT=YES XPPT=YES
PCTEXTSEC=OVERRIDE    PCTCMDSEC=OVERRIDE    PCTRESSEC=OVERRIDE
DSNCHECK=NO   LTLOGOFF=NO
MAXSIGN=10,RETRY                MAXUSER=3000
```

Figure 1-2: CICSTEST Default Attributes

To display the default Bypass List parameters issue:

```
TSS MODIFY FAC(CICSTEST=BYPLIST)
```

The default parameters appear below.

```
BYPASS: RESOURCE=TRANID    NAMES:   CAQP    CATA    CATD
        CATP      CATR     CAUT     CCMF    CDBD    CDBN
        CDBO      CDBT     CDTS     CECS    CEGN    CEHP
        CEHS      CESF     CESN     CFTS    CGRP    CITS
        CLS1      CLS2     CLS3     CLS4    CMPX    CMSG
        CMTS      COVR     CPLT     CPMI    CSNE    CNPX
        CSM1      CSM2     CSM3     CSM4    CSM5    CSNC
        CSPG      CSPK     CSRK     CSPP    CSPQ    CSPS
        CSRS      CSSC     CSSF     CSSN    CSSX    CSSY
        CSTA      CSTB     CSTE     CSTP    CSTT    CSXM
        CSXX      CSZI     CVMI     CVST    CWTR    CXCU
        CXRT      TS       8888     9999
```

1.3.2 Associating a Region With a Facility

You may benefit from creating a separate CA-Top Secret facility for each CICS region, or for a group of related regions. For example, you may wish to group all CICS test regions together and associate them with the same CA-Top Secret facility. This requires a common entry in the Facilities Matrix for the CICS test regions.

When brought up, all CICS regions are associated with the CICSPROD facility by default. You can override this default association by using the MASTFAC attribute on the ACID for bringing up the region.

Suppose, for example, you have a test CICS region called CICST1 that is brought up as a batch job, and you wish to associate this region with the CICSTEST facility. You would

first define the region control ACID that is used for the region, including the MASTFAC parameter, using these commands:

```
TSS CREATE(CICST1) NAME('CICS TEST REGION ACID') FAC(BATCH)
    PAS(NOPW,0) DEPT(deptacid) NOVOLCHK NODSNCHK
    MASTFAC(CICSTEST)
```

Note: The MASTFAC parameter can also be easily added to an existing ACID via the TSS ADD command:

```
TSS ADD(CICS1) MASTFAC(CICSTEST)
```

From this point on, when the CICST1 region is brought up, access to the region is governed by the options defined for the CICSTEST facility. All users who sign on to the region, as well as the resources they use, must be authorized for FACILITY(CICSTEST).

1.3.3 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 may 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, etc.) 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.

1.3.4 Defining the CICS Region Control ACID

Since a CICS region begins its execution as a batch job a CA-Top Secret ACID must be associated with each CICS region. This ACID must be able to access the BATCH facility, and must be authorized to all VSE 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 // ID USER=*acidname* batch JCL statement for the CICS region initiated as BATCH job, or via the CA-Top Secret job submit inheritance.

The following examples suggest ways to associate a CICS region (CICSP1) with either the CICSTEST or CICSPROD default facilities.

CICSPROD

```
TSS CREATE(CICSP1) NAME('CICS PRODUCTION REGION')
    FAC(BATCH) PAS(NOPW,0) DEPT(deptacid)  MASTFAC(CICSPROD)
    NOVOLCHK NODSNCHK NORESCHK NOLCFCHK NOSUBCHK
```

CICSTEST

```
TSS CREATE(CICST1) NAME('CICS TEST REGION')
    FAC(BATCH) PAS(NOPW,0) DEPT(deptacid) MASTFAC(CICSTEST)
```

FAC(BATCH) 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.

PAS(NOPW,0) Turns off password checking.

MASTFAC(facility) Must be specified with the CICS region. Users cannot log on unless MASTFAC is added to their user or profile record; see Chapter 4 for details.

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

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.

Note: 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.

1.4 Using the NOXDEF and XDEF Suboptions

The NOXDEF FACILITY suboption is set by default 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 via the following LCF command:

```
TSS ADD(acid) TRANS(fac,(transaction(s)))
```

Note: By default, CA-Top Secret treats any transaction not included in the TRANS list just like entries in the XTRAN list.

1.5 Administration Requirements

The following sections detail how to define CICS to CA-Top Secret once the installation is complete.

1.5.1 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 ADD command like this:

```
TSS ADD(acid) MASTFAC(facility)
```

1.5.2 Defining CICS

CICS can be defined to CA-Top Secret as a batch job.
An ACID created with FAC(BATCH) allows CICS to execute as a batch job. Therefore, the entries made while adding the CICS region ACID must contain FAC(BATCH). For example:

```
TSS ADD(CICSP1) FACILITY(BATCH)
```

In addition, your systems programmer must code the // ID USER=*acidname* JCL statement, then submit the necessary CICS JCL. Using the example above as a reference, your programmer would code // ID USER=CICSP1 as a batch JCL statement.

1.6 CICS Table Changes

This section describes how to define CICS security parameters for the CA-Top Secret security 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 are required in these CICS tables:

- System Initialization Table (SIT)
- Signon Table (SNT)

Additional CICS Table Entries For initial startup, these additions need to be made to the PCT and PPT tables, as well as to the TSS, TSSC, TSLO, and TSSS transactions. Sample PPT and PCT entries can be found in the Optional Materials file in the members DFHPPTTS and DFHPCTTS.

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

- Destination Control Table (DCT)
- File Control Table (FCT)
- Journal Control Table (JCT)
- Program Control Table (PCT)
- Program Processing Control Table (PPT)
- Terminal Control Table (TCT)
- Temporary Storage Table (TST)

1.6.1 Required Table Changes

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

1.6.1.1 The System Initialization Table (SIT)

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

CA-Top Secret provides an equivalent security suboption in the Facilities Matrix for every SIT security parameter, as well as additional suboptions. If you decide to use CA-Top Secret to define your security for the CICS region, you must set an additional suboption in the CA-Top Secret Facilities Matrix called **FACMATRX**.

Note: Using the Facilities Matrix causes the XPARMs to be static and enhances performance (intercepts are not installed for the XPARMs that are not used).

1.6.1.2 The Signon Control Table (SNT)

The Signon Control Table (SNT) is used for CICS signon security processing. The EXTSEC= parameter must be set on either the DFHSNT TYPE=INITIAL macro, or the DFHSNT TYPE=ENTRY macro.

The EXTSEC security parameter settings are:

EXTSEC= YES|NO

YES Calls CA-Top Secret to validate signon processing for the user.

NO Allows CICS to validate signon processing for the user.

Look at the following examples as a guide for setting the EXTSEC parameters. The example below illustrates the **DFHSNT TYPE=INITIAL** macro.

```
DFHSNT TYPE=INITIAL,
       EXTSEC=YES
DFHSNT TYPE=(ENTRY,DEFAULT),
       PASSWORD=anything
DFHSNT TYPE =FINAL
```

In the example above, **everyone** signing on to CICS will invoke CA-Top Secret for signon processing; this is the recommended setting. The following example illustrates the **DFHSNT TYPE=ENTRY** macro. In this example only Mark Smith signing on to CICS will invoke CA-Top Secret to perform signon processing.

```
DFHSNT TYPE=INITIAL,
DFHSNT TYPE=ENTRY,
       OPNAME='MARK SMITH',
       EXTSEC=YES,
       RSLKEY=(1,2,6,20),
       OPIDENT=DG,
       USERID=GIBSON,
       TIMEOUT=5,
       SCTKEY=(1,2,7,64),
       OPPRTY=128
DFHSNT TYPE =FINAL
```

In the next example using the **DFHSNT TYPE=ENTRY** macro, the Master Terminal C will not invoke CA-Top Secret for signon processing since the EXTSEC security parameter is set to NO, the default.

```
DFHSNT TYPE=INITIAL,
DFHSNT TYPE=ENTRY,
       OPNAME='MASTER TERMINAL C',
       PASSWRD=MASTR,
       RSLKEY=(1,2,6,20),
       OPIDENT=MTC,
       USERID=GIBSON,
       TIMEOUT=5,
       SCTKEY=(1,2,3,7,32,44,64),
       OPPRTY=250
DFHSNT TYPE =FINAL
```

1.6.1.3 CA-Top Secret Defined User Requesting CICS Security

If the user is defined to CA-Top Secret and requests CICS security key checking, the following SNT security parameters and their associated CA-Top Secret keywords must be defined to CA-Top Secret via the ACID for that user.

OPCLASS
OPIDENT
OPPRTY
SCTYKEY

These SNT security parameters have associated CA-Top Secret keywords. For a detailed description of the CA-Top Secret keywords listed above, refer to the *Command Functions Guide.*

Also, the SNT TIMEOUT= security parameter is honored by CA-Top Secret. For details, refer to IBM's *CICS Resource Definition (Macro)* guide.

1.6.2 Optional CICS Table Changes

Changes to these CICS tables are optional: DCT, FCT, JCT, PCT, PPT, TCT, and TST. This section describes the changes to these tables in detail.

For the DCT, FCT, JCT, PPT, and TST, the RSL security parameter is recognized by CA-Top Secret during CICS security key checking. To specify the RSL security parameter in each of these tables, refer to IBM's *CICS Resource Definition (Macro)* or the *CICS Resource Definition (Online)* guides.

Note: For the FCT CICS table changes, if DSN name checking is requested by specifying DSNCHECK(YES) via the FACILITY suboption, CICS security key checking **is not** performed for the FCT resource class.

1.6.2.1 The Program Control Table (PCT)

The Program Control Table (PCT) contains information for identifying and initializing transactions and for interval control started tasks.

If PCT security is set on the EXTSEC macro, then:

EXTSEC= YES|NO

YES Invokes CA-Top Secret to determine if you have been authorized access to the selected transaction, and performs resource (OTRAN) or LCF checking.

NO Performs CICS security key checking to determine if you have been authorized access to the selected transaction.

Specify the RSLC security parameter of the PCT as follows:

RSLC= **NO|YES|EXTERNAL**

NO Bypasses resource security checking.

YES Performs CICS resource security checking. Access to each resource via this transaction depends on the RSL value in the macro entry or resource definition for the resource. If the resource being checked has an RSL value of:

- PUBLIC, then access is allowed to any user.
- 0, access is denied.
- From 1 to 24, then the value is checked against either: the RSLKEY values from the user's SNT, the OPERRSL value specified on the user's terminal definition, or the security keys added to the ACID via the SCTYKEY attributes. If a match is found, the transaction allows access to the resource.

EXTERNAL Performs CA-Top Secret resource security checking.

In the PCT, specify SPURGE=NO for the CSSN/CESN and CSSF/CESF transactions.

Note: The FACILITY suboption PCTEXTSEC controls the PCT EXTSEC parameter. If the PCTEXTSEC suboption is set to **OVERRIDE** CA-Top Secret always performs LCF/OTRAN security checking against the transaction ID.

1.6.2.2 The Terminal Control Table (TCT)

The Terminal Control Table (TCT) contains terminal ID information.

Note: Devices can be made receive-only by setting ATI=YES and TTI=NO.

The following TCT security parameters may be defined. Refer to IBM's *CICS Resource Definition (Macro)* and the *CICS Resource Definition (Online)* guides for specific details.

For the DFHTCT TYPE=TERM settings:

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 Has an associated CA-Top Secret keyword, OPPRTY. Refer to the *Command Functions Guide* for a detailed description of the OPPRTY keyword.

OPERRSL It is recommended that you do not specify this parameter if CA-Top Secret is performing security checking. Devices defined as printers are automatically bypassed for CA-Top Secret security checking. Other devices must be defined to CA-Top Secret. Refer to Chapter 4 "Implementing Security" for details.

OPERSEC Use the default setting of 1 only.

SIGNOFF This security parameter is honored by CA-Top Secret.

USERID The specified userid will be 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: TCT output-only definitions are not protected terminals. ATS will not be used for an output-only terminal.

1.6.3 Activating CA-Top Secret Security

Once CA-Top Secret is installed, it is present in every CICS region. CA-Top Secret runs in one of two states: active or inactive. The *active* state means that CA-Top Secret is actively performing security checks. The *inactive* state means that CA-Top Secret is present, but running in standby state; no CA-Top Secret security checking is performed. This functionality allows you to decide whether you want to use CICS security key checking for the region or CA-Top Secret security.

To use CA-Top Secret to secure your region, you must activate CA-Top Secret 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 EXTSEC security parameter in the DFHSIT to YES (see the next heading "SIT Security Parameter Settings", for details).
- 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.

To use the facility matrix override feature, enter:

```
TSS MODIFY FAC(CICSPROD=FACMATRX=YES)
```

Note: Remember that these FACILITY suboptions can be changed **dynamically** at any time without reinitializing your CICS region. Refer to the *Control Options Guide* for details on how to change control options.

To use CICS security key checking, you must set CA-Top Secret to the inactive state. To set CA-Top Secret **inactive**, specify one of the following combinations:

In DORMANT mode:

- Set the the DFHSIT EXTSEC parameter to NO, or
- Set the CA-Top Secret FACMATRX suboption to YES and the EXTSEC suboption to NO.

In WARN, IMPLEMENT, and FAIL modes:

- Set the DFHSIT EXTSEC parameter to NO and enter the SYSID for the region in the SYSID Bypass List, or
- Set the CA-Top Secret FACMATRX suboption to YES, the EXTSEC suboption to NO, and add an entry for the region to the SYSID Bypass List.

1.6.3.1 SIT Security Parameter Settings

SIT security parameter settings recognized by CA-Top Secret are listed below. Any other settings are not recognized. Remember that these settings only provide CA-Top Secret security for the region. They **do not** control user signon. (See the Signon Control Table (SNT) changes for CA-Top Secret signon processing.)

Note: CA-Top Secret CICS provides identical parameters when the FACMATRX suboption is set to YES.

EXTSEC= YES|NO

YES CA-Top Secret security is active for this region.

NO CA-Top Secret security is inactive, but still present in this region; the default. For security to remain inactive in a mode other than DORMANT, add a SYSID entry to the SYSID Bypass List for the facility. Refer to the section on Facility Bypass List suboptions in Chapter 2.

XCMD= YES|NO

YES All SPI commands are checked by CA-Top Secret.

NO All SPI commands are not checked by CA-Top Secret.

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

Note: The XCMD parameter does not exist in CICS **Release 2.3** and below. However, by default, CA-Top Secret provides SPI command protection for these releases. If you want to turn off default SPI protection for CICS Release 2.3, you must specify FACMATRX=YES and XCMD=NO.

XDCT= YES|NO

YES Transient data entries for this region are checked by CA-Top Secret.

NO Transient data entries for this region are not checked by CA-Top Secret.

XFCT= YES|NO

YES File control entries for this region are checked by CA-Top Secret.

NO File control entries for this region are not checked by CA-Top Secret.

XJCT= YES|NO

YES Journal control entries for this region are checked by CA-Top Secret.

NO Journal control entries for this region are not checked by CA-Top Secret.

XPCT= YES|NO

YES EXEC CICS START transactions for this region are checked by CA-Top Secret. This also includes background transaction (non-terminal) protection.

NO Non-terminal transactions for this region are not checked by CA-Top Secret.

XPPT= YES|NO

YES Program entries for this region are checked by CA-Top Secret.

NO Program entries for this region are not checked by CA-Top Secret.

XPSB= YES|NO

YES Database PSB entries for this region are checked by CA-Top Secret.

NO Database PSB entries for this region are not checked by CA-Top Secret.

XTRAN= YES|NO

YES Terminal attached transaction entries for this region are checked by CA-Top Secret.

NO Terminal attached transaction entries for this region are not checked by CA-Top Secret.

XTST= YES|NO

YES Temporary storage keys for this region are checked by CA-Top Secret.

NO Temporary storage keys for this region are not checked by CA-Top Secret.

XUSER= YES|NO

YES Performs surrogate user checking.

NO Does not perform surrogate user checking.

Note: The XUSER parameter does not exist in CICS **Release 2.3**. However, by default, CA-Top Secret provides SPI command protection for these releases. If you want to turn off default SPI protection for CICS Release 2.3, you must specify FACMATRX=YES and XUSER=NO.

1.7 Interactive Interface Signon Compatability

Many sites require some or all of their users to signon to the Interactive Interface (II), an IBM productivity option that facilitates VSE/ESA system administration and provides a pathway into CICS. This section describes three methods for implementing an external security signon for your II users. If you do not have users signing on to the II, skip this section.

External security attributes must be associated with each user that requires their II signon be secured by CA-Top Secret. There are three methods available for implementation, each site may choose the method that best suits their needs. The methods are:

- CICS Signon Control Table (SNT)
- II User Profile Maintenance
- TSSIIEXT Batch Utility

Details on each method appear in the following sections. Any II user that does not have the external security attribute assigned may continue to signon using the VSE Control File userid and password but will not have a CA-Top Secret security profile established for their session. This may lead to problems accessing various CICS resources, such as transactions and programs, based on the active security mode.

1.7.1 CICS Signon Control Table (SNT)

Each user of the Interactive Interface may or may not have a CICS Signon Control Table (SNT) entry. For those users that do have an explicit SNT entry, the external security parameter can be set within that SNT entry. Refer to 1.6, “CICS Table Changes,” for instructions on setting the EXTSEC= parameter on a DFHSNT TYPE=ENTRY macro.

Note: This method only works for II users that have an EXPLICIT SNT entry, making external security active for all CICS users on the DFHSNT TYPE=INITIAL macro will have no effect on the II signon process.

1.7.2 II User Profile Maintenance

You can provide the external security attribute to any defined user via II User Profile Maintenance. The attribute can only be assigned through online panel maintenance, the VSE/ESA batch utility IESUPDCF does not provide a keyword for updating this attribute. Follow path '211' to the Maintain User Profile panel, and specify External Security = 1 (YES). Do this for each user that will require an external security signon via CA-Top Secret. This method is useful during the testing and implementation phase by affecting only a limited number of users. Please refer to the IBM VSE/ESA *Administration Guide* for further information on user profile maintenance.

1.7.3 TSSIIEXT Batch Utility

TSSIIEXT is a batch utility program that will read the current VSE Control File (IESCNTL) and update the external security attribute for each user profile defined to that file. Via a control option, the utility will either turn the attribute on or off for all user profiles. There is no option to selectively update individual profiles. This method should be used when you are ready to fully implement CA-Top Secret for all II users, or for an emergency backout.

The TSSIIEXT JCL: A copy of the TSSIIEXT JCL follows.

```
// JOB TSSIIEXT *** TSS II EXTERNAL SECURITY ***
// ID USER=SYSA,PWD=SYSA
*
* THIS UTILITY ACTIVATES OR DE-ACTIVATES THE EXTERNAL SECURITY
* CONTROL BIT FOR ALL USER PROFILES DEFINED TO THE VSE CONTROL FILE.
*
* UPSI 00—SET BIT OFF
* UPSI 01—SET BIT ON
*
// UPSI 00
// DLBL IESCNTL,'VSE.CONTROL.FILE',,VSAM,CAT=VSESPUC
// EXEC TSSIIEXT
/*
/&
```

Chapter 2. Control Option Requirements

2.1 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 either existing CICS security or CA-Top Secret security is in effect.

2.1.1 Modes of Operation

CA-Top Secret supports four separate modes of operation for a CICS environment, as it does for all facilities. The modes are DORMANT, WARN, IMPLEMENT, and FAIL. Modes are assigned at five different levels:

Global The default for the entire CA-Top Secret community.
Example: **MODE(WARN)**

Facility Affects a particular facility within the community.
Example: **FAC(CICS=MODE=IMPL)**

Profile Affects a particular group of users attached to the profile.
Example: **TSS PER(PROF01) MODE(IMPL)**

User Affects a particular user within the community.
Example: **TSS PER(USER01) MODE(FAIL)**

Resource Forces a particular resource authorization to be processed in FAIL mode. Example:
TSS PER(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.

The following section describes how each CA-Top Secret security mode affects CICS users, based on the type of security checking requested. Modes of operation are listed for both resource and LCF (Limited Command Facility) transactions.

2.1.2 Modes for Resource Level Checking

The tables in this section define how CA-Top Secret modes are administered for resource checking.

The following table explains how modes for users and resources defined to CA-Top Secret are administered.

Table 2-1. Modes for Users Defined to CA-Top Secret—Resources Defined to CA-Top Secret

MODE	**ACTION**
DORMANT	Only CICS security key checking is performed.
WARN	If the user is permitted access, security checking is performed by CA-Top Secret only. CICS security key checking is not performed. If the user is not permitted access to the resource, a warning message is issued to the user and CICS security key checking is performed.
IMPLEMENT	Security checking is performed by CA-Top Secret only. CICS security key checking is not performed.
FAIL	Security checking is performed by CA-Top Secret only. CICS security key checking is not performed.

In addition to the information contained in the previous table, 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.

  For more information about these control options, refer to the *Control Options Guide.*

The next table explains how modes for users defined to CA-Top Secret and resources **not** defined to CA-Top Secret are administered.

Table 2-2. Modes for Users Defined to CA-Top Secret—Resources Not Defined to CA-Top Secret

MODE	**ACTION**
DORMANT	Only CICS security key checking is performed.
WARN	No security checking is performed. If default protection is specified, a warning message is issued to the user, and CICS security checking is performed as well.
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. No CICS security key checking is performed.
FAIL	Only CICS security key 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. No CICS security key checking is performed.

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.

  For more information about these control options, refer to the *Control Options Guide.*

This table explains how modes for users **not** defined to CA-Top Secret and resources that are defined to CA-Top Secret are administered.

Table 2-3. Modes for Users Not Defined to CA-Top Secret—Resources Defined to CA-Top Secret

MODE	ACTION
DORMANT	CICS security key checking only is performed.
WARN	If the resource is defined, and access is permitted via the TSS ALL record, security checking is performed by CA-Top Secret only. No CICS security key checking is performed. If the resource is defined, and access is not permitted via the TSS ALL record, CICS security key checking is performed.
IMPLEMENT	Security checking is performed by CA-Top Secret only. No CICS security key checking is performed.
FAIL	Undefined users are not allowed system entry.

This table explains how modes for users and resources **not** defined to CA-Top Secret are administered.

Table 2-4. Modes for Users Not Defined to CA-Top Secret—Resources Not Defined to CA-Top Secret

MODE	ACTION
DORMANT	CICS security key checking only is performed.
WARN	CICS security key checking only is performed.
IMPLEMENT	CICS security key checking only is performed.
FAIL	Undefined users are not allowed system entry.

2.1.3 Modes for LCF Checking

The following tables represent CA-Top Secret modes of operation for protection of transactions via the Limited Command Facility (LCF). Both inclusive and exclusive LCF lists are protected by CA-Top Secret.

- 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

Refer to the *User Guide* for a complete explanation of CA-Top Secret LCF protection.

Note: Transactions defined as OTRAN transactions override LCF transactions and follow the guidelines documented in the previous section, "Modes for Resource Level Checking".

The following table explains how modes for inclusive LCF lists are administered.

Table 2-5. Modes for Users Defined to CA-Top Secret for TRANS

MODE	ACTION
DORMANT	Only CICS security key checking is performed.
WARN	If the user has a TRANS LCF list for the facility, and the transaction ID accessed is found in that list, security checking is performed by CA-Top Secret only. CICS security key checking is not performed. If the transaction ID accessed is not found in the TRANS LCF list for the facility, a warning message is issued and CICS security key checking is performed.
IMPLEMENT	If the user has a TRANS LCF list, for the facility, and the transaction ID accessed is found in that list, security checking is performed by CA-Top Secret only. In all cases, CICS security key checking is not performed.
FAIL	If the user has a TRANS LCF list for the facility, and the transaction ID accessed is found in that list, security checking is performed by CA-Top Secret only. In all cases, CICS security key checking is not performed.

Here is an example of an LCF inclusive list:

```
TSS ADD(acid) TRANS(CICSPROD,(PAYT,MAIL,PAYP))
```

The following table explains how modes for exclusive LCF lists are administered.

Table 2-6. Modes for Users Defined to CA-Top Secret for XTRANS

MODE	ACTION
DORMANT	If the user is defined to CA-Top Secret, CICS security key checking is performed.
WARN	If the user specifies an XTRANS LCF list, and the transaction accessed is not found in the list for the facility, security checking is performed by CA-Top Secret only. CICS checking is not performed. If the transaction accessed is not found in the XTRANS LCF list for the facility, a warning message is issued to the user and CICS security key checking is performed.
IMPLEMENT	If the user is defined to CA-Top Secret, and the transaction is found in the XTRANS LCF list, the user is failed.
FAIL	If the user is defined to CA-Top Secret, and the transaction is found in the XTRANS LCF list, the user is failed.

Here is an example of an LCF exclusive list:

```
TSS ADD(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 by CA-Top Secret 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.

The following table explains how modes for users **not** defined to CA-Top Secret are administered.

Table 2-7. Modes for Users NOT Defined to CA-Top Secret	
MODE	ACTION
DORMANT	CICS security key checking is performed.
WARN	CICS security key checking is performed.
IMPLEMENT	CICS security key checking is performed.
FAIL	Undefined users are not allowed system entry.

2.2 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.

Refer to the *Control Options Guide* for an explanation of how to set CA-Top Secret control options, what they can be used for, and a detailed description of each option.

2.3 CICS FACILITY Suboptions

CICS FACILITY suboptions can be divided into three groups:

- suboptions associated with the FACMATRX suboption,
- suboptions in the Bypass List, and
- additional suboptions.

A list of CA-Top Secret FACILITY suboptions equivalent to the CICS DFHSIT security parameters appears below.

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

FACMATRX Suboptions

EXTSEC
XAPPC
XCMD
XDCT
XFCT
XJCT
XPCT
XPPT
XPSB
XTRAN
XTST
XUSER

Bypass List Suboptions

BYPADD(resource)
BYPREM(resource)
BYPLIST

Additional Suboptions

DSNCHECK
LTLOGOFF
MAXSIGN
MAXUSER
PCTCMDSEC
PCTEXTSEC

For definitions, syntax and usage for each option, refer to the *Control Options Guide.*

2.3.1 Using Suboptions or DFHSIT Parameters

CA-Top Secret allows you to make a choice about 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. CA-Top Secret provides a FACILITY suboption that you can set to indicate whether you are using the DFHSIT security parameters or the CA-Top Secret FACILITY suboptions. This suboption is called FACMATRX.

- If you want to use the CA-Top Secret FACILITY suboptions to implement security for CICS initialization, you must set the FACMATRX suboption to YES. Then, security is implemented by the CA-Top Secret equivalent FACILITY suboptions.
- If you want to use the DFHSIT security parameters to implement security for CICS initialization, you must 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.

  Note: You can also selectively disable CAIENF/CICS calls for CICS resources that are not protected by CA-Top Secret to eliminate unnecessary overhead. Details appear in the section "Selectively Disabling CAIENF/CICS Calls".

Both the DFHSIT security parameters and their CA-Top Secret equivalent FACILITY suboptions are listed next. For a complete description of how to use the DFHSIT security parameters, refer to IBM's *Resource Definition (Macro) Guide.* See the *Control Options Guide* for complete information on how to specify FACILITY suboptions.

DFHSIT Parameters	FACILITY Suboptions
	FACMATRX
EXTSEC=	EXTSEC=
XAPPC=	XAPPC=
XCMD=	XCMD=
XDCT=	XDCT=
XFCT=	XFCT=
XJCT=	XJCT=
XPCT=	XPCT=
XPPT=	XPPT=
XPSB=	XPSB=
XTRAN=	XTRAN=
XTST=	XTST=
XUSER=	XUSER=

2.3.1.1 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. Perform the following steps to disable CAIENF/CICS calls.

1. The facility entry must have FACMATRX set to YES.
2. 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.

Note: This procedure makes the Facilities Matrix entry XPARMs static. Therefore, to change a particular region, use the TSS Modify command to make the necessary changes to the FACILITY entry and then recycle the region so all changes appear.

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

Note: **For CICS Release 2.3**, the DCT intercepts are automatically installed so that CA-Top Secret can determine when the system is shutting down.

2.3.2 Bypass List Suboptions

CA-Top Secret Bypass List suboptions for CICS allow you to:

- Set CA-Top Secret security inactive
- Bypass security for certain CEMT commands
- Bypass security for INQUIRE and SET commands
- Bypass security for transactions
- Bypass security for terminals
- Bypass LOCKTIME checking for certain terminals or transactions
- Selectively set security for specific resources

If you put an entry in a CA-Top Secret Bypass List, CA-Top Secret **does not** perform security checking on that resource. If the entry is not on the Bypass List, then it is protected by default.

Note: The protect list under CICS is **only** in effect if you are running CA-Top Secret Release 3.0 with CICS TS 1.1 and above. You can view the protect list if you are running CICS Release 2.3, but any changes made will not be honored.

To add an entry to the Bypass List for CICS, use the BYPADD FACILITY suboption.

Note: Resource names added to the Bypass List are interpreted as generic prefixes.

To remove an entry from the Bypass List for CICS, use the BYPREM FACILITY suboption.

Note: The GENERIC/NONGENERIC RDT attribute has no effect on resources when checking the CICS Bypass List.

The following parameters belong to the Bypass List and can be used with the BYPADD and BYPREM suboptions.

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

These parameters are discussed in Chapter 4, "Implementing Security".

2.3.2.1 Setting CA-Top Secret Security Inactive

Use the SYSID=*sysid* parameter that contains the system identification names of the CICS systems that you do not want CA-Top Secret to secure. An example of how this parameter would be specified is shown below.

```
TSS MODIFY FAC(cicsfac=BYPADD(SYSID=sysid))
```

cicsfac Replace with the type of system: CICSPROD for a production system or CICSTEST for a test system.

sysid Replace with the region's system identification name from the SIT.

Note: If you have coded EXTSEC=NO in either the DFHSIT parameter or the FACMATRX suboption, an entry must be added to the Bypass List

2.3.2.2 Bypassing Security for CEMT Commands

Use the CEMT=*action* parameter that contains the Extended Master Terminal Command actions for which you want to bypass security checking. Valid actions are:

ADD
INQUIRE
PERFORM
REMOVE
SET

For example, to allow access to all CEMT INQUIRE commands, enter:

```
TSS MODIFY FAC(cicsfac=BYPADD(CEMT=INQUIRE))
```

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

2.3.2.3 Bypassing Security for SPI Commands

Use the SPI resource to bypass security for SPI commands.

For example, to bypass **all** EXEC CICS and CEMT INQUIRE SYSTEM commands, enter:

```
TSS MODIFY FAC(cicsfac=BYPADD(SPI=SYSTEM))
```

2.3.2.4 Implementing SPOOLWRITE

If you require SPOOLWRITE to run effectively, you must follow these rules:

- SPI protection must be active (XCMD=YES).
- The transaction issuing SPOOLWRITE commands cannot be in the Bypass List.
- SPI=SET and CEMT=SET cannot be in the Bypass List.
- Bypass security checks for PWRSPOOL by specifying SPI=PWRSPOOL in the Bypass List.
- The userid must contain the value INTRDR.

2.3.2.5 Bypassing Transaction Security

To bypass transaction security, add an entry to the TRANID or TRAN parameter of the CA-Top Secret Bypass List. The TRANID parameter contains transaction ID entries that will bypass **all** security checking for the transaction. The default entries are:

CAQP, CATD, CATP, CAUT, CBRC, CCMF, CESF, CECS, CESC, CESN, CLS1, CLS2, CMSG, CMXP, CSNE, CNPX, CRDR, CQRY, CRDR, CRSQ, CRSR, CRSY, CRSR, CRTE, CRTR, CLSU, CSAC, CSCY, CSGM, CSGX, CSFU, CSGM, CSIR, CSGX, CSJC, CSKP, CSLG, CSL3, CSMI, CSM1, CSM2, CSM3, CSM4, CSM5, CSNC, CSPG, CSPK, CSRK, CSPP, CSPQ, CSPS, CSRS, CSSC, CSSF, CSSN, CSSX, CSSY, CSTA, CSTB, CSTE, CSTT, CSXX, CVST, CWTR, CWTO, CXCU, CXRE, CXRT, TSLO, TSLK, TSSS, TSLA, TSLM, 8888, 9999.

Multiple transactions (up to four) can be specified on one line for the bypass list, by entering:

```
TSS MODIFY FAC(cicsfac=BYPADD(TRANID=(trn1,trn2,trn3,trn4)
```

The difference between the Bypass List parameters TRAN and TRANID is that the entries for TRAN contain TRANIDs that will bypass resource OTRAN or LCF security checking only. The difference between the Bypass List parameters TRAN and TRANID is that the entries for TRAN contain TRANIDs that will bypass resource OTRAN or LCF security checking only. Entries in the TRANID Bypass List contain TRANIDs that will bypass all types of security checking (OTRAN, LCF, FCT, or any type of resource check, including LOCKTIME).

You can use the PCTEXTSEC parameter to bypass LCF or OTRAN (resource) transaction security. If the FACILITY suboption PCTEXTSEC parameter is set to HONOR, CA-Top Secret honors the PCT parameter EXTSEC= settings. To bypass both LCF and OTRAN security checking, the transaction must have both the PCTEXTSEC FACILITY suboption set to HONOR, and the PCT parameter EXTSEC set to NO.

Also note the following information about PCTEXTSEC:

- The FACILITY suboption PCTEXTSEC controls the PCT EXTSEC parameter. If the PCTEXTSEC suboption is set to **OVERRIDE,** CA-Top Secret always performs LCF/OTRAN security checking against the transaction ID.

2.3.2.6 Bypassing Terminal Security

The TCT parameter 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 FAC(cicsfac=BYPADD(TCT=K06L3544))
```

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

2.3.2.7 Bypassing LOCKTIME Security

The LOCKTIME parameter contains terminal entries or transaction IDs that will not be 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 FAC(CICSTEST=BYPADD(LOCKTIME=K06L3544))
```

To bypass LOCKTIME security for transaction PUBL, enter:

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

2.3.2.8 Bypassing Security for Specific Resources

You can **selectively** bypass security for specific resources using these parameters:

DCT Contains transient data entries that will not be checked by CA-Top Secret.

DSN Contains data sets that will not be checked by CA-Top Secret. The DSNCHECK= suboption must be set to YES. The entries in this Bypass List are not the actual data set names, but the File Control Table entries associated with the data sets.

FCT Contains File Control Table entries (DDnames) that will not be checked by CA-Top Secret. The DSNCHECK= suboption must be set to NO.

JCT Contains Journal Control Table entries (journal names) that will not be checked by CA-Top Secret.

PCT Contains interval control started transaction identifiers that will not be checked by CA-Top Secret.

PPT Contains program entries that will not be checked by CA-Top Secret.

PSB Contains PSB entries that will not be checked by CA-Top Secret.

TRAN Contains transaction identifiers that will not be checked by CA-Top Secret.

TST Contains Temporary Storage entries (queue names) that will not be checked by CA-Top Secret.

2.3.3 Additional Suboptions

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

2.3.3.1 Limiting User Signon Storage

Use the MAXUSER= suboption to limit the amount of storage allocated by CA-Top Secret CICS for user signon storage blocks (USCBs), which are GETVISed at CICS initialization time. The MAXUSER value is used to calculate the number of USCBs 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 USCBs 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) in the CICS Payroll region to 500, you can use the TSS MODIFY command like this:

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

2.3.3.2 Controlling Simultaneous User Signon

Use the MAXSIGN= suboption to restrict the number of simultaneous user signons. This suboption allows you to set a threshold for the number of concurrent signons, and controls the action taken if the threshold is exceeded. The threshold may be manually set in the range of 1 to 50, inclusive; the default value is 10. The action may be set to KILL or RETRY. When KILL is set and more users than the threshold are queued to sign on, additional attempts to sign on are failed. When RETRY is set, additional attempts to sign on are requested to CICS. For example, you can restrict the number of concurrent signons to a CICS facility called CICSPAY to a threshold of 15 by using the TSS MODIFY command like this:

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

The KILL option abends the signon transaction; RETRY requeues the signon transaction. See "CICS Related Facility Suboptions" in the *Control Options Guide*.

Note: The SIGN(S) control option disallows simultaneous logons for the same ACID for each CICS region running under a specified facility. With multiple CICS regions under one facility, a single ACID can sign on, one terminal for each region.

2.3.3.3 Securing Data Set Names Instead of FCTs

Use the DSNCHECK(YES|NO) suboption to tell CA-Top Secret to perform security checking on either 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 DSN name, specify DSNCHECK=YES.

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

The XFCT=YES suboption is required in either case.

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

```
TSS MODIFY FACILITY(CICSP1=DSNCHECK=YES,XFCT=YES,RES)
```

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

```
TSS MODIFY FAC(CICSP1=DSNCHECK=NO,XFCT=YES,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,XFCT=YES,RES)
```

Note: If the FCT is remote, all DSN checks will be bypassed. You must remove the CSMI transaction from the FACILITY Bypass List to provide protection for remote DSNs. Security checking is performed in the region where the FCT resides.

CICS data set protection (DSNCHECK=YES) does not protect DL1 databases. These are validated under the CICS region control ACID.

2.3.3.4 Securing Transactions Not Associated With a Terminal

To secure transaction initiation for transactions not associated with a terminal either:

- Use the XPCT security parameter in the DFHSIT, or
- Specify both XPCT=YES and FACMATRX=YES in the FACILITY suboptions.

If you specify XPCT=YES in the DFHSIT or in the FACILITY suboption, background transactions inherit the security authorizations associated with the ACID invoking the transaction. All access authorizations, ownership, user records, etc., associated with the ACID now secure the issued transaction as well.

Transactions scheduled via ATI are security checked when a user attempts to schedule a transaction. If XPCT=YES is coded in the DFHSIT or if XPCT=YES and FACMATRX=YES FACILITY suboptions are specified, the userid (ACID) will be inherited (or associated) with the transaction when it executes.

Note: For transactions not associated with a terminal initiated out of the PLT phase of CICS startup (or related transactions) an entry must be made in the TRANID Bypass List.

With transactions or normal MRO and ISC links, the userid (ACID) signed on in the local (TOR) region is the userid (ACID) inherited and validated in the destination (AOR) region.

If you want background transactions running in the Payroll Account 1 region to execute using the authorities of the issuer of the transaction, you can code XPCT=YES in the DFHSIT or issue a CA-Top Secret command like this:

```
TSS MODIFY FACILITY(PAYACCT1=FACMATRX=YES,XPCT=YES)
```

If you code XPCT=NO, no security checking is performed to determine if the ACID has permission to start the background transaction. Once the transaction is started, resource checking is performed.

If you want background transactions to run **without** security, you can code XPCT=NO in the DFHSIT or issue a CA-Top Secret command like this:

```
TSS MODIFY FACILITY(PAYACCT1=FACMATRX=YES,XPCT=NO)
```

2.3.3.5 Selecting CA-Top Secret Security for Transaction IDs

You can use the PCTEXTSEC= suboption in one of two ways:

- Override the DFHPCT EXTSEC setting and force a CA-Top Secret security check against the transaction ID for **all defined** transactions, or
- Perform selective security checking against the transaction ID for **specific** transactions by honoring the DFHPCT EXTSEC= parameter settings. OVERRIDE is the default value.

To perform security checking against **all** transaction IDs enter:

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

To selectively perform security checking on transaction IDs in the Payroll Account 1 region (and honor the DFHPCT EXTSEC= parameter settings) use this command:

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

Note the following items:

- The DFHPCT macro can be replaced by RDO transaction processing. If you are using RDO, you can substitute "RDO transaction EXTSEC" for the "DFHPCT EXTSEC" in the above explanations.
- The FACILITY suboption PCTEXTSEC overrides the DFHPCT EXTSEC parameter. If the PCTEXTSEC suboption is set to **OVERRIDE,** CA-Top Secret always performs LCF/OTRAN security checking against the transaction ID.

2.3.4 Transaction Validation

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

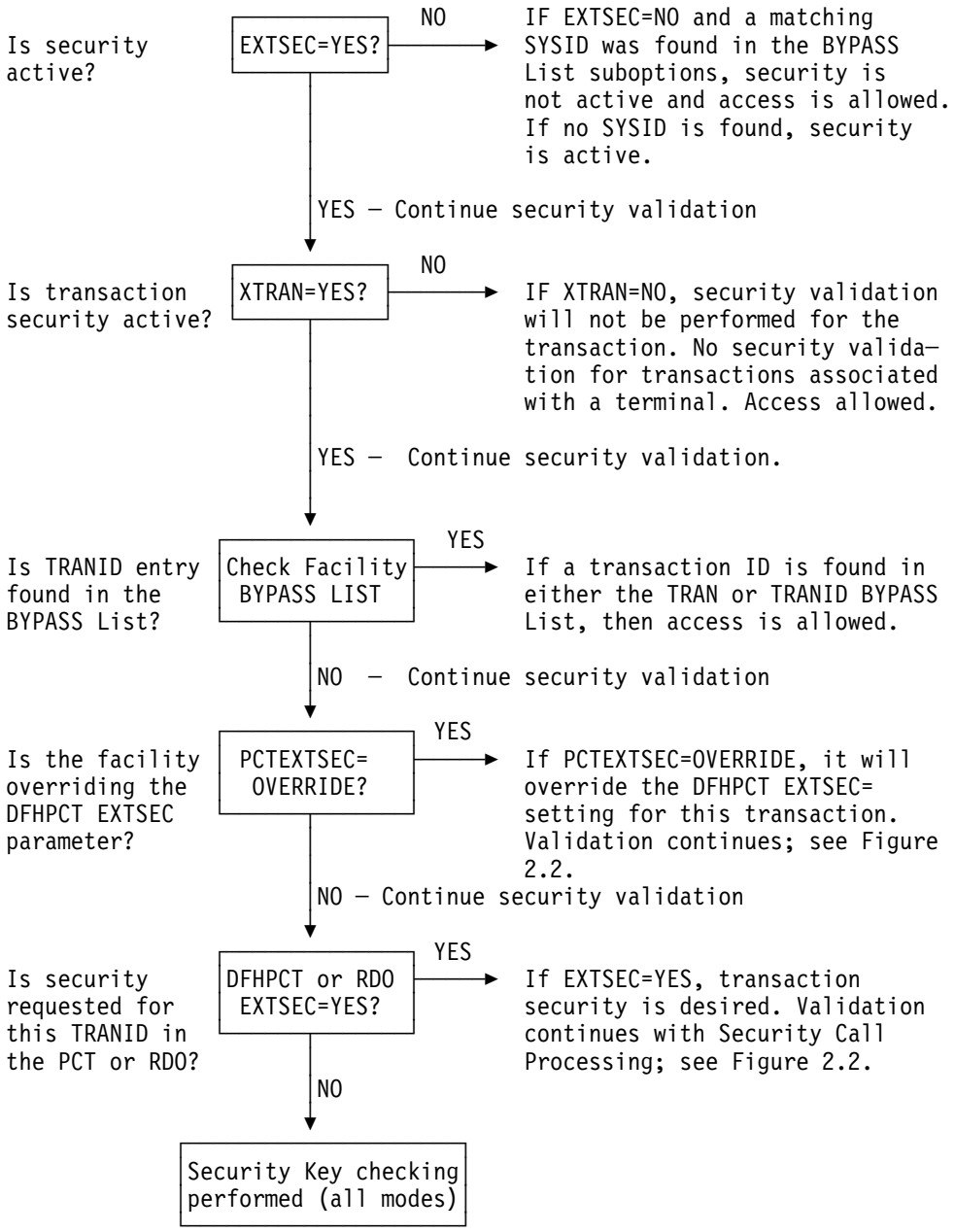


Figure 2-1. Transaction Validation Logic Flow

The following chart illustrates CA-Top Secret CICS security call processing.

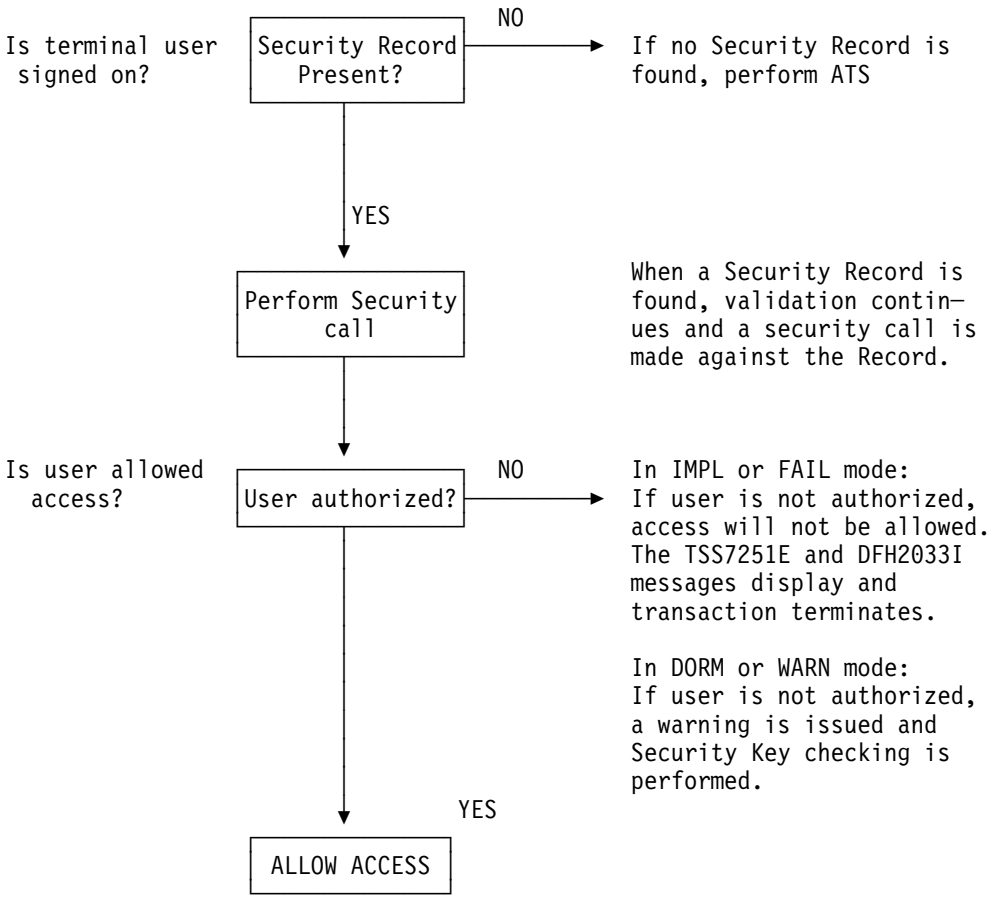


Figure 2-2. Security Call Processing

Chapter 3. Security for a Multi-system Environment

This chapter details CICS and CA-Top Secret security for Intersystem Communication (ISC) and Multiregion Operation (MRO) environments.

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, read the chapter on "Security in the Intercommunication Environment" in IBM's *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

Note: 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 more information on using TSSUTIL, refer to your *Report and Tracking Guide.*

3.1 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 chart shows which CICS parameters must be defined (via RDO or RDM) to set up MRO or ISC in your CICS region.

Definition	Using RDO	Using RDM
CONNECTION	SECURITYNAME	XSNAME
	ATTACHSEC	USERSEC
SESSION	OPERSECURITY	OPERSEC
	OPERRSL	OPERRSL

Note: Discussions in this chapter refer to the equivalent security name definitions for both RDO SECURITYNAME and RDM XSNAME as: **securityname**.

3.2 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 either of these 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.

3.2.1 For MRO Connections

To establish an MRO connection for each region in an MRO-connected pair, you must define the **securityname** in one region equivalent to the CICS region ACID of the other regions. If the two values agree, the MRO connection request is accepted. Otherwise, the connection request is rejected and the MRO connection is not established.

Note: If the **securityname** is not specified, the connection request is allowed. In other words, bind-time security is not in effect.

The **securityname** is used for both bind-time and link-level security.

Figure 3.1 shows how to define bind-time security for MRO connections.

```
RDO definition                    RDM definition

DEFINE                            DFHTCT TYPE=SYSTEM
  CONNECTION(sysidnt)
  GROUP(groupname)
  ACCESSMETHOD(IRC|XM)              ACCMETH={IRC|(IRC,XM)}
  NETNAME(name)                     NETNAME=name
  SECURITYNAME(name)                XSNAME=name
   .                                       .
   .                                       .
```

Figure 3-1. Defining MRO Bind-time Security to CICS

An example of how bind-time security works for MRO connections is shown here in Figure 3.2. using RDO parameters.

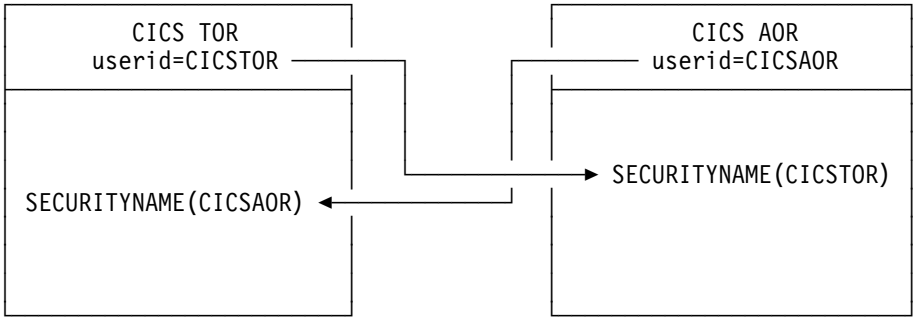


Figure 3-2. MRO Bind-time Security

In the example above, the TOR passes its CICS region ACID to the AOR. This CICS region ACID is compared to the **securityname** defined in the CONNECTION definition for the link with the TOR. In this example they match, therefore the bind request is successful.

Again looking at the example, the AOR passes its CICS region ACID to the TOR. This CICS region ACID is compared with the **securityname** defined in its connection definition for the link with the AOR. In this example they match, therefore the bind request is successful.

3.2.2 For ISC Connections

External bind-time security for ISC is established by specifying a bind password 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.

Figure 3.3 shows how to define external bind-time security for ISC connections. Note that you must specify a bind password.

```
RDO definition                      RDM definition

DEFINE                              DFHTCT TYPE=SYSTEM
  CONNECTION(sysidnt)
  GROUP(groupname)
  ACCESSMETHOD(VTAM)                  ,ACCMETH=VTAM
  NETNAME(name)                       ,NETNAME=name
  PROTOCOL(APPC)                      ,TRMTYPE=LUTYPE62
  SINGLESESS(N)                       ,FEATURE=PARALLEL
  SECURITYNAME(name)                  ,XSNAME=name
  BINDPASSWORD(security)              ,BINDPWD(password)
```

Figure 3-3. Defining ISC External Bind-time Security to CICS

An example of how external bind-time security works for ISC connections is shown here in Figure 3.4.

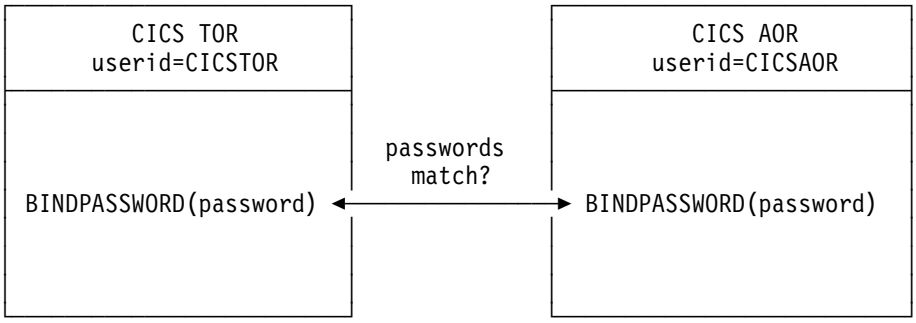


Figure 3-4. ISC External Bind-time Security

Specifying a bind password causes CICS 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).

When the bind request completes successfully, CICS performs link signon processing to check link security.

3.3 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 **securityname** defined in the connection or the CICS region ACID, if the **securityname** is omitted.

Since security calls are being made against the link, the CICS region ACID for the link must either have permission to these resources, or have the NORESCHK and NOLCFCHK attributes defined to the ACID.

Use the default settings for OPERSECURITY and OPERRSL session operands, which are OPERSECURITY(1) and OPERRSL(0). If you specify OPERSECURITY and/or OPERRSL, CICS will not perform a signon for the link, even if you have specified a security name. Link security then defaults to the destination region ACID.

No signon for the link takes place if the requesting system passes a **securityname** that matches the receiving CICS regions ACID. Therefore, if you are using bind-time security and you want to apply effective link security, the **securityname** on one side of an MRO link must **not** match the **securityname** 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 may 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.

Note: For ISC, IBM does not restrict the value of the security name. It is recommended that you code a distinct name other than the region ACID and protect it using password protection.

3.3.1 For MRO and ISC

Link security works by signing on to each end of a session (via receive terminals) using the **securityname** specified on the CONNECTION definition.

Figure 3.5 shows how to define link security for both MRO and ISC connections. Note that you must specify a **securityname.**

```
RDO definition                    RDM definition

DEFINE                            DFHTCT TYPE=SYSTEM
  CONNECTION                          .
   .                                  .
  SECURITYNAME(name)                XSNAME=name
   .                                  .
   .                                  .
```

Figure 3-5. Defining Link Security to CICS

An example of how link security works in MRO and ISC connections is shown here in Figure 3.6.

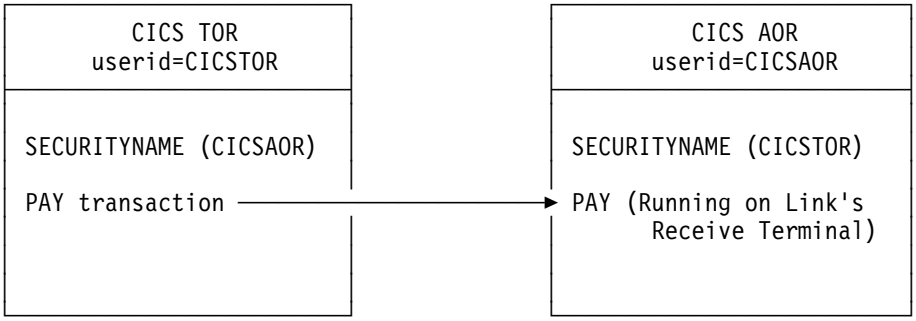


Figure 3-6. Link Security in MRO and ISC Connections

In this 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) will not be able to execute in the AOR region if the CICS region ACID CICSTOR:

- is not permitted to the PAY transaction, or
- is not defined with the NORESCHK or NOLCFCHK attributes.

3.4 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 Figure 3.7.

```
RDO definition                          RDM definition

DEFINE                                  DFHTCT TYPE=SYSTEM
  CONNECTION(sysidnt)                       ,SYSIDNT=name
  GROUP(groupname)
  .                                       .
  ATTACHSEC{Local|                        ,USERSEC={Local|
           Identify|                                Identify|
           Verify|                                  Verify|
```

Figure 3-7. Defining Attach-time Security to CICS

There are three levels of attach-time security: LOCAL, IDENTIFY, VERIFY.

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.

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

How CA-Top Secret relates to these levels is explained below; for complete information, refer to IBM's *Intercommunication Facilities Guide.*

3.4.1 Local Security Considerations

If you include a remote resource name in your CICS resource definitions, CA-Top Secret performs security checking locally against that remote resource, just as if the remote resource were a local one. The security check (on the remote side) is performed against the **securityname** for the connection or, if **securityname** was not specified, the CICS region ACID is used.

The following resource areas are secured under attach-time LOCAL security:

- All locally-defined resources
- Remote-defined transactions (transaction routing requests)
- Remote-defined files, transient data, and temporary storage queues (function shipping requests).

However, a LOCAL specification does not provide security coverage for the following resource areas:

- Remote-defined transactions that have alias resource names. The alias resource name is not protected.
- Remote data sets (DSNCHECK=YES)
- Any resource request originating from the remote side.

3.4.2 Remote Security Considerations

The attach-time parameters IDENTIFY and VERIFY 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:
  - **For CICS 2.3,** a period of 30 minutes has elapsed since the previous attach request from this user.
  - The link to the remote CICS region has been broken.
  - The CICS system has been recycled.
- Some CICS releases will 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. Refer to the *CICS Interregion Communication Guide* for your appropriate CICS release to determine if this is a situation you must allow for. This ACID should be coded with a non-expiring password. The following example shows that there is no need for permission just to add the facility the SYSID is associated with.

```
TSS CREATE('SYSID') NAME('CICS SYSID ACID')
    FAC(CICS) PAS(XXXX,0) DEPT(deptacid)
```

Chapter 4. Implementing Security

The previous chapters explained how to get your CICS system running in a secured environment. This chapter details the day-to-day operations you need to administer your CICS system in a secured environment, including:

- 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. Refer to the *User Guide* for details on PassTicket.

4.1 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.

The sign on procedures listed here include:

- Standard CICS Signon
- New Password Signon
- Automatic Terminal Signon (ATS)
- Interactive Interface Signon

4.1.1 Using CESN and CSSN

You can sign on to CA-Top Secret CICS via the IBM-supplied CESN and CSSN transactions.

- 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 **CSSN** transaction is used to sign on an up to eight-character alphanumeric userid. This userid should be the same as the CA-Top Secret user ACID.

Both CESN and CSSN signon procedures can be executed via screen prompts or by stringing the commands together.

4.1.1.1 Command Strings

Use the following syntax to sign on to CESN and CSSN transactions. Note that the password is displayed.

For CESN:

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

For CSSN:

```
CSSN PS=password,NAME=name,NEWPS=newpassword
```

Make the appropriate entries where:

USERID= For CESN—is the up to eight-character alphanumeric userid or the defined CA-Top Secret user ACID.

NAME= For CSSN—is the up to eight-character alphanumeric for the userid or the defined CA-Top Secret user ACID.

PS= For CESN and CSSN—is the password associated with the user ACID (maximum eight-characters).

NEWPS= For CESN and CSSN—is the "new" password replacing your lost, expired, or existing password.

4.1.1.2 Screen Prompts

At most sites, the signon screen is automatically displayed. If it is not, you can sign on to CICS using CESN or CSSN via screen prompts. Both CESN and CSSN signon screens are shown below; however, all examples use the CESN signon procedures. Refer to IBM's *CICS-Supplied Transactions* guide for details on CESN and CSSN signon procedures.

The signon screen for CESN looks like this:

```
CESN - CICS/VS SIGNON - ENTER USERID AND PASSWORD

     USERID: _

     PASSWORD:
     NEWPASSWORD:
```

The signon screen for CSSN looks like this:

```
CICS/VS SIGNON - ENTER PERSONAL DETAILS

NAME:_

PASSWORD:

NEWPASSWORD:
```

Make the appropriate entries in each field where:

USERID For CESN—is the eight-character alphanumeric userid or the defined CA-Top Secret user ACID.

NAME For CSSN—is the eight-character alphanumeric for the userid or the defined CA-Top Secret user ACID.

PASSWORD For CESN and CSSN—is the password associated with the user ACID (maximum eight-characters).

NEWPASSWORD For CESN and CSSN—is the new password associated with the user ACID (maximum eight-characters) which replaces a lost, expired, or existing password.

The standard signon procedure using the CESN screen prompts is:

1. Type your CA-Top Secret ACID (maximum eight-character alphanumeric) in the USERID: field.
2. Type your selected password (maximum eight-character alphanumeric). The characters in the password field will not display.
3. Then press ENTER.
4. 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
```

These messages are automatically cleared after a few seconds. See the *Messages and Codes Guide* for details about messages.

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

4.1.2 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 will be 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 eligible 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 CRE(K067T018) NAME('EMAIL SYSTEM GR 1')
    FAC(CICSPROD) DEPT(CIPCC)
    PASSWORD(NOPW,0)
    SOURCE(K067T018)
```

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

Note: ATS is not performed if: Validation is not required for a transaction being entered at a terminal; XTRAN=NO; or the transaction is in the Bypass List.

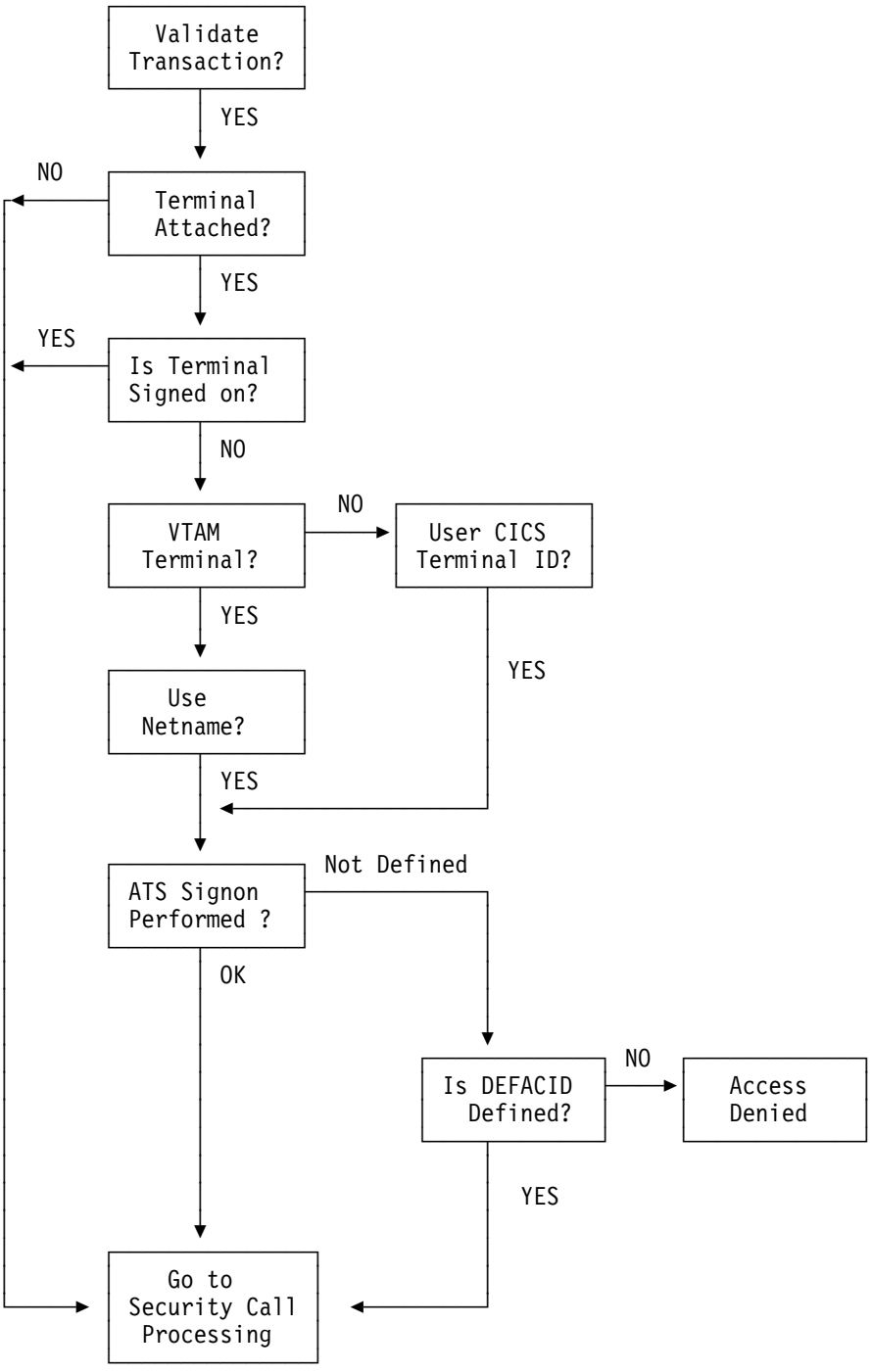


Figure 4-1. Automatic Terminal Processing (ATS)

4.1.3 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 ADD(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
DFHIC TYPE=PUT
DFHIC TYPE=INITIATE

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.

4.1.4 Signon-generated Return Codes

The following table lists the reponse codes and descriptions of the ESMRESP, EIBRESP, EIBRESP2 return codes that can be generated by an EXEC CICS signon.

Description	Response Code	Return Code
Undefined ACID	ESMRESP	X'04'
	EIBRESP	X'46'
	EIBRESP2	X'08'
Password Missing	ESMRESP	X'08'
	EIBRESP	X'46'
	EIBRESP2	X'01'
Password Incorrect	ESMRESP	X'08'
	EIBRESP	X'46'
	EIBRESP2	X'02'
Password Expired/New Password Missing	ESMRESP	X'0C'
	EIBRESP	X'46'
	EIBRESP2	X'03'
New Password Invalid	ESMRESP	X'10'
	EIBRESP	X'46'
	EIBRESP2	X'04'
Suspended ACID	ESMRESP	X'1C'
	EIBRESP	X'46'
	EIBRESP2	X'13'
Access Denied to Terminal	ESMRESP	X'30'
	EIBRESP	X'46'
	EIBRESP2	X'10'
Access Denied to Facility	ESMRESP	X'34'
	EIBRESP	X'46'
	EIBRESP2	X'11'

4.1.5 Interactive Interface Signon

Many sites require some or all users to signon to the Interactive Interface (IUI), an IBM productivity option that facilitates VSE/ESA system administration and provides a pathway into CICS. CA-Top Secret will recognize this signon for any user that has been set up for external security as described in 1.7, “Interactive Interface Signon Compatability.” The signon screen for the Interactive Interface follows:

```
IESADMSO1                        VSE/ESA ONLINE
    5690-VSE and Other Materials (C) Copyright IBM Corp. 1997 and other dates

    VV    VV    SSSSS    EEEEEEE          ++
    VV    VV   SSSSSSS   EEEEEEE         ++
    VV    VV  SS         EE             ++    EEEEEEE    SSSSS       AA
    VV    VV   SSSSSS    EEEEEE        ++     EEEEEEE   SSSSSSS     AAAA
    VV    VV    SSSSSS   EEEEEE       ++      EE       SS          AA  AA
     VV  VV          SS  EE          ++       EEEEEE    SSSSSS    AA    AA
      VVVV     SSSSSSS   EEEEEEE    ++        EEEEEE     SSSSSS   AA     AA
       VV       SSSSS    EEEEEEE   ++         EE             SS  AAAAAAAA
                                   ++          EEEEEEE   SSSSSSS   AA    AA
                                  ++           EEEEEEE    SSSSS    AA    AA

        Your terminal is L304 and its name in the network is D72L304
        Today is 08/04/1998   To sign on to DBDCCICS  --  enter your:

       USER-ID........ ________   The name by which the system knows you .
       PASSWORD.......            Your personal access code.

PF1=HELP       2=TUTORIAL               4=REMOTE APPLICATIONS     6=ESCAPE(U)
                           9=Escape(m) 10=NEW PASSWORD
```

Make the appropriate entries in each field where:

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

PASSWORD The password associated with the user ACID (maximum six characters).

The standard signon procedure using the IUI screen prompts is:

1. Type your CA-Top Secret ACID (maximum eight-character alphanumeric) in the USER-ID field.
2. Type your selected password (maximum six-character alphnumeric) in the PASSWORD field. The characters in the password field will not display.
3. Press Enter.
4. When signon is successful the following message is displayed:

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

These messages are automatically cleared after a few seconds. See the *Messages and Codes Guide* for details about messages.

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

4.1.5.1 IUI Signon Special Considerations

- The maximum six-character password rule is a physical limitiation of the IUI signon panel. Security administrators must be aware of this limitation so that assigned passwords are never longer than six characters If they are longer, the user will not be able to signon. Likewise, the users must be aware of the limitation so that they do not attempt to use a new password longer than six characters when their password expires.
- An IUI user that has been set up for an external security signon will not be able to use the PF10 option to change their password. If a new password is desired, it must be assigned by a security administrator, or the new password must be established via a CICS-specific signon (CSSN or CESN). Password expiration will occur during IUI signon processing, at which time a new password can be selected. Refer to 4.2.4, “Password Expiration,” for more information.
- The passwords described so far refer to passwords stored and maintained in the CA-Top Secret security file. The password stored in the VSE Control File for each IUI user is not used during an external security signon, nor is there any requirement that it match the CA-Top Secret password at any time. The VSE/ESA administrator need only maintain the control file password for ICCF signon compatibliltiy. The IESCNTL and DTSFILE passwords can be maintained without regard for the CA-Top Secret security file password.

4.2 Administering Passwords

This section covers:

- How to assign a new password
- How to change a password
- Random password generation
- Expired passwords
- Forgotten passwords

Because CA-Top Secret offers an extensive variety of password controls, you should develop password usage strategies particular to your site.

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

1. **Memorize** your password.
2. All written records of your password should be destroyed.
3. Do not post your ACID or password near the terminal, disks, cabinets, bulletin boards, or other areas accessible to unauthorized individuals.
4. Do not maintain your password in an unprotected data set where others could view it.
5. 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.
6. 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. Refer to the *Control Options Guide* for more information about password administration.

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

4.2.1 New Password Signon Procedure

You can assign a new password when you sign on using CESN or CSSN. Either screen prompts or a command string can be used. The following new password signon procedure uses the CSSN screen prompts shown below.

```
CESN - CICS/VS SIGNON - ENTER USERID AND PASSWORD

     USERID: _

     PASSWORD:
     NEWPASSWORD:
```

1. Type your CA-Top Secret ACID (maximum eight-character alphanumeric) in the USERID: field.
2. Type your selected password (maximum eight-character alphanumeric).
3. Type your new password (maximum eight-character alphanumeric). The characters in the password field will not display.
4. When all the information is complete, press the ENTER key.
5. When the password is changed, these messages are displayed:

```
TSS7030I Password Changed
TSS7000I acidname Last-Used mm/dd/yy hh:mm System=xxxx Facility=xxxxxxxx
TSS7001I Count=xxxxx Mode=xxxx Locktime=xxxxx Name=xxxxxxxxxxxxxxxxxxxx
```

4.2.2 Changing Passwords

You can change your password using CESN or CSSN signon transactions via screen prompts or a command string. The following procedure uses the CESN screen prompts.

```
CESN - CICS/VS SIGNON - ENTER USERID AND PASSWORD

     USERID: _

     PASSWORD:
     NEWPASSWORD:
```

1. Type your CA-Top Secret ACID (maximum eight-character alphanumeric) in the USERID: field.
2. Type your existing password (maximum eight-character alphanumeric). The password will not display.

3. Type your **new** password (maximum eight-character alphanumeric) in the NEWPASSWORD: field. The following message is displayed when the NPWR FACILITY suboption is in effect:

```
TSS7197I  Enter NEW Password Again for Reverification
```

4. Type your **new** password again. The following messages are displayed:

```
TSS7030I Password Changed
TSS7000I acidname Last-Used mm/dd/yy hh:mm System=xxxx Facility=xxxxxxxx
TSS7001I Count=xxxxx Mode=xxxx Locktime=xxxxx Name=xxxxxxxxxxxxxxxxxxxx
```

4.2.3 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). The procedure is:

1. Type your CA-Top Secret ACID in the USERID: field.
2. Type your selected 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 following messages are displayed:

```
TSS7030I  Password Changed
TSS7020I  Random Password About to be Displayed. Hit Enter to Continue.
```

4. Press the ENTER key again. The following messages are displayed:

```
TSS7021I Your New Password is xxxxxxxx
TSS7022I Memorize Password & Hit Enter Key - DO ***NOT*** RECORD
```

   Memorize the password generated for you (indicated above as xxxxxxxx.) Without this password you will not be able to sign on again.

5. 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
```

4.2.4 Password Expiration

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

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

mm/dd/yy displays the month, day, and year that your password will expire.

When your password expires, this message is displayed:

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

Following a two second pause, this message is displayed:

```
TSS7110I Please Enter Your ***NEW*** Password or 'RANDOM'
```

If you wish to select a new password, proceed to step 3 in 4.2.2, “Changing Passwords” and enter your new password. If you wish to select a random password, proceed to step 3 in 4.2.3, “Random Password Generation” and enter 'RANDOM'. If you wish to terminate the signon attempt, press Enter or CLEAR and the signon attempt will terminate.

4.3 Administering Transaction Security

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

4.3.1 OTRAN Security

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

Note: A transaction protected via OTRAN will not go through LCF checking.

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

```
TSS ADD(acid) OTRAN(transaction)
```

See the *User Guide* for a detailed discussion on OTRAN security.

4.3.2 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 either 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 ADD(acid) TRANS(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.

See the *User Guide* for a detailed discussion on LCF security.

4.4 Administering Resource Level Security

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

4.5 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.

Refer to the *User Guide* for a detailed description of the keywords used with SDT.

4.5.1 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. They are:

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).

Implementing RLP is a four step process:

Step 1 Gather Information

Step 2 Enter Definitions

Step 3 Permit Access to the Defined Records

Step 4 Enable Protection

Each step is described in detail in the following sections.

4.5.1.1 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.

Step 1 Determine which of your applications would benefit from RLP.

Step 2 Meet with the programmers to gather information about the application (like FCT name, field names, positions, data types, length of field, and selection criteria).

Step 3 Become familiar with the application.

Step 4 Plan the details needed to implement RLP for this application For example, you may decide on a selection criteria that limits the user to viewing only the records within their departmental scope.

Step 5 Determine who will be the administrator(s) for implementing RLP and give them MISC3(SDT) authority.

4.5.1.2 Enter Definitions

Once you have completed the preliminary steps as outlined in the previous section, you are ready to begin entering the definitions into the SDT. All definitions are entered using the TSS ADD(SDT) command.

Note: Depending on the number of records and fields you are protecting, these steps can be labor intensive.

Step 1 Define the RECORD definitions to the SDT. A sample RECORD definition is shown below.

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 will be used (referenced) in the selection process.
- What is the format of the data in the fields?
- What sizes are the fields?

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

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

Step 2 Define the SELECT expressions to the SDT that you will be using on the PERMIT command. A sample definition is shown below.

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?

```
TSS ADD(SDT) SELECT(dp1000)
             SELDATA('IF dept GE "1000" AND dept LE "1099")
```

Step 3 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 shown below.

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

Step 4 Check your work by listing the SDT records you just created. To list all records, use the command shown below.

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

Step 5 To correct any errors, first use the TSS REMOVE(SDT) command to remove any field you wish to modify, then use the TSS ADD(SDT) command to add the field you want to replace.

Step 6 When you are satisfied that everything is correct, refresh the SDT in-core tables using the command shown below.

```
TSS MODIFY(SDTTABLE)
```

4.5.1.3 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 may have for these FCTs.
2. Then, re-PERMIT the FCTs using the SELECT and/or MASKREC clauses. A sample PERMIT command is shown below.

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

4.5.1.4 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.) Use the command shown below to enable RLP in the CICS region.

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

4.5.1.5 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) ACC(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 may contain the masking values.
- Make certain that the data types specified in either the MASKDATA or RECDATA definitions match the data types of those contained on the actual file record.
- When RLP=YES, NOTAUTH conditions may have EIBRESP2=0000 instead of the expected EIBRESP2=101.

4.6 Administering Screen Level Protection (SLP)

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

- Define the screen (map) to CA-Top Secret Static Data Table (SDT). SDT record types of MAPREC and SELECT are required for SLP processing.

  Note: The SDT provides the framework on which field(s) within the terminal screen (map) are defined to SLP, and the selection criteria that identify if the map is allowed to be displayed.

- A SELECT clause must be specified on either the OTRAN or PPT permit of the transaction or program that is participating in SLP. The SELECT clause contains the name of the SDT SELECT record that is to be used as input into the SLP validation process.

  For example:

```
TSS PERMIT(userid) OTRAN(ABC) SELECT(MAPABC)
```

Refer to the *User Guide* to obtain more details on the SDT and SLP requirements.

4.7 Administering Terminal Security

The following sections explain how to:

- Restrict terminal use
- Secure sequential terminals
- Secure VSE console terminals
- Control when inactive or unattended terminals are locked
- Use preset terminal security

4.7.1 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 will be logged off after signon.

Refer to the TERMINAL keyword of the TSS ADD and PERMIT commands in the *Command Functions Guide* for details.

4.7.2 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 CSSN signon transaction can be specified. However, this is not recommended since the password would also have to be specified in the data set.

4.7.3 Securing VSE Console Terminals

Full security is enforced for VSE console terminals. Since an explicit signon is not appropriate for VSE 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 VSE 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 ADD(xxxx) TRANSACTION(fac,(CSMT,CEMT))
```

This prevents a VSE operator from entering sensitive transactions.

4.7.4 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:

1. When you signon a terminal, CA-Top Secret begins to monitor LOCKTIME thresholds.
2. When the LOCKTIME threshold is expired, at the next action key (ENTER, CLEAR, PF key, etc.) the terminal screen is cleared and you are prompted for your password.

   For CICS Release 2.3, you can enter your password, CSSF, or press the CLEAR key. The CLEAR key produces the same results as entering CSSF.
3. 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 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.

Note: Lock time processing is only performed in the CICS region where the terminal is defined. An entry in the LOCKTIME Bypass List overrides the lock time threshold specified for a terminal via the LOCKTIME control option.

4.7.5 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 below.

```
TSS ADD(CICSDEPT) SURROGAT(WAREHOUS.DFHINSTAL)
TSS PER(CICSADM) SURROGAT(WAREHOUS.DFHINSTAL) ACC(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.

4.8 Administering CICS Command Security

CA-Top Secret provides the SPI resource for added security checking. With the CA-Top Secret SPI resource you can secure the following:

- CEMT commands
- EXEC CICS INQUIRE and SET commands
- EXEC CICS ENABLE, DISABLE and EXTRACT commands
- EXEC CICS SPOOLOPEN command.

Note: SPI secondary resource protection under CICS Release 2.3 is limited since CICS does not acknowledge SPI resources until CICS version 3. The limitation consists of not being able to protect secondary resources within a CEMT command list. However, the SPI resource is still protected.

4.8.1 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.

Table 4-1 *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]
```

Table 4-2 *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 Table 4-3 *CEMT Secondary Resource Checks*.

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

Table 4-1. SPI Access Levels for CEMT

CEMT Action	**SPI Access Level**
ADD	SET
INQUIRE	INQUIRE
PERFORM	PERFORM
REMOVE	SET
SET	SET

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.

4.8.1.1 INQUIRE and SET Commands

The following table lists the CEMT command keywords and their associated SPI resource names. for specified resources, secondary resource checking applies. See the section Secondary Resource Checks later in this chapter for details.

Table 4-2 (Page 1 of 2). SPI Keywords for CEMT INQUIRE and SET Actions

Command Keyword	**SPI Keyword**
'Blanks' (default)	SPI(SYSTEM)
AUTINSTMODEL	SPI(AUTINSTM)
AUTOINSTALL	SPI(AUTOINST)
AUXTRACE	SPI(AUXTRACE)
CONNECTION	SPI(CONNECTI)
CONTROL	SPI(CONTROL)
DLIDATABASE	SPI(DLIDATAB)
DSNAME	SPI(DSNAME)
DUMP	SPI(DUMP)
DUMPDS	SPI(DUMPDS)
DUMPOPTIONS	SPI(DUMPOPTI)
DATASET	SPI(DATASET)
FILE	SPI(DATASET)
GTFTRACE	SPI(GTFTRACE)
INTTRACE	SPI(INTTRAC)

Table 4-2 (Page 2 of 2). SPI Keywords for CEMT INQUIRE and SET Actions

Command Keyword	**SPI Keyword**
IRBATCH	SPI(IRBATCH)
IRC	SPI(IRC)
JOURNAL	SPI(JOURNAL)
JOURNALNUM	SPI(JOURNAL)
LINE	SPI(LINE)
MODENAME	SPI(MODENAME)
MONITOR	SPI(MONITOR)
NETNAME	SPI(NETNAME)
PARTNER	SPI(PARTNER)
PITRACE	SPI(PITRACE)
PROFILE	SPI(PROFILE)
PROGRAM	SPI(PROGRAM)
QUEUE	SPI(QUEUE)
SYSTEM	SPI(SYSTEM)
STATISTICS	SPI(STATISTI)
SYSDUMPCODE	SPI(SYDUMPCO)
TASK	SPI(TASK)
TCLASS	SPI(TCLASS)
TDQUEUE	SPI(TDQUEUE)
TERMINAL	SPI(TERMINAL)
TRANSACTION	SPI(TRANSACT)
TRDUMPCODE	SPI(TRDUMPCO)
VOLUME	SPI(VOLUME)
VTAM	SPI(VTAM)

Examples for securing CEMT INQUIRE and SET commands appear next.

CEMT INQUIRE

Using the commands below, 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 ADD(deptacid) SPI(SYSTEM)

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

CEMT INQUIRE DUMP

Using the commands below, the user only has permission to execute CEMT INQUIRE DUMP commands.

```
TSS ADD(deptacid) SPI(DUMPDS)

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

CEMT INQUIRE AUTOINSTALL

Using the commands below, the user only has permission to execute CEMT INQUIRE AUTOINSTALL commands.

```
TSS ADD(deptacid) SPI(AUTOINST)

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

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

CEMT SET VTAM OPEN

Using the commands below, the user only has permission to execute CEMT SET VTAM OPEN commands.

```
TSS ADD(deptacid) SPI(VTAM)

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

4.8.1.2 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 Table 4-3 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 or not SPI access to INQUIRE or SET the CICS resource class has been granted.

Table 4-3. CEMT Secondary Resource Checks

CEMT Keyword	**Secondary Resource Type**
DSN	DATASET
FILE	FCT
JOURNAL	JCT
PROGRAM	PPT
QUEUE	DCT
TRANSACTIONS	OTRAN or LCF
VOLUMES	VOLUMES

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

```
TSS MODIFY FAC(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.

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

```
TSS MODIFY FAC(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 for securing CEMT secondary resources appear below.

CEMT INQUIRE TRAN(CS*)

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

```
TSS ADD(deptacid) SPI(TRANSACT)

TSS ADD(deptacid) OTRAN(CS)

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

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

4.8.1.3 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 table below lists the CEMT command keywords and their SPI equivalents for the CEMT PERFORM action.

Table 4-4. SPI Keywords for CEMT PERFORM

Command Keyword	**SPI Keyword**
RECONNECT	SPI(RECONNEC)
RESET	SPI(RESET)
SECURITY	SPI(SECURITY)
SHUTDOWN	SPI(SHUTDOWN)
SNAP	SPI(SNAP)

Examples for securing CEMT PERFORM commands appear next.

CEMT PERFORM SHUTDOWN

Using the commands below, the user only has permission to execute CEMT PERFORM SHUTDOWN commands.

```
TSS ADD(deptacid) SPI(SHUTDOWN)

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

4.8.1.4 ADD and REMOVE Commands

You can secure CEMT ADD and REMOVE commands for VOLUMEs only. The following access levels are valid:

Table 4-5. CEMT ADD and REMOVE Commands

Command Keyword	**SPI Keyword**
VOLUME	SPI(VOLUME)

Examples for securing CEMT ADD and REMOVE commands appear next.

Using the commands below, the user only has permission to execute CEMT ADD and REMOVE commands for VOLUMEs only.

```
TSS ADD(deptacid) SPI(VOLUME)

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

TSS REMOVE(acidname) SPI(VOLUME)

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

4.8.2 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* is equivalent to the CA-Top Secret SPI resource.
- *argument* is the data element being examined or modified.

For example:

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

Follow these steps to secure EXEC CICS commands:

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

For example:

```
TSS ADD(divacid) SPI(FILE)

TSS PER(acid) SPI(FILE) ACC(SET)
```

Note: SPI protection is active by default. To turn off SPI checking, you must specify both the FACMATRX=YES and the XCMD=NO FACILITY suboptions.

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.

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

Table 4-6. SPI Access Levels for EXEC CICS

EXEC CICS Command	**SPI Access Level**
SET	SET

EXEC CICS INQUIRE and SET commands have related *options.* The next section describes how CA-Top Secret secures each option and its associated command.

4.8.2.1 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.

Table 4-7. SPI Keywords for EXEC CICS INQUIRE and SET Options	
Command Option	**SPI Keyword**
CONNECTION	SPI(CONNECTI)
DATASET	SPI(DATASET)
FILE	SPI(FILE)
MODENAME	SPI(MODENAME)
PROGRAM	SPI(PROGRAM)
SYSTEM	SPI(SYSTEM)
TERMINAL	SPI(TERMINAL)
TRANSACTION	SPI(TRANSACT)

4.8.2.2 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.

Below is a table containing EXEC CICS keywords, the resource types called in the secondary CA-Top Secret security check, and the associated access levels.

Table 4-8. EXEC CICS Secondary Resource Checks		
EXEC CICS Keyword	**Secondary Resource Type**	**Access Level**
DATASET	FCT	INQUIRE, SET
FILE	FCT	INQUIRE, SET
PROGRAM	PPT	INQUIRE, SET
TRANSACTIONS	OTRAN	INQUIRE, SET

Examples for securing EXEC CICS INQUIRE and SET commands appear next.

EXEC CICS INQUIRE PROGRAM(TSSCAI)

Using the commands below, the user only has permission to execute EXEC CICS INQUIRE PROGRAM(TSSCAI) commands.

```
TSS ADD(deptacid) SPI(PROGRAM)

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

TSS ADD(deptacid) PPT(TSSCAI)

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

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

EXEC CICS SET TRANSACTION(TSS)

Using the commands below, the user only has permission to execute EXEC CICS SET TRANSACTION(TSS) commands.

```
TSS ADD(deptacid) SPI(TRANSACT)

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

TSS ADD(deptacid) OTRAN(TSS)

TSS PERMIT(acidname) OTRAN(TSS) ACC(INQUIRE)
```

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

4.8.2.3 ENABLE, DISABLE, and EXTRACT Commands

You can secure the ENABLE, DISABLE, and EXTRACT 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, and EXTRACT 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:

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

The equivalent CA-Top Secret commands for the IBM EXEC CICS command shown above are:

```
TSS ADD(deptacid) SPI(ENABLE)

TSS PER(acid) SPI(ENABLE) ACC(SET)
```

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

Table 4-9. SPI Access Levels for EXEC CICS

Command Function	**SPI Access Level**
ENABLE	SET
DISABLE	SET
EXTRACT	INQUIRE

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

4.8.2.4 Securing Functions

The table below lists the EXEC CICS command functions and their SPI equivalents for the EXEC CICS ENABLE, DISABLE, and EXTRACT commands.

Table 4-10. SPI Keywords for EXEC CICS ENABLE, DISABLE, and EXTRACT

Command Function	**SPI Keyword**
ENABLE	SPI(ENABLE)
DISABLE	SPI(DISABLE)
EXTRACT	SPI(EXTRACT)

Examples for securing EXEC CICS ENABLE, DISABLE, and EXTRACT commands appear next.

EXEC CICS ENABLE

Using the commands below, the user only has permission to execute the EXEC CICS ENABLE commands.

```
TSS ADD(deptacid) SPI(ENABLE)

TSS PERMIT(acidname) SPI(ENABLE) ACC(SET)
```

EXEC CICS DISABLE

Using the commands below, the user only has permission to execute the EXEC CICS DISABLE commands.

```
TSS ADD(deptacid) SPI(DISABLE)

TSS PERMIT(acidname) SPI(DISABLE) ACC(SET)
```

EXEC CICS EXTRACT

Using the commands below, the user only has permission to execute the EXEC CICS EXTRACT commands.

```
TSS ADD(deptacid) SPI(EXTRACT)

TSS PERMIT(acidname) SPI(EXTRACT) ACC(INQUIRE)
```

4.8.2.5 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)
```

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

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

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

Table 4-11. SPI Keywords for EXEC CICS SPOOLOPEN

Command Function	**SPI Keyword**
SPOOLOPEN	SPI(PWRSPOOL)

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

Table 4-12. SPI Access Levels for EXEC CICS SPOOLOPEN

Command Options	**SPI Access Level**
INPUT	READ
OUTPUT	WRITE

Examples for securing EXEC CICS SPOOLOPEN commands appear next.

EXEC CICS SPOOLOPEN INPUT

Using the commands below, the user only has permission to execute the EXEC CICS SPOOLOPEN INPUT commands.

```
TSS ADD(deptacid) SPI(PWRSPOOL)

TSS PERMIT(acidname) SPI(PWRSPOOL) ACC(READ)
```

EXEC CICS SPOOLOPEN OUTPUT

Using the commands below, the user only has permission to execute the EXEC CICS SPOOLOPEN INPUT commands.

```
TSS ADD(deptacid) SPI(PWRSPOOL)

TSS PERMIT(acidname) SPI(PWRSPOOL) ACC(WRITE)
```

4.8.2.6 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.

EXEC CICS SPOOLOPEN INPUT USERID(ext writer name)

Using the commands below, the user only has permission to execute the EXEC CICS SPOOLOPEN INPUT USERID commands.

```
TSS ADD(deptacid) ABSTRACT(ext writer name)

TSS PERMIT(acidname) ABSTRACT(ext writer name)
```

EXEC CICS SPOOLOPEN OUTPUT USERID(userid)

Using the commands below, the user only has permission to execute the EXEC CICS SPOOLOPEN OUTPUT USERID commands.

```
TSS ADD(deptacid) ABSTRACT(userid)

TSS PERMIT(acidname) ABSTRACT(userid)
```

4.9 Securing DL/I PSBs and DBDs

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.

4.10 Securing ICCF

This information only applies to running batch jobs in ICCF pseudo partitions.

CA-Top Secret protects VSE libraries, files and program loads for jobs executing in ICCF pseudo partitions in the same way it supports any other VSE batch process.

Any jobs executing in an ICCF pseudo partition will execute under the facility name of the CICS region instead of the default BATCH facility.

To implement and support ICCF security checks, the RES facility option must be added for the facility in which ICCF resides:

```
TSS MODIFY(CICSPROD=RES)
```

Alternatively, it can be added permanently in the TSSPARM file.

Chapter 5. Programmable Interfaces

This chapter discusses the Application Interface and the CA-Top Secret CICS exits.

5.1 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.

5.1.1 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 VSE library available to the CICS partition via a LIBDEF, and must be defined to the DFHPPT definition macro. The Application Interface can be invoked by either Command-Level 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 the section "Coding Samples", later in this chapter. Samples of PL/I, Assembler, and COBOL coding are located on the CA-Top Secret distribution tape, under File 11, TSSOPMAT.

5.1.2 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 TSSCAIO must have CSD entries like the one shown below (the example shows that the program is written in the Assembler language).

```
DEFINE PROGRAM(TSSCAI) GROUP(TOPSGRP)
       LANGUAGE(ASSEMBLER) RELOAD(NO)
       RESIDENT(NO)
       STATUS(ENABLED)
DEFINE PROGRAM(TSSCAIO) GROUP(TOPSGRP)
       LANGUAGE(ASSEMBLER) RELOAD(NO)
       RESIDENT(NO)
       STATUS(ENABLED)
```

- You must pass the Application Interface a parameter list. The parameter list is passed by either a temporary storage queue or by Command-Level COMMAREA.
- For **Release 3.0,** the length of the COMMAREA or the temporary storage queues that contain the Application Interface parameter 1ist must be 370 or 1138 bytes (The 1138 value refers to the FACLIST RESLIST and FLDXTR calls; all other calls use 370 bytes.)

5.1.3 Installation-defined Resources

Installation-defined resource classes (such as FIELD, UR1, UR2, and ABSTRACT) that are also predefined 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, refer to the TSS ADD(RDT) command function in the *Command Functions Guide.*

5.1.4 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. Refer to Chapter 4, "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.

5.1.5 Coding Samples

Use the coding examples provided on the following pages customization samples.

5.1.5.1 Test TSSCAI Using Temporary Storage Record

```
      TITLE 'TESTCAI 1' --- TSS CICS APPLIATION INTERFACE
************************************************************
*  NAME     - TESTCAI1                                     *
*  FUNCTION - COMMAND LEVEL ASSEMBLER CODE ......          *
*             TEST TSSCAI USING TEMPORARY STORAGE RECORD.  *
*  CALLS    - THE CICS APPLICATION INTERFACE PROGRAM       *
************************************************************
           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.
*
```

Figure 5-1. Customization Sample

```
         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'TCPLV4L4'
         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)
               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)
               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
```

5.1.5.2 Test TSSCAI Using CICS COMMAREA

```
         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'TCPLV4L4'
         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
```

Figure 5-2. Customization Sample

5.2 CA-Top Secret CICS Exits

CA-Top Secret provides two user exits: TSSPGM01 and TSSPGM02.

- TSSPGM01 is a message exit that lets you suppress or change the text of messages.
- TSSPGM02 is a message exit that lets you suppress or change the text of the locktime prompt.

The following sections explain how to modify these exits.

5.2.1 The TSSPGM01 Exit

The TSSPGM01 exit is enabled by defining the PPT (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
```

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 will not be 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).

5.2.2 The TSSPGM02 Exit

The TSSPGM02 exit is enabled by defining the PPT (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:

```
WPARMLIST DS    0H           PROGRAM PARAMETER LIST
WMGAREA   DS    0XL79        MESSAGE AREA
WMSGLN    DS    H            MESSAGE LENGTH
          DS    H
WMESSAGE  DS    XL75         MESSAGE AREA
WPPWAREA  DS    0XL8         PASSWORD AREA
WPSWD     DS    XL8          PASSWORD
WPLISTLN  EQU   *-WPARMLST   PROGRAM P-LIST LENGTH
$TABEND   EQU   X'FF'        ABEND TASK
WMSGALN   EQU   *-WPARMLST   PARAMETER LIST LENGTH
```

WMGAREA Contains the password prompt message.

WPPWAREA Is the field that the TSSPGM02 Exit program places the user's password in for reverification.

5.2.3 Sample Program Definitions

Program definitions for the TSSPGM01 and TSSPGM02 message exits should look like the example shown below.

```
DEFINE  PROGNAM(TSSPGM01) GROUP(TOPGRP)
        LANGUAGE(ASSEMBLER)
        RESIDENT(NO)
        STATUS(ENABLED)
```

Note: EXECKEY must be CICS, otherwise errors will result.

Chapter 6. CA-Top Secret Supplied Transactions

This chapter discusses the CA-Top Secret-supplied transactions and the CICS-supplied transactions.

6.1 LOCKTIME Logoff Feature Support (TSLO, TSSS)

The TSLO and TSSS transactions and their associated programs are required if you are specifying the LTLOGOFF FACILITY suboption.

6.1.1 TSLO Transaction

TSLO is a CA-Top Secret CICS transaction used to support the new LOCKTIME logoff feature set via the LTLOGOFF FACILITY suboption. LTLOGOFF controls whether or not CA-Top Secret causes user logoff after the second LOCKTIME interval expires. This transaction is used to perform CSSN LOGOFF and can be issued from EXEC CICS START.

6.1.2 TSSS Transaction

TSSS is a CA-Top Secret CICS transaction used with the TSLO transaction (described previously) to support LOCKTIME processing.

6.2 The User Executed Transaction Utility (TSSC)

TSSC 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:

TSSC=INSTALL	Analyzes the installation specifications of a CICS region.
TSSC=WHOSON	Indicates who is signed on to a particular region.
TSSC=TRANS=(trans)	Gives information about a specified CICS transaction, where *trans* is the four-character CICS transaction ID specified in the PCT.

TSSC=MAXT=INQ	Inquires about the maximum number and actions available for concurrent signon/signoff requests that are set.
TSSC=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.
TSSC=TRACE=(INQ|ON|OFF)	Controls or displays the status of the CA-Top Secret CICS diagnostic tracing facility.

6.2.1 Executing TSSC

This section contains a detailed description of the information returned by each of the transactions.

TSSC=INSTALL details the following information about a region:

- Whether EXTSEC=YES or EXTSEC=NO is coded on SIT.
- Where the CA-Top Secret modules are located.
- Whether the Application Interface is installed.
- Whether the TSS command is installed.
- The name of the control region.
- The name of the MASTFAC.

TSSC=WHOSON indicates who is signed on to a particular region.

TSSC=TRANS=(trans) describes the following information about a specific CICS transaction:

- The address of the PCT entry.
- Whether the transaction is local or remote.
- The priority at which it is defined.
- The CICS security key associated with the transaction.
- 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.

TSSC=MAXT=INQ has the ability to *inquire* about the maximum number and actions available for concurrent signon/signoff requests that are set.

Note: The SET option is not used with CA-Top Secret Release 4.4. Refer to the MAXSIGN FACILITY suboption in the *Control Options Guide* for details.

TSSC=NEWC=(program) refreshes the running copy of a CA-Top Secret CICS module, allowing emergency maintenance to be applied to a single CICS region without recycling that specific region.

Note: The following TSS CICS modules are **not refreshable:** TSSCCMGR, TSSCLOCK, TSSCPI*nn*, TSSCPM*nn*, TSSCPTSS, TSSCRINT, TSSCRT*nn*, TSSCSCAN, TSSCSMGR, TSSTCINT, and TSSTRACK.

Additionally, the following CA-Top Secret CICS modules are refreshed through the CICS command CEMT SET PROGRAM(*name*) NEWCOPY: TSSCCHEK, TSSCICS, TSSCICSO, TSSCINST, TSSCMAXT, TSSCNEWC, TSSCTERM, TSSCTRAC, TSSCTRAN, TSSCMACS, and TSSCWHOS.

TSSC=TRACE=(INQ|ON|OFF) Controls or displays the status of the CA-Top Secret CICS diagnostic tracing facility.

INQ Displays the current status (ON|OFF) of the CA-Top Secret CICS diagnostic tracing facility.

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.

OFF Turns off the CA-Top Secret CICS diagnostic trace.

Chapter 7. CICS Installation Checklist

Use this checklist when you are installing and implementing CA-Top Secret security using CICS Release 2.3.

1. CAIENF Considerations

   The CA-Top Secret CICS interface requires the CA-CIS CAIENF (Event Notification Facility) 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:

   a. Check the LIBDEF to see if it contains the CA-Top Secret library. If it does not, add the library to the LIBDEF in the CICS JCL to properly load TSSTCINT. If TSSTCINT is not loaded, CA-Top Secret will not install into the CICS regions and you will not receive a Phase I message.

   b. Check the LIBDEF to see if it contains the CAIENF library. If it does not, add the library to the LIBDEF in the CICS JCL to properly load TSSTCINT. If TSSTCINT is not loaded, CAIENF will not install into the CICS regions and you will not receive a Phase I message.

   c. Ensure that EXTSEC is specified in either the CICS SIT or the Facility Matrix Table if the FACMATRX option has been set to YES.

   d. Ensure that one of the XPARMS (XFCT, XJCT, XPCT, etc) is set to YES.

   Refer to the *CA-CIS Installation Guide* for detailed information.

2. System Initialization Table (SIT)

   Although not required, EXTSEC=YES should be specified. For CA-Top Secret resource protection, allow the following to default or code YES for: XTRAN, XPCT, XFCT, XPPT, XTST, XPSB, XJCT, and XDCT.

3. Program Control Table (PCT)

   Add the EXTSEC=YES parameter to the DFHPCT TYPE=INITIAL macro as:

   ```
   DFHPCT TYPE=INITIAL,EXTSEC=YES
   ```

   CSSN/CESN and CSSF/CESF must be defined in the PCT with the SPURGE=NO attribute.

4. Sign-on Table (SNT)

   Add the EXTSEC=YES parameter to the DFHSNT TYPE=INITIAL macro:

   ```
   DFHSNT TYPE=INITIAL,EXTSEC=YES
   ```

   Place the entry default at the end of the signon table:

   ```
   DFHSNT TYPE=(ENTRY,DEFAULT),PASSWRD=password
   ```

   Followed by:

```
DFHSNT TYPE=FINAL
```

5. CA-Top Secret Supplied Transactions and Programs

   The following CA-Top Secret transactions are defined via updates to your CICS PCT and PPT definitions:

   - TSS command
   - TSSC User Executed Transaction Utility
   - CA-Top Secret Application Interface
   - TSSTRACK Utility

   Note: You must make updates to the PCT and PPT definition statements to use these CA-Top Secret-supplied transactions.

   All CA-Top Secret programs listed in the PPT statements must reside in the CICS LIBDEF search library.

   Refer to Chapter 8, "PPT and PCT Sample Entries" for the updated definition statements.

6. TSSTRACK Utility

   Allocate the Audit Tracking file to the CICS region.

7. ISC/MRO Considerations

   Review Chapter 3, "Security for a Multi-System Environment" prior to implementing security under an MRO and/or ISC environment.

8. Starting Your CICS Region

   a. Verify that your CICS region ACIDs have either the NORESCHK or NOLCFCHK attributes or have been PERMITted to the appropriate transactions.

   b. At this point CA-Top Secret and CAIENF are installed and active.

   c. The following messages are displayed in succession. Refer to the *Messages and Codes Guide* for complete descriptions of reasons and actions associated with each message.

      Note: Review the Install Check messages that follow message **TSS6004I Initialization Phase 3 Completed.** The security parameter values displayed in the Install Check messages are valid for the life of the session.

      If you want to dynamically change these security parameters use the MODIFY command on the facility. For permanent changes to these security parameters, use the MODIFY command in the TSS Parameter File.

      Install Check Messages

```
TSS6000I - TSS/CICS Initialization Phase 1 Started
TSS6001I - TSS/CICS Initialization Phase 1 Completed
TSS6002I - TSS/CICS Initialization Phase 2 Started
TSS6003I - TSS/CICS Initialization Phase 2 Completed
TSS6004I - TSS/CICS Initialization Phase 3 Started
TSS6030I - TSS/CICS Install Check  =====> FACMATRX=YES
TSS6031I - TSS/CICS Install Check  =====> EXTSEC=YES
TSS6032I - TSS/CICS Install Check  =====> XCMD=YES
TSS6033I - TSS/CICS Install Check  =====> XDCT=YES
TSS6034I - TSS/CICS Install Check  =====> XFCT=YES
TSS6035I - TSS/CICS Install Check  =====> XJCT=YES
TSS6036I - TSS/CICS Install Check  =====> XPCT=YES
TSS6037I - TSS/CICS Install Check  =====> XPPT=YES
TSS6038I - TSS/CICS Install Check  =====> XPSB=YES
TSS6039I - TSS/CICS Install Check  =====> XDBD=YES
TSS6040I - TSS/CICS Install Check  =====> XTRAN=YES
TSS6041I - TSS/CICS Install Check  =====> XTST=YES
TSS6042I - TSS/CICS Install Check  =====> DSNCHECK=NO
TSS6043I - TSS/CICS Install Check  =====> PCTEXTSEC=OVERRIDE
```

```
TSS6045I - TSS/CICS Install Check    ===> LTLOGOFF=NO
TSS6069I - TSS/CICS C S T M A   Processing
TSS6047I - TSS/CICS G U S H C I Processing
TSS6080I - TSS/CICS G D U H C I Processing
TSS6075I - TSS/CICS G U S C B C Processing
TSS6081I - TSS/CICS G D U S B C Processing
TSS6076I - TSS/CICS G U S C B B Processing
TSS6082I - TSS/CICS G D U S B B Processing
TSS6077I - TSS/CICS G T T P T   Processing
TSS6078I - TSS/CICS G T T C B C Processing
TSS6079I - TSS/CICS G T T C B B Processing
TSS6049I - TSS/CICS L C N-R D M Processing
TSS6050I - TSS/CICS L C R D M   Processing
TSS6051I - TSS/CICS L C R R T M Processing
TSS6053I - TSS/CICS L C T M     Processing
TSS6054I - TSS/CICS L C T P A M Processing
TSS6055I - TSS/CICS L C S M     Processing
TSS6016I - TSS/CICS PTSCECB2 Posted
TSS6017I - TSS/CICS PTSCECB3 Posted
TSS6005I - TSS/CICS Initialization Phase 3 Complete
TSS6007I - TSS/CICS Security Activated
```

d. Once the **TSS6007I Security Activated** message is displayed, the CA-Top Secret CICS interface is installed and active in the region.

9. 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.

Chapter 8. PPT and PCT Sample Entries

All PPT and PCT sample entries are provided in the product install library as members of type 'A'. The PPT table entries are contained in member TSSPPT23.A, and the PCT table entries are contained in member TSSPCT23.A. Additionally, member TSSCSD23.Z provides the required entries in a format suitable for updating the CSD dataset when using CICS RDO.

8.1 CA-Top Secret CICS Sample PPT Entries

```
DFHPPT TYPE=INITIAL,
     SUFFIX=TS

DFHPPT TYPE=ENTRY,
     PROGRAM=TSSCLOCK,      TSLO – LOGOFF FOR LTLOGOFF
     PGMLANG=ASSEMBLER

DFHPPT TYPE=ENTRY,
     PROGRAM=TSSCICS,       TSS – TSS COMMAND PROCESSOR
     PGMLANG=ASSEMBLER

DFHPPT TYPE=ENTRY,
     PROGRAM=TSSCCHEK,      TSSC – FUNCTION ROUTER
     PGMLANG=ASSEMBLER

DFHPPT TYPE=ENTRY,
     PROGRAM=TSSCINST,      TSSC – INSTALLATION OPTIONS DISPLAY
     PGMLANG=ASSEMBLER

DFHPPT TYPE=ENTRY,
     PROGRAM=TSSCMAXT,      TSSC – MAXIMUM USER DISPLAY
     PGMLANG=ASSEMBLER

DFHPPT TYPE=ENTRY,
     PROGRAM=TSSCNEWC,      TSSC – PROGRAM REFRESH PROGRAM
     PGMLANG=ASSEMBLER

DFHPPT TYPE=ENTRY,
     PROGRAM=TSSCTRAC,      TSSC – DIAGNOSTIC TRACE PROGRAM
     PGMLANG=ASSEMBLER
```

Figure 8-1. PPT Sample Entries

```
DFHPPT TYPE=ENTRY,
     PROGRAM=TSSCMACS,      TSSC – MACRO LEVEL SERVICES PROGRAM
     PGMLANG=ASSEMBLER

DFHPPT TYPE=ENTRY,
     PROGRAM=TSSCTRAN,      TSSC – TRANSACTION DISPLAY
     PGMLANG=ASSEMBLER

DFHPPT TYPE=ENTRY,
     PROGRAM=TSSCTERM,      TSSC – TERMINAL DISPLAY
     PGMLANG=ASSEMBLER

DFHPPT TYPE=ENTRY,
     PROGRAM=TSSCWHOS,      TSSC – SIGNED ON USER DISPLAY
     PGMLANG=ASSEMBLER

DFHPPT TYPE=ENTRY,
     PROGRAM=TSSCWRIT,      TSSC – MESSAGE WRITE PROGRAM
     PGMLANG=ASSEMBLER

DFHPPT TYPE=ENTRY,
     PROGRAM=TSSCSCAN,      TSSS – SCAN PROGRAM
     PGMLANG=ASSEMBLER

DFHPPT TYPE=ENTRY,
     PROGRAM=TSSCICSO,      TSS – TSS COMMAND PROCESSOR
     PGMLANG=ASSEMBLER

DFHPPT TYPE=ENTRY,
     PROGRAM=TSSCAI,        XXXX – APPLICATION INTERFACE
     PGMLANG=ASSEMBLER

DFHPPT TYPE=ENTRY,
     PROGRAM=TSSCAIO,       XXXX – APPLICATION INTERFACE
     PGMLANG=ASSEMBLER

DFHPPT TYPE=FINAL
```

8.2 CA-Top Secret CICS Sample PCT Entries

```
 PCT Sample Entries
DFHPCT TYPE=ENTRY,
     TRANSID=TSLO,          TSLO – TERMINAL LOGOFF
     EXTSEC=NO,
     PROGRAM=TSSCLOCK

DFHPCT TYPE=ENTRY,
     TRANSID=TSS,           TSS – TSS COMMAND PROCESSING
     EXTSEC=NO,
     PROGRAM=TSSCICS

DFHPCT TYPE=ENTRY,
     TRANSID=TSSC,          TSSC – TSS ENVIRONMENT UTILITY
     EXTSEC=NO,
     PROGRAM=TSSCCHEK

DFHPCT TYPE=ENTRY,
     TRANSID=TSSS,          TSSS – TSS SCAN TRANSACTION
     EXTSEC=NO,
     PROGRAM=TSSCSCAN
```

Chapter 9. Defining CICS Transaction Server 1.1 and Above to CA-Top Secret

This chapter explains how to install CA-Top Secret in your CICS Transaction Server (TS) Release 1.1 and above environment. All the tasks necessary to get your CICS system up and running in a secured environment are described. Later chapters give additional information for modifying your security environment.

9.1 Pre-Installation Considerations

The CA-Top Secret CICS interface requires the CA-CIS 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. Refer to the *CA-CIS Installation Guide* for detailed information.

9.1.1 Migration Considerations

- The DFHSIT parameter XUSER, new for CICS TS 1.1, now controls non-terminal (background) security.
- When running under CA-Top Secret Release 3.0, the XUSER parameter can only be altered from the DFHSIT (FACMATRX=NO). The default if you are using the facility overrides (FACMATRX=YES) is XUSER=YES.
- The following FACILITY suboptions that control security features can no longer be dynamically changed: DSNCHECK, EXTSEC, FACMATRX, LTLOGOFF, MAXSIGN, MAXUSER, SIGN, XAPPC, XCMD, XDCT, XFCT, XJCT, XPCT, XPPT, XPSB, XTRAN, XTST, and XUSER. After changing a FACILITY sub-option, you must re-cycle your CICS region for the changes to take effect.
- You no longer have to specify the sysid of the region in the Bypass List to deactivate security. Only SEC=NO is required.
- Installation check messages are no longer displayed at start up. Use TSEU=INSTALL to see the security parameter settings.
- If SEC=YES and ENF CICS is inactive when you try to start up, the CICS region will not initialize and CICS will abend with U1800.

- The EXEC CICS ENABLE, DISABLE, and EXTRACT commands are now protected by the SPI resource of EXITPROG.
- The PCTCMDSEC FACILITY suboption is now part of the DFHSIT parameter overrides. Based on the Facilities Matrix setting, this parameter either honors or overrides the DFHSIT parameter CMDSEC=.
- When running under CA-Top Secret Release 3.0, the PCTRESSEC parameter can only be altered from the DFHSIT (FACMATRX=NO). The default if you are using the facility overrides (FACMATRX=YES) is PCTRESSEC=OVERRIDE.
- The MAXUSER FACILITY suboption has been improved.
  - 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 may be active in the CICS region.
  - In addition to calculating the number of users for the user pool allocation, MAXUSER is used for a new feature called *resource caching.* This feature uses the MAXUSER setting to build the cache box pool.
- MRO securityname is no longer used for bind and link.
- PCT and PPT entries have been replaced by CSD entries.
- CICS TS 1.1 implements transaction security for background (non-terminal) transactions. You may 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 has been extended so that it is now invoked during any resource validation. Previously, ATS was only invoked at transaction startup time.
- The real Port-of-Entry (POE) is now 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.

- Certain transaction IDs and program names have been changed:

CICS Release 2.3	CICS TS 1.1
TSSS	TSLM
TSLO	TSLA
TSSC	TSEU
TSSCSCAN	CAKSSCAN
TSSCLOCK	CAKSLOCK

- TSEU=WHOSON also displays background (non-terminal) users, as well as terminals signed on through ATS.
- In addition to programs TSSCAI, TSSCICS, and TSSCTRK, you must also define the following items based on which release of CICS you are running.

CICS Release 2.3	CICS TS 1.1
TSSCAIO	TSSCAIN
TSSCICSO	TSSCICSN

9.2 Installing CA-Top Secret in CICS

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. For more information, refer to 9.3.6, “Defining the CICS Region Control ACID” on page 9-12.
- Define CICS as a batch job in the CA-Top Secret security environment. For more information, refer to 9.5.2, “Defining CICS” on page 9-16.

To install CA-Top Secret in your CICS system, you need to:

- Make sure that CA-Top Secret Release 3.0 and CAIENF are installed.

  Note: If the CA-Top Secret library is not in the LIBDEF, you must add it to the LIBDEF of the CICS startup JCL. If CAKSCINT is not loaded, CA-Top Secret will not install into the CICS regions and you will not receive a Phase 0 message.
- Any program defined in the CSD job displayed in Chapter 16, must be in the CICS LIBDEF search.
- Either set CICS security parameters in the CICS tables or define CA-Top Secret FACILITY suboptions for controlling CICS security processing in the Facility Matrix table.
- Activate your CICS region. A series of CA-Top Secret messages are displayed indicating the phase of initiation for the region; to view the region's security parameters, issue TSEU=INSTALL. A list of these messages appear in Chapter 15.

9.3 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 may 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. For these reasons, CA-Top Secret allows each CICS region to be associated with a unique facility, several CICS regions to be associated with a common facility, or any combination thereof.

To tell CA-Top Secret about your CICS region, you must associate a 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.**

CA-Top Secret defines a facility as a VSE subsystem that a user must have access to in order to enter the system. For example, Batch is defined to CA-Top Secret as a facility that must be accessed by users and jobs. All facilities that interface with CA-Top Secret must be defined in the Facilities Matrix.

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 facility-defined CICS regions 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 a list and definitions of the CA-Top Secret FACILITY suboptions, refer to the *Control Options Guide*. For an explanation of how to configure these suboptions for your CICS system, refer to Chapter 12, "Implementing Security".

9.3.1 Providing CICS Default Facilities

CA-Top Secret provides two CICS default facilities—CICSPROD and CICSTEST—that are already defined in the Facilities Matrix. This means that the security attributes that control CA-Top Secret processing for CICSPROD and CICSTEST are predefined. These attributes, listed in the following figures, are actually suboptions of the FACILITY control option. You can use the CICSPROD and CICSTEST default facilities as they are, or you can customize them for your site.

CICSPROD Facility: The defaults for the CICSPROD facility are:

```
INITPGM=DFH          ID=C  TYPE=04
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,NODORMPW,NONPWR,NOIMSXTND
MODE=FAIL  LOGGING=INIT,MSG,SEC9,SMF
UIDACID=8  LOCKTIME=000  DEFACID=*NONE*  KEY=8
FACMATRX=NO         EXTSEC=YES
XJCT=YES  XFCT=YES  XCMD=YES XDCT=YES  XTRAN=YES
XTST=YES  XPSB=YES  XPCT=YES XPPT=YES  XAPPC=YES  XUSER=YES
PCTEXTSEC=OVERRIDE    PCTCMDSEC=OVERRIDE    PCTRESSEC=OVERRIDE
DSNCHECK=NO   LTLOGOFF=NO
MAXSIGN=10,RETRY               MAXUSER=3000
```

To display the default Bypass List parameters issue:

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

The default parameters appear below.

```
BYPASS: RESOURCE=TRANID    NAMES:   CAQP    CATA    CATD
        CATP      CATR     CAUT     CCMF    CDBD    CDBN
        CDBO      CDBT     CDTS     CECS    CEGN    CEHP
        CEHS      CESF     CESN     CFTS    CGRP    CITS
        CLS1      CLS2     CLS3     CLS4    CMPX    CMSG
        CMTS      COVR     CPLT     CPMI    CSNE    CNPX
        CSM1      CSM2     CSM3     CSM4    CSM5    CSNC
        CSPG      CSPK     CSRK     CSPP    CSPQ    CSPS
        CSRS      CSSC     CSSF     CSSN    CSSX    CSSY
        CSTA      CSTB     CSTE     CSTP    CSTT    CSXM
        CSXX      CSZI     CVMI     CVST    CWTR    CXCU
        CXRT      TS       8888     9999
```

CICSTEST Facility The defaults for the CICSTEST facility are:

```
INITPGM=DFH          ID=K  TYPE=04
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,NODORMPW,NONPWR,NOIMSXTND
MODE=FAIL  LOGGING=INIT,MSG,SEC9,SMF
UIDACID=8  LOCKTIME=000  DEFACID=*NONE*  KEY=8
FACMATRX=NO         EXTSEC=YES
XJCT=YES  XFCT=YES  XCMD=YES XDCT=YES  XTRAN=YES
XTST=YES  XPSB=YES  XPCT=YES XPPT=YES  XAPPC=YES  XUSER=YES
PCTEXTSEC=OVERRIDE    PCTCMDSEC=OVERRIDE    PCTRESSEC=OVERRIDE
DSNCHECK=NO   LTLOGOFF=NO
MAXSIGN=10,RETRY                MAXUSER=3000
```

To display the default Bypass List parameters issue:

```
TSS MODIFY FAC(CICSTEST=BYPLIST)
```

The default parameters appear next.

```
BYPASS: RESOURCE=TRANID    NAMES:   CAQP    CATA    CATD
        CATP      CATR     CAUT     CCMF    CDBD    CDBN
        CDBO      CDBT     CDTS     CECS    CEGN    CEHP
        CEHS      CESF     CESN     CFTS    CGRP    CITS
        CLS1      CLS2     CLS3     CLS4    CMPX    CMSG
        CMTS      COVR     CPLT     CPMI    CSNE    CNPX
        CSM1      CSM2     CSM3     CSM4    CSM5    CSNC
        CSPG      CSPK     CSRK     CSPP    CSPQ    CSPS
        CSRS      CSSC     CSSF     CSSN    CSSX    CSSY
        CSTA      CSTB     CSTE     CSTP    CSTT    CSXM
        CSXX      CSZI     CVMI     CVST    CWTR    CXCU
        CXRT      TS       8888     9999
```

9.3.2 Defining a New Facility to the Matrix

In addition to the two CICS default facilities, CA-Top Secret also provides a total of 100 predefined facilities 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 options, such as MODE and LOCKTIME, should also be defined. The use of these options is described in detail in the *Control Options Guide.*

Once the new facility has been defined, the region can be associated with the facility using the method described on the next page.

9.3.3 Changing a Facility Entry

The procedure for modifying a facility entry is the same as the procedure for defining a new facility discussed in the previous section. Your security administrator simply makes changes to the existing USER or predefined facility via the FACILITY control option.

9.3.4 Associating a Region With a Facility

You may benefit from creating a separate CA-Top Secret facility for each CICS region, or for a group of related regions. For example, you may wish to group all CICS test regions together and associate them with the same CA-Top Secret facility. This requires a common entry in the Facilities Matrix for the CICS test regions.

When brought up, all CICS regions are associated with the CICSPROD facility by default. You can override this default association by using the MASTFAC attribute on the ACID for bringing up the region.

Suppose, for example, you have a test CICS region called CICST1 that is brought up as a batch job, and you wish to associate this region with the CICSTEST facility. You would first define the region control ACID that is used for the region, including the MASTFAC parameter, using these commands:

```
TSS CREATE(CICST1) NAME('CICS TEST REGION ACID') FAC(BATCH)
    PAS(NOPW,0) DEPT(deptacid) NOLCFCHK NODSNCHK NORESCHK
    MASTFAC(CICSTEST)
```

Note: The MASTFAC parameter can also be easily added to an existing ACID via the TSS ADD command:

```
TSS ADD(CICS1) MASTFAC(CICSTEST)
```

From this point on, when the CICST1 region is brought up, access to the region is governed by the options defined for the CICSTEST facility. All users who sign on to the region, as well as the resources they use, must be authorized for FACILITY(CICSTEST).

9.3.5 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 may 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, etc.) 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.

9.3.6 Defining the CICS Region Control ACID

Since a CICS region begins its execution as either a batch job or a started task, a CA-Top Secret ACID must be associated with each CICS region. This ACID must be able to access the BATCH facility, and must be authorized to all VSE 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 // ID USER=*acidname* batch JCL statement for the CICS region initiated as a BATCH job, or via the CA-Top Secret job submit inheritance.

The following examples define region acids and suggest ways to associate a CICS region acid (CICSP1 or CICST1) with either the CICSPROD or CICSTEST default facilities.

CICSPROD

```
TSS CREATE(CICSP1) NAME('CICS PRODUCTION REGION')
    FAC(BATCH) PAS(NOPW,0) DEPT(deptacid)  MASTFAC(CICSPROD)
    NORESCHK NOLCFCHK NODSNCHK
```

CICSTEST

```
TSS CREATE(CICST1) NAME('CICS TEST REGION')
    FAC(BATCH) PAS(NOPW,0) DEPT(deptacid) MASTFAC(CICSTEST)
```

FAC(BATCH) 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.

PAS(NOPW,0) Turns off password checking.

MASTFAC(facility) Must be specified with the CICS region. Users cannot log on unless MASTFAC is added to their user or profile record.

NOLCFCHK Bypasses LCF checking.

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

Note: You should 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.

9.3.7 Defining a CICS Default User

CICS TS 1.1 and above require the definition of a default user. This userid is used for all security checking done before anyone signs onto a terminal. It is specified in the CICS SIT parameter DFLTUSER. The userid coded must be a valid ACID and have access to the CICS facility. If the ACID is not properly defied or suspended, CICS will fail to initialize. It is recommended that the ACID used be given limited authority to reduce the potential for misuse. In addition, the ACID should be given the NOSUS (no suspend) attribute to keep it from being suspended.

9.4 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 may be authorized through LCF, or OTRAN. See "Protecting Online Transactions" in the *User Guide*.

9.5 Administration Requirements

The following sections detail how to define CICS to CA-Top Secret once the installation is complete.

9.5.1 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 ADD(acid) MASTFAC(facility)
```

9.5.2 Defining CICS

CICS can be defined to CA-Top Secret as a batch job.

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

```
TSS ADD(CICSP1) FACILITY(BATCH)
```

In addition, your systems programmer must code the // ID USER=*acidname* JCL statement, then submit the necessary CICS JCL. Using the example above as a reference, your programmer would code // ID USER=CICSP1 as a batch JCL statement.

9.6 CICS Table Changes

This section describes how to define CICS security parameters for the CA-Top Secret security 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 are 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 Chapter 16 and in SAMPJCL member TSSCSD.

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

- Destination Control Table (DCT)
- TERMINAL Definitions
- Temporary Storage Table (TST)
- TRANSACTION Definitions

9.6.1 Required Table Changes

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

9.6.1.1 The System Initialization Table (SIT)

The System Initialization Table (SIT) contains parameter settings for CICS initialization. Included in the SIT are security-related parameters. With CA-Top Secret, there are two choices for implementing 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: Once you have started CICS, all DFHSIT security parameters that were either specified in the DFHSIT Table or via the equivalent FACILITY suboptions are static. To pick up any DFHSIT changes you have made you must re-cycle your CICS region.

9.6.2 Activating CA-Top Secret Security

To use CA-Top Secret to secure your region, you must activate CA-Top Secret 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 next heading "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.

To use the facility matrix override feature, enter:

```
TSS MODIFY FAC(CICSPROD=FACMATRX=YES)
```

9.6.2.1 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= ASIS|ALWAYS

ASIS CA-Top Secret honors the CMDSEC value for all transactions; corresponds to PCTCMDSEC=HONOR.

ALWAYS CA-Top Secret overrides the CMDSEC value for all transactions and forces SPI security checking; corresponds to PCTCMDSEC=OVERRIDE.

RESSEC= ASIS|ALWAYS

ASIS CA-Top Secret honors the RESSEC value for all transactions.

ALWAYS CA-Top Secret overrides the RESSEC value for all transactions and forces resource security checking.

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

SEC= **YES|NO**

YES CA-Top Secret security is active for this region; corresponds to EXTSEC=YES.

NO CA-Top Secret security is inactive; corresponds to EXTSEC=NO.

XAPPC= **YES|NO**

YES Uses session security.

NO Session security is not used.

XCMD= **YES|NO**

YES All SPI commands are checked by CA-Top Secret.

NO All SPI commands are not checked by CA-Top Secret.

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

XDCT= **YES|NO**

YES Transient data entries for this region are checked by CA-Top Secret.

NO Transient data entries for this region are not checked by CA-Top Secret.

XFCT= **YES|NO**

YES File control entries for this region are checked by CA-Top Secret.

NO File control entries for this region are not checked by CA-Top Secret.

XJCT= **YES|NO**

YES Journal control entries for this region are checked by CA-Top Secret.

NO Journal control entries for this region are not checked by CA-Top Secret.

XPCT= YES|NO

YES Tranids specified on EXEC CICS START, INQ, SET, DISCARD, and COLLECT STATISTICS commands for this region are checked by CA-Top Secret

NO Tranids specified on EXEC CICS START, INQ, SET, DISCARD, and COLLECT STATISTICS commands for this region are not checked by CA-Top Secret

XPPT= YES|NO

YES Program entries for this region are checked by CA-Top Secret.

NO Program entries for this region are not checked by CA-Top Secret.

XPSB= YES|NO

YES Database PSB entries for this region are checked by CA-Top Secret.

NO Database PSB entries for this region are not checked by CA-Top Secret.

XTRAN= YES|NO

YES Transaction entries for this region are checked by CA-Top Secret, prior to execution.

NO Transaction entries for this region are not checked by CA-Top Secret, prior to execution.

XTST= YES|NO

YES Temporary storage keys for this region are checked by CA-Top Secret.

NO Temporary storage keys for this region are not checked by CA-Top Secret.

XUSER= YES|NO

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.

9.6.3 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.

9.6.3.1 TERMINAL Definitions

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

For the DFHTCT TYPE=TERM settings:

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 Has an associated CA-Top Secret keyword, OPPRTY. Refer to the *Command Functions Guide* for a detailed description of the OPPRTY keyword.

SIGNOFF This security parameter is honored by CA-Top Secret.

USERID The specified userid will be 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 will not be used for an output-only terminal.

9.6.3.2 TRANSACTION Definitions

Specify the following operands to indicate whether or not you want resource checking and SPI security checking done on this transaction.

RESSEC= YES|NO

YES Activates security checking for resources used by the transaction.

NO Bypasses security checking for resources used by the transaction.

CMDSEC= YES|NO

YES Activates command (SPI) security checking for the transaction.

NO Bypasses command security checking for the transaction.

9.6.3.3 DESTINATION CONTROL TABLE INTRAPARTITION Definitions

Trigger level transactions now run under the CICS default userid, not the id 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 IBM *CICS Resource Definition Guide* for details.

9.6.3.4 TEMPORARY STORAGE TABLE Definitions

If you want security checking done on your temporary storage queues, you must reassemble your TST table with the following type of entry:

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

See IBM *CICS Resource Definition Guide* for details.

9.6.4 Setting CA-Top Secret Security Inactive

There are two ways to deactivate security:

1. Specify SEC=NO in the DFHSIT option and FACMATRX=NO

   Use this method to turn security off by region.

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

   Use this method to turn security off by facility.

Chapter 10. Control Option Requirements

10.1 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 either existing CICS security or CA-Top Secret security is in effect.

10.1.1 Modes of Operation

CA-Top Secret supports four separate modes of operation for a CICS environment, as it does for all facilities. The modes are DORMANT, WARN, IMPLEMENT, and FAIL. Modes are assigned at five different levels:

Global The default for the entire CA-Top Secret community.
Example: **MODE(WARN)**

Facility Affects a particular facility within the community.
Example: **FAC(CICS=MODE=IMPL)**

Profile Affects a particular group of users attached to the profile.
Example: **TSS PER(PROF01) MODE(IMPL)**

User Affects a particular user within the community.
Example: **TSS PER(USER01) MODE(FAIL)**

Resource Forces a particular resource authorization to be processed in FAIL mode. Example:
TSS PER(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.

The following section describes how each CA-Top Secret security mode affects CICS users, based on the type of security checking requested. Modes of operation are listed for both resource and LCF (Limited Command Facility) transactions.

10.1.2 Modes for Resource Level Checking

The tables in this section define how CA-Top Secret modes are administered for resource checking.

The following table explains how modes for users and resources defined to CA-Top Secret are administered.

Table 10-1. Modes for Users Defined to CA-Top Secret—Resources Defined to CA-Top Secret

MODE	**ACTION**
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.

In addition to the information contained in the previous table, 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.

  For more information about these control options, refer to the *Control Options Guide.*

The next table explains how modes for users defined to CA-Top Secret and resources **not** defined to CA-Top Secret are administered.

Table 10-2. Modes for Users Defined to CA-Top Secret—Resources Not Defined to CA-Top Secret

MODE	**ACTION**
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.

  For more information about these control options, refer to the *Control Options Guide.*

10.1.3 Modes for LCF Checking

The following tables represent CA-Top Secret modes of operation for protection of transactions via the Limited Command Facility (LCF). Both inclusive and exclusive LCF lists are protected by CA-Top Secret.

- 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

Refer to the *User Guide* for a complete explanation of CA-Top Secret LCF protection.

Note: Transactions defined as OTRAN transactions override LCF transactions and follow the guidelines documented in the previous section, "Modes for Resource Level Checking".

The following table explains how modes for inclusive LCF lists are administered.

Table 10-3. Modes for Users Defined to CA-Top Secret for TRANS

Mode	Action
DORMANT	No security is active.
WARN	If the user has a TRANS LCF list for the facility, and the transaction ID accessed is found in that list, security checking is performed by CA-Top Secret.
IMPLEMENT	If the user has a TRANS LCF list, for the facility, and the transaction ID accessed is found in that list, security checking is performed by CA-Top Secret.
FAIL	If the user has a TRANS LCF list for the facility, and the transaction ID accessed is found in that list, security checking is performed by CA-Top Secret.

Here is an example of an LCF inclusive list:

```
TSS ADD(acid) TRANS(CICSPROD,(PAYT,MAIL,PAYP))
```

The following table explains how modes for exclusive LCF lists are administered.

Table 10-4. Modes for Users Defined to CA-Top Secret for XTRANS

Mode	Action
DORMANT	No security is active.
WARN	If the user specifies an XTRANS LCF list, and the transaction accessed is not found in the list for the facility, security checking is performed by CA-Top Secret.
IMPLEMENT	If the user is defined to CA-Top Secret, and the transaction is found in the XTRANS LCF list, the user is failed.
FAIL	If the user is defined to CA-Top Secret, and the transaction is found in the XTRANS LCF list, the user is failed.

Here is an example of an LCF exclusive list:

```
TSS ADD(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 by CA-Top Secret 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.

10.2 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.

Refer to the *Control Options Guide* for an explanation of how to set CA-Top Secret control options, what they can be used for, and a detailed description of each option.

10.3 CICS FACILITY Suboptions

CICS FACILITY suboptions can be divided into four groups:

- suboptions associated with the FACMATRX suboption,
- suboptions in the Bypass List,
- suboptions in the Protect List, and
- additional suboptions.

A list of CA-Top Secret FACILITY suboptions equivalent to the CICS DFHSIT security parameters appears below.

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

FACMATRX Suboptions: The FACMATRX suboptions are:

EXTSEC
PCTCMDSEC
PCTRESSEC
XAPPC
XCMD
XDCT
XFCT
XJCT
XPCT
XPPT
XPSB
XTRAN
XTST
XUSER

Resource List Suboptions: Bypass List suboptions are:

BYPADD(resource)
BYPREM(resource)
BYPLIST

Protect List suboptions are:

PROTADD(resource)
PROTREM(resource)

Additional Suboptions: Additional suboptions are:

DSNCHECK
LTLOGOFF
MAXSIGN
MAXUSER

For definitions, syntax and usage for each option, refer to the *Control Options Guide*.

10.3.1 Using Suboptions or DFHSIT Parameters

CA-Top Secret allows you to make a choice about 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. CA-Top Secret provides a FACILITY suboption that you can set to indicate whether you are using the DFHSIT security parameters or the CA-Top Secret FACILITY suboptions. This suboption is called FACMATRX.

- If you want to use the CA-Top Secret FACILITY suboptions to implement security for CICS initialization, you must set the FACMATRX suboption to YES. Then, security is implemented by the CA-Top Secret equivalent FACILITY suboptions.
- If you want to use the DFHSIT security parameters to implement security for CICS initialization, you must 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.

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

DFHSIT Parameters	FACILITY Suboptions
	FACMATRX
SEC=	EXTSEC=
XAPPC=	XAPPC=
XCMD=	XCMD=
XDCT=	XDCT=
XFCT=	XFCT=
XJCT=	XJCT=
XPCT=	XPCT=
XPPT=	XPPT=
XPSB=	XPSB=
XTRAN=	XTRAN=
XTST=	XTST=
XUSER=	XUSER=
CMDSEC=	PCTCMDSEC=
RESSEC=	PCTRESSEC=

10.3.1.1 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. Perform the following steps to disable CAIENF/CICS calls.

1. The facility entry must have FACMATRX set to YES.
2. 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.

Note: This procedure makes the Facilities Matrix entry XPARMs static. Therefore, to change a particular region, use the TSS Modify command to make the necessary changes to the FACILITY entry and then recycle the region so all changes appear.

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

Note: The TST intercepts are automatically installed so that CA-Top Secret can determine when the system is shutting down.

10.3.1.2 CICS Resource Lists

CA-Top Secret lets you construct two types of resource lists:

- The Bypass List
- The Protect List

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.

The Protect List: The Protect List is used to ensure that security checking is performed for certain resources that begin with a generic prefix appearing on the Bypass List.

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

Note: If a resource is added to both lists, the entry on the Protect List overrides the one on the Bypass List.

The following examples show how the Bypass and Protect Lists are used.

To avoid security checking for transactions beginning with XY, add an entry to the Bypass List as shown below.

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

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

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

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

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

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

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

A detailed discusssion of these parameters appears in Chapter 12, "Implementing Security". The next sections briefly cover how these parameters can be used to avoid security checking. Note that the same parameters can be used for the Protect List.

10.3.1.3 Bypassing Security for CEMT Commands

Use the CEMT=*action* parameter that contains the Extended Master Terminal Command actions for which you want to bypass security checking. Valid actions are:

ADD
INQUIRE
PERFORM
REMOVE
SET
DISCARD

For example, to allow access to all CEMT INQUIRE commands, enter:

```
TSS MODIFY FAC(cicsfac=BYPADD(CEMT=INQUIRE))
```

Note: To bypass SET you also need to add INQUIRE to the Bypass List.

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

10.3.1.4 Bypassing Security for SPI Commands

Use the SPI resource to bypass security for SPI commands.

For example, to bypass **all** EXEC CICS and CEMT INQUIRE SYSTEM commands, enter:

```
TSS MODIFY,FAC(cicsfac=BYPADD(SPI=SYSTEM))
```

10.3.1.5 Bypassing SPOOLWRITE Job Submission Protection

In CICS TS 1.1 and above, there is an SPI check to determine if the user has permission to open the SPOOL data set. To bypass this check, execute the followind command:

```
TSS MODIFY,FAC(cicsprod=BYPADD(SPI=PWRSPOOL))
```

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

```
TSS MODIFY,FAC(cicsprod=BYPADD(SPI=SPOOLSUB))
```

10.3.1.6 Bypassing Transaction Security

To bypass transaction security, add an entry to the TRANID or TRAN parameter of the CA-Top Secret Bypass List. The TRANID parameter contains transaction ID entries that will bypass **all** security checking for the transaction. The default entries are:

```
BYPASS: RESOURCE=TRANID    NAMES:   CAQP    CATA    CATD
        CATP      CATR     CAUT     CCMF    CDBD    CDBN
        CDBO      CDBT     CDTS     CECS    CEGN    CEHP
        CEHS      CESF     CESN     CFTS    CGRP    CITS
        CLS1      CLS2     CLS3     CLS4    CMPX    CMTS
        COVR      CPLT     CPMI     CQRY    CRDR    CRSQ
        CRSR      CRSY     CRTE     CRTR    CSAC    CSCY
        CSFU      CSGM     CSIR     CSGX    CSJC    CSKP
        CSLG      CSMI     CSNE     CNPX    CSM1    CSM2
        CSM3      CSM4     CSM5     CSNC    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
```

Multiple transactions (up to four) can be specified on one line for the bypass list, by entering:

```
TSS MODIFY,FAC(cicsfac=BYPADD(TRANID=(trn1,trn2,trn3,trn4)
```

The difference between the Bypass List parameters TRAN and TRANID is that the entries for TRAN contain TRANIDs that will bypass resource OTRAN or LCF security checking only. Entries in the TRANID Bypass List contain TRANIDs 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).

As CICS issues both a standard transaction check and an additional PCT check for the transaction CEDF and you wish to bypass checking on that transaction name, you must put it in the TRAN bypass list, not the TRANID bypass list.

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 the execution of the transaction.

10.3.1.7 Bypassing Terminal Security

The TCT parameter contains terminal entries that will bypass CA-Top Secret security checking where VTAM is an eight-character NETNAME.

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

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

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

10.3.1.8 Bypassing LOCKTIME Security

The LOCKTIME parameter contains terminal entries or transaction IDs that will not be 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 is an eight-character NETNAME.

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

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

To bypass LOCKTIME security for transaction PUBL, enter:

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

10.3.1.9 Bypassing Security for Specific Resources

You can **selectively** bypass security for specific resources using these parameters:

DCT Contains transient data entries that will not be checked by CA-Top Secret.

DSN Contains data sets that will not be checked by CA-Top Secret. The DSNCHECK= suboption must be set to YES. The entries in this Bypass List are not the actual data set names, but the File Control Table entries associated with the data sets.

FCT Contains File Control Table entries (DDnames) that will not be checked by CA-Top Secret. The DSNCHECK= suboption must be set to NO.

JCT Contains Journal Control Table entries (journal names) that will not be checked by CA-Top Secret.

PCT Contains interval control started transaction identifiers that will not be checked by CA-Top Secret.

PPT Contains program entries that will not be checked by CA-Top Secret.

PSB Contains PSB entries that will not be checked by CA-Top Secret.

TRAN Contains transaction identifiers that will not be checked by CA-Top Secret.

TST Contains Temporary Storage entries (queue names) that will not be checked by CA-Top Secret.

10.3.2 Additional Suboptions

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

10.3.2.1 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 (FAC(CICS=MAXUSER=500))
```

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

10.3.2.2 Controlling Simultaneous User Signon

Use the MAXSIGN= suboption to restrict the number of simultaneous user signons; the default value is 50. This suboption allows you to set a threshold for the number of concurrent signons, and controls the action taken if the threshold is exceeded. For example, you can restrict the number of concurrent signons to a CICS region called CICSPAY to a threshold of 75 using the TSS MODIFY command like this:

```
TSS MODIFY (FAC(CICS=MAXSIGN=(75,kill|retry))
```

The KILL option abends the signon transacton; RETRY requeues the signon transaction. See "CICS Related Suboptons" in the *Control Options Guide*.

Note: The SIGN(S) control option disallows simultaneous logons for the same ACID for each CICS region running under a specified facility. With multiple CICS regions under one facility, a single ACID can sign on, one terminal for each region.

10.3.2.3 Securing Data Set Names Instead of FCTs

Use the DSNCHECK(YES|NO) suboption to tell CA-Top Secret to perform security checking on either 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 DSN name, specify DSNCHECK=YES.

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

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

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

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

```
TSS MODIFY FAC(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: After changing a FACILITY suboption, you must recycle your CICS region for the changes to take effect.

Note: If the FCT is remote, all DSN checks will be bypassed. You must remove the CSMI transaction from the FACILITY Bypass List to provide protection for remote DSNs. Security checking is performed in the region where the FCT resides.

CICS data set protection (DSNCHECK=YES) does not protect DL1 databases. These are validated under the CICS region control ACID.

10.3.2.4 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:

1. Specify XUSER=YES in the DFHSIT, or
2. Specify FACMATRX=YES, then specify XUSER=YES in the Facilities Matrix.

If surrogate user checking is being used, 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 command EXEC CICS START TRANID('ABC') USERID('STARUSER') is issued, you must add authority to CURRUSER for the surrogate of STARUSER as shown below.

```
TSS ADD(CURRUSER) SURROGAT(STARUSER)
```

If the userid is omitted, the userid is inherited from the transaction that issued the START command.

To activate surrogate or background security enter:

```
TSS MODIFY FAC(CICSPROD=FACMATRX=YES,XUSER=YES)
```

10.3.2.5 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 CA-Top Secret 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)
```

10.3.2.6 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:

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

To perform security checking for all resources enter:

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

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

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

10.3.3 Transaction Validation

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

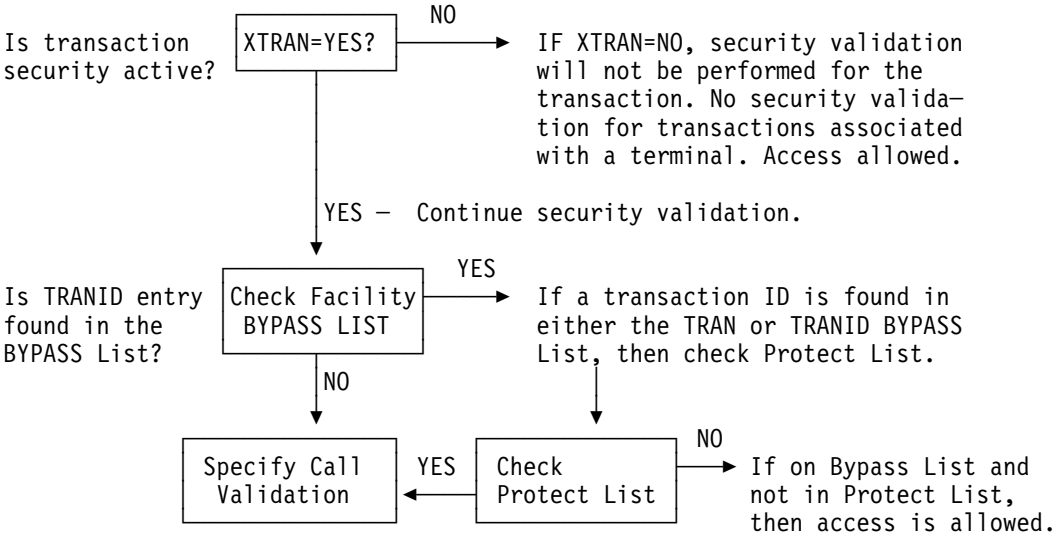


Figure 10-1. Transaction Validation Logic Flow

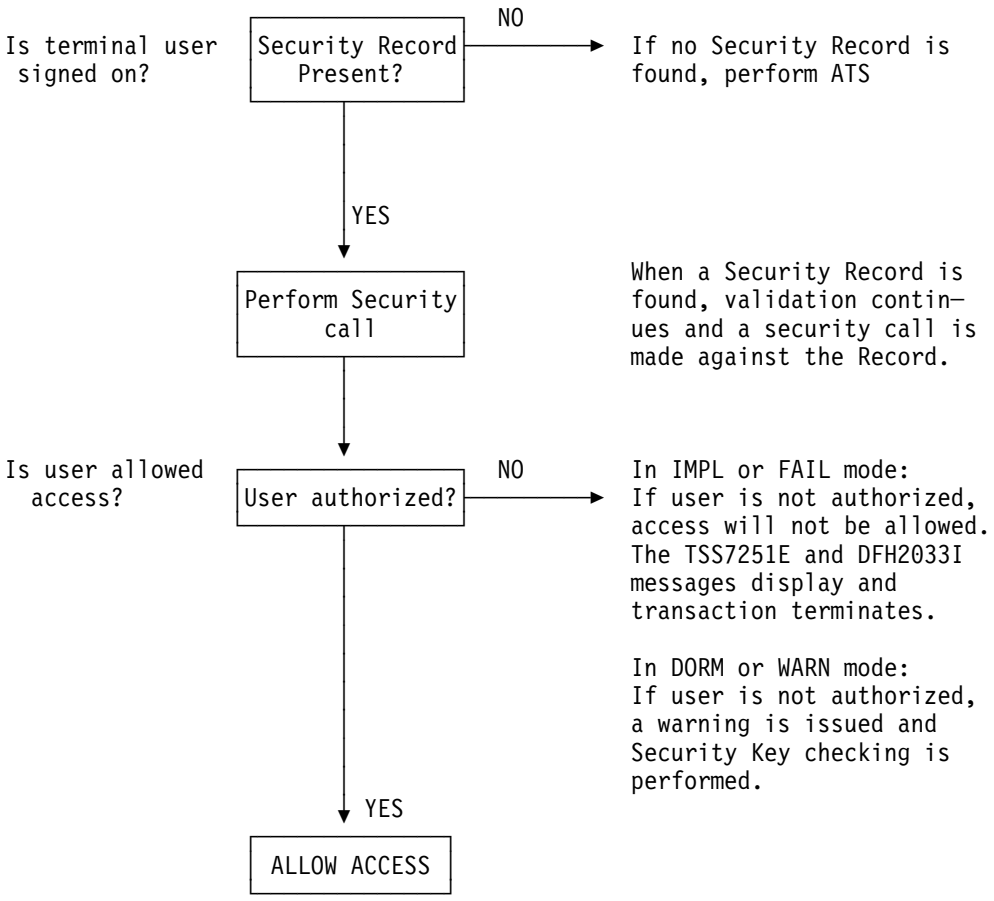


Figure 10-2. Security Call Processing

The following chart illustrates CA-Top Secret CICS security call processing.

Chapter 11. Security for a Multi-system Environment

This chapter details CICS and CA-Top Secret security for the Intersystem Communication (ISC) and Multiregion Operation (MRO) environments.

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, read the chapter on "Security in the Intercommunication Environment" in IBM's *Intercommunication Facilities Guide.*

You can define additional levels of security for ISC and MRO 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

Note: 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 more information on using TSSUTIL, refer to your *Report and Tracking Guide.*

11.1 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 chart shows which CICS parameters must be defined (via RDO or RDM) to set up MRO or ISC in your CICS region.

Definition	**Using RDO**
CONNECTION	ATTACHSEC
SESSION	USERID

11.2 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 either of these 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.

11.2.1 For MRO Connections

An example of how bind-time security works for MRO connections is shown here in the following figure using RDO parameters.

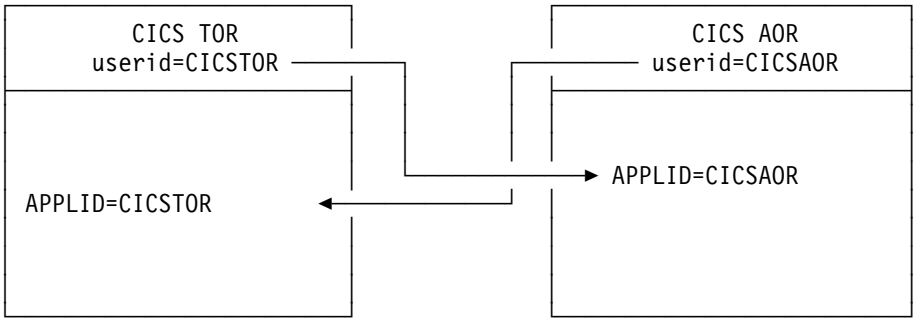


Figure 11-1. MRO Bind-time Security

11.2.2 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.

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)
```

Figure 11-2. Defining ISC External Bind-time Security to CICS

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

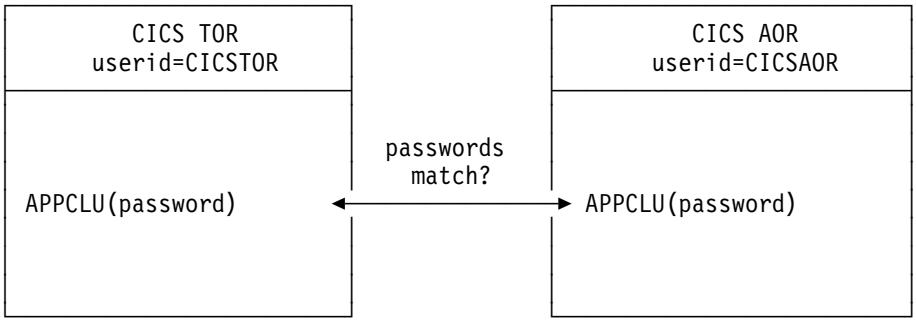


Figure 11-3. ISC External Bind-time Security

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).

11.3 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 either 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 may 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.

11.3.1 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 region userid.

The following figure shows how to define link security for both MRO and ISC connections. Note that you must specify a **userid.**

```
RDO definition

DEFINE
  SESSION
   .
  USERID(userid)
   .
   .
```

Figure 11-4. Defining Link Security to CICS

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

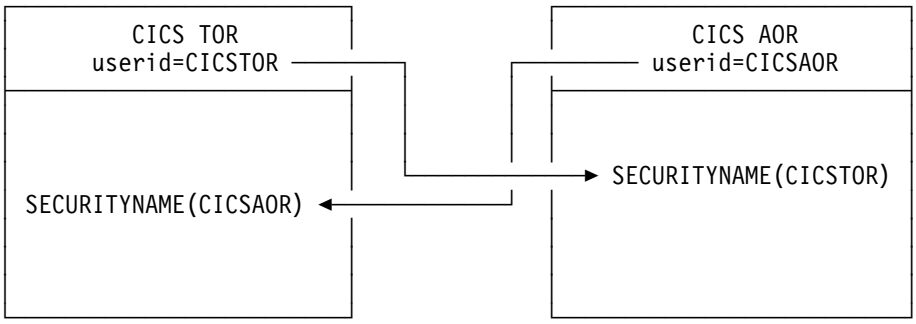


Figure 11-5. Link Security for MRO and ISC

In this 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.

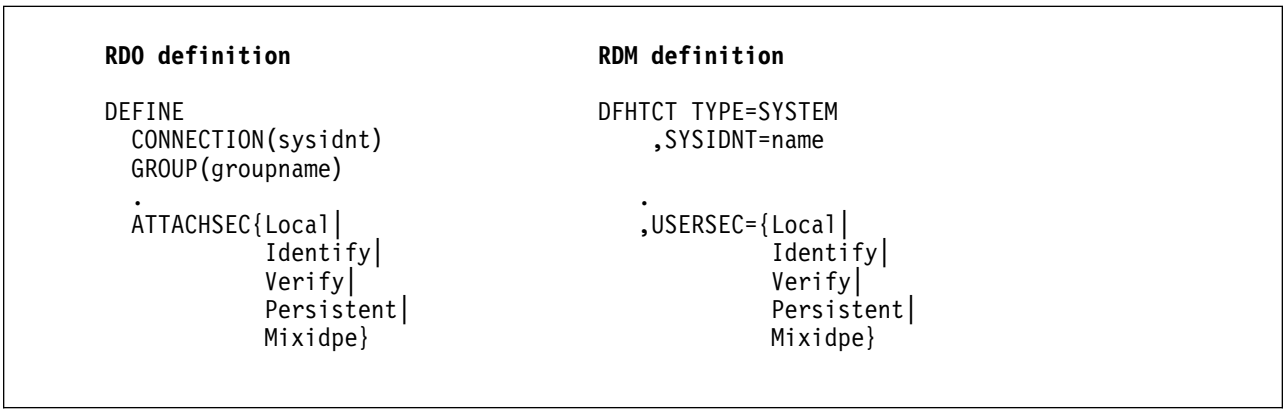
The PAY transaction (depending on the mode) will not be able to execute in the AOR region if the CICS region ACID CICSTOR:

- is not permitted to the PAY transaction, or
- is not defined with the NORESCHK or NOLCFCHK attributes.

11.4 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

```
DEFINE
  CONNECTION(sysidnt)
  GROUP(groupname)
  .
  ATTACHSEC{Local|
           Identify|
           Verify|
           Persistent|
           Mixidpe}
```

RDM definition

```
DFHTCT TYPE=SYSTEM
    ,SYSIDNT=name

   .
   ,USERSEC={Local|
            Identify|
            Verify|
            Persistent|
            Mixidpe}
```

Figure 11-6. Defining Attach-time Security

There are five levels of attach-time security: LOCAL, IDENTIFY, VERIFY, PERSISTENT, and MIXIDPE.

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.

How CA-Top Secret relates to these levels is explained below; for complete information, refer to IBM's *Intercommunication Facilities Guide.*

11.4.1 Local Security Considerations

If you include a remote resource name in your CICS resource definitions, CA-Top Secret performs security checking locally against that remote resource, just as if the remote resource were a local one. The security check (on the remote side) is performed against the **userid** for the session or, if **userid** was not specified, the CICS region ACID is used.

The following resource areas are secured under attach-time LOCAL security:

- All locally-defined resources
- Remote-defined transactions (transaction routing requests)
- Remote-defined files, transient data, and temporary storage queues (function shipping requests).

However, a LOCAL specification does not provide security coverage for the following resource areas:

- Remote-defined transactions that have alias resource names. The alias resource name is not protected.
- Remote data sets (DSNCHECK=YES)
- Any resource request originating from the remote side.

11.4.2 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 ISRDELAY) has elapsed since the previous attach request from this user.

  - The link to the remote CICS region has been broken.
  - The CICS system has been recycled.

- Some CICS releases will 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. Refer to the *CICS Interregion Communication Guide* for your appropriate CICS release to determine if this is a situation you must allow for. This ACID should be coded with a non-expiring password. The following example shows that there is no need for permission just to add the facility the SYSID is associated with.

```
TSS CREATE('SYSID') NAME('CICS SYSID ACID')
    FAC(CICS) PAS(XXXX,0) DEPT(deptacid)
```

Chapter 12. Implementing Security

The previous chapters explained how to get your CICS system running in a secured environment. This chapter details the day-to-day operations you need to administer your CICS system in a secured environment, including:

- 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. Refer to the *User Guide* for details on PassTicket.

12.1 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.

The sign on procedures listed here include:

- Standard CICS Signon
- New Password Signon
- Automatic Terminal Signon (ATS)
- Interactive Interface Signon (IUI)

12.1.1 Using CESN

You can sign on to CA-Top Secret CICS via 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 via screen prompts or by stringing the commands together.

12.1.1.1 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 up to eight-character alphanumeric userid or the defined CA-Top Secret user ACID.

PS= The password associated with the user ACID (maximum eight-characters).

NEWPS= The "new" password replacing your lost, expired, or existing password.

12.1.1.2 By Screen Prompt

At most sites, the signon screen is automatically displayed. If it is not, you can sign on to CICS using CESN via screen prompts. The CESN signon screen is shown below; refer to IBM's *CICS-Supplied Transactions* guide for details on CESN signon procedures.

The signon screen for CESN looks like this:

```
CESN - CICS/VS SIGNON - ENTER USERID AND PASSWORD

     USERID: _                GROUP: _

     PASSWORD:
     NEWPASSWORD:
```

Make the appropriate entries in each field where:

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

GROUP= Not needed for CA-Top Secret; ignore.

PASSWORD The password associated with the user ACID (maximum eight-characters).

NEWPASSWORD The new password associated with the user ACID (maximum eight-characters) which replaces a lost, expired, or existing password.

The standard signon procedure using the CESN screen prompts is:

1. Type your CA-Top Secret ACID (maximum eight-character alphanumeric) in the USERID: field.
2. Type your selected password (maximum eight-character alphanumeric). The characters in the password field will not display.
3. Then press ENTER.
4. 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
```

These messages are automatically cleared after a few seconds. See the *Messages and Codes Guide* for details about messages.

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

12.1.2 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 will be 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 for VTAM is an eight-character netname.

Your installation selects which terminals are eligible 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 CRE(K067T018) NAME('EMAIL SYSTEM GR 1')
    FAC(CICSPROD) DEPT(CIPCC)
    PASSWORD(NOPW,0)
    SOURCE(K067T018)
```

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

Note: ATS is not performed if: Validation is not required for a transaction being entered at a terminal; XTRAN=NO; or the transaction is in the Bypass List.

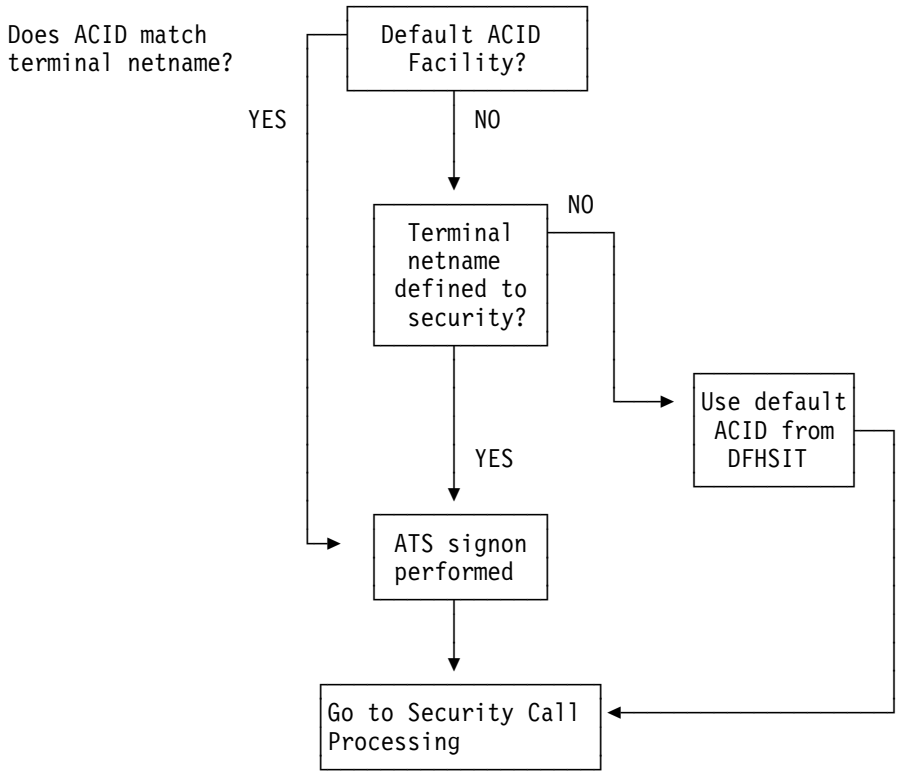


Figure 12-1. Automatic Terminal Processing (ATS)

12.1.3 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 ADD(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.

12.1.4 Signon-generated Return Codes

The following table lists the reponse codes and descriptions of the ESMRESP, EIBRESP, EIBRESP2 return codes that can be generated by an EXEC CICS signon.

Description	Response Code	Return Code
Undefined ACID	ESMRESP	X'04'
	EIBRESP	X'46'
	EIBRESP2	X'08'
Password Missing	ESMRESP	X'08'
	EIBRESP	X'46'
	EIBRESP2	X'01'
Password Incorrect	ESMRESP	X'08'
	EIBRESP	X'46'
	EIBRESP2	X'02'
Password Expired/New Password Missing	ESMRESP	X'0C'
	EIBRESP	X'46'
	EIBRESP2	X'03'
New Password Invalid	ESMRESP	X'10'
	EIBRESP	X'46'
	EIBRESP2	X'04'
Suspended ACID	ESMRESP	X'1C'
	EIBRESP	X'46'
	EIBRESP2	X'13'
Access Denied to Terminal	ESMRESP	X'30'
	EIBRESP	X'46'
	EIBRESP2	X'10'
Access Denied to Facility	ESMRESP	X'34'
	EIBRESP	X'46'
	EIBRESP2	X'11'

12.1.5 Interactive Interface Signon

Many sites require some or all users to signon to the Interactive Interface (IUI), an IBM productivity option that facilitates VSE/ESA system administration and provides a pathway into CICS. With external security active, CA-Top Secret will attempt to logon all users that enter the system using the IUI. The signon screen for the Interactive Interface follows:

```
IESADMSO1                        VSE/ESA ONLINE
    5690-VSE and Other Materials (C) Copyright IBM Corp. 1997 and other dates

    VV    VV    SSSSS    EEEEEEE          ++
    VV    VV   SSSSSSS   EEEEEEE         ++
    VV    VV  SS         EE             ++    EEEEEEE    SSSSS       AA
    VV    VV   SSSSSS    EEEEEE        ++     EEEEEEE   SSSSSSS     AAAA
    VV    VV    SSSSSS   EEEEEE       ++      EE       SS          AA  AA
     VV  VV          SS  EE          ++       EEEEEE    SSSSSS    AA    AA
      VVVV     SSSSSSS   EEEEEEE    ++        EEEEEE     SSSSSS   AA     AA
       VV       SSSSS    EEEEEEE   ++         EE             SS  AAAAAAAA
                                   ++          EEEEEEE   SSSSSSS   AA    AA
                                  ++           EEEEEEE    SSSSS    AA    AA

        Your terminal is L304 and its name in the network is D72L304
        Today is 08/04/1998   To sign on to DBDCCICS  --  enter your:

       USER-ID........ ________   The name by which the system knows you .
       PASSWORD.......            Your personal access code.

PF1=HELP       2=TUTORIAL                4=REMOTE APPLICATIONS     6=ESCAPE(U)
                           9=Escape(m) 10=NEW PASSWORD
```

Make the appropriate entries in each field where:

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

PASSWORD The password associated with the user ACID (maximum eight characters).

The standard signon procedure using the IUI screen prompts is:

1. Type your CA-Top Secret ACID (maximum eight-character alphanumeric) in the USER-ID field.
2. Type your selected password (maximum eight-character alphnumeric) in the PASSWORD field. The characters in the password field will not display.
3. Press Enter.
4. When signon is successful the following 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
```

These messages are automatically cleared after a few seconds. See the *Messages and Codes Guide* for details about messages.

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

12.1.5.1 IUI Signon Special Considerations

- A VSE control file user profile is not required for a user to logon through the IUI signon panel. If the user is defined to the VSE control file, he will be presented with his initial panel definition. If the user is NOT defined, the following display will occur, and the user will be placed in native CICS:

```
----------------------
There is no profile information for your user ID.
----------------------

----------------------
You are authorized to use the system, however no user profile
was found. You cannot use functions of the Interactive Interface.
----------------------

----------------------
Specifics about the error have been logged for
analysis and action by your System Administrator.
----------------------
```

- Password expiration will occur during IUI signon processing, at which time a new password can be selected. Refer to *Password Expiration* for more information. Additionally, the new password process can be forced using the PF10 key on the IUI signon panel. If your CA-Top Secret password should expire during the current signon attempt, or the PF10 key is selected, the following panel will be displayed:

```
IESADMSO2             VSE/ESA SIGN-ON WITH NEW PASSWORD

   Enter your new password in both places below then enter your current
   password for sign-on verification.
   Then press the ENTER key.

    NEW PASSWORD ==>            3-8 characters

    NEW PASSWORD ==>            Re-Enter new password for verification

    OLD PASSWORD ==>            Current password



 PF1=HELP                   3=END
```

  Enter the required information to establish the new password. Please refer to 12.2.2, “Changing Passwords” on page 12-13 and 12.2.3, “Random Password Generation” on page 12-14 for additional information concerning the new password process.

- All users of the Interactive Interface must have a three-character CICS OPID associated with their CA-Top Secret ACID. The OPIDENT keyword can be used to assign this attribute to each user that requires it. If a user does not have this attribute, the IUI logon will complete successfully, but the first attempt to use any IUI panel

option or PF key will force the user to be logged off and sent back to the IUI Signon panel.

12.2 Administering Passwords

This section covers:

- How to assign a new password
- How to change a password
- Random password generation
- Expired passwords
- Forgotten passwords

Because CA-Top Secret offers an extensive variety of password controls, you should develop password usage strategies particular to your site.

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

1. Memorize your password.
2. All written records of your password should be destroyed.
3. **Do not** post your ACID or password near the video terminal, disks, cabinets, bulletin boards, or other areas accessible to unauthorized individuals.
4. **Do not** maintain your password in an unprotected data set where others could view it.
5. **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.
6. 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. Refer to the *Control Options Guide* for more information about password administration.

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

12.2.1 New Password Signon Procedure

You can assign a new password when you sign on using either screen prompts or a command string. The following new password signon procedure uses the CESN screen prompts shown below.

```
CESN - CICS/VS SIGNON - ENTER USERID AND PASSWORD

     USERID: _                GROUP: _

     PASSWORD:
     NEWPASSWORD:
```

1. Type your CA-Top Secret ACID (maximum eight-character alphanumeric) in the USERID: field.
2. Type your selected password (maximum eight-character alphanumeric).
3. Type your new password (maximum eight-character alphanumeric). The characters in the password field will not display.
4. When all the information press the ENTER key.
5. When the password is changed, these messages are displayed:

```
TSS7030I Password Changed
TSS7000I acidname Last-Used mm/dd/yy hh:mm System=xxxx Facility=xxxxxxxx
TSS7001I Count=xxxxx Mode=xxxx Locktime=xxxxx Name=xxxxxxxxxxxxxxxxxxxx
```

12.2.2 Changing Passwords

You can change your password using the CESN transaction via screen prompts or a command string. The following procedure uses the CESN screen prompts.

```
CESN - CICS/VS SIGNON - ENTER USERID AND PASSWORD

     USERID: _                GROUP: _

     PASSWORD:
     NEWPASSWORD:
```

1. Type your CA-Top Secret ACID (maximum eight-character alphanumeric) in the USERID: field.
2. Type your existing password (maximum eight-character alphanumeric). The password will not display.
3. Type your **new** password (maximum eight-character alphanumeric) in the NEWPASSWORD: field. The following message is displayed when the NPWR FACILITY suboption is in effect:

```
TSS7197I  Enter NEW Password Again for Reverification
```

4. Type your **new** password again. The following messages are displayed:

```
TSS7030I Password Changed
TSS7000I acidname Last-Used mm/dd/yy hh:mm System=xxxx Facility=xxxxxxxx
TSS7001I Count=xxxxx Mode=xxxx Locktime=xxxxx Name=xxxxxxxxxxxxxxxxxxxxx
```

12.2.3 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). The procedure is:

1. Type your CA-Top Secret ACID in the USERID: field.
2. Type 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 following messages are displayed:

```
TSS7030I  Password Changed
TSS7020I  Random Password About to be Displayed. Hit Enter to Continue.
```

4. Press the ENTER key again. The following messages are displayed:

```
TSS7021I Your New Password is xxxxxxxx
TSS7022I Memorize Password & Hit Enter Key - DO ***NOT*** RECORD
```

   Memorize the password generated for you (indicated above as xxxxxxxx.) Without this password you will not be able to sign on again.

5. 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
```

12.2.4 Password Expiration

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

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

mm/dd/yy displays the month, day, and year that your password will expire.

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. This procedure was discussed earlier in this chapter in the section "Changing Passwords".

12.2.5 Lost Passwords

If you forget your password, CA-Top Secret does not permit you to access a facility. Do not try to guess your password. Notify the appropriate security administrator immediately to have a new password assigned to you.

12.3 Administering Transaction Security

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

12.3.1 OTRAN Security

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

Note: A transaction protected via OTRAN will not go through LCF checking.

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

```
TSS ADD(acid) OTRAN(transaction)
```

After which you could allow users access to the transaction by entering:

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

See the *User Guide* for a detailed discussion on OTRAN security.

12.3.2 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 either 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 ADD(acid) TRANS(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.

See the *User Guide* for a detailed discussion on LCF security.

12.4 Administering Resource Level Security

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

12.5 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.

Refer to the *User Guide* for a detailed description of the keywords used with SDT.

12.5.1 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. They are:

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).

Implementing RLP is a four step process:

Step 1 Gather Information

Step 2 Enter Definitions

Step 3 Permit Access to the Defined Records

Step 4 Enable Protection

Each step is described in detail in the following sections.

12.5.1.1 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.

Step 1 Determine which of your applications would benefit from RLP.

Step 2 Meet with the programmers to gather information about the application (like FCT name, field names, positions, data types, length of field, and selection criteria).

Step 3 Become familiar with the application.

Step 4 Plan the details needed to implement RLP for this application For example, you may decide on a selection criteria that limits the user to viewing only the records within their departmental scope.

Step 5 Determine who will be the administrator(s) for implementing RLP and give them MISC3(SDT) authority.

12.5.1.2 Enter Definitions

Once you have completed the preliminary steps as outlined in the previous section, you are ready to begin entering the definitions into the SDT. All definitions are entered using the TSS ADD(SDT) command.

Note: Depending on the number of records and fields you are protecting, these steps can be labor intensive.

Step 1 Define the RECORD definitions to the SDT. A sample RECORD definition is shown below.

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 will be used (referenced) in the selection process.
- What is the format of the data in the fields?
- What sizes are the fields?

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

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

Step 2 Define the SELECT expressions to the SDT that you will be using on the PERMIT command. A sample definition is shown below.

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?

```
TSS ADD(SDT) SELECT(dp1000)
             SELDATA('IF dept GE "1000" AND dept LE "1099")
```

Step 3 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 shown below.

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

Step 4 Check your work by listing the SDT records you just created. To list all records, use the command shown below.

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

Step 5 To correct any errors, first use the TSS REMOVE(SDT) command to remove any field you wish to modify, then use the TSS ADD(SDT) command to add the field you want to replace.

Step 6 When you are satisfied that everything is correct, refresh the SDT in-core tables using the command shown below.

```
TSS MODIFY(SDTTABLE)
```

12.5.1.3 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 may have for these FCTs.
2. Then, re-PERMIT the FCTs using the SELECT and/or MASKREC clauses. A sample PERMIT command is shown below.

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

12.5.1.4 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.) Use the command shown below to enable RLP in the CICS region.

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

12.5.1.5 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) ACC(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 may contain the masking values.
- Make certain that the data types specified in either the MASKDATA or RECDATA definitions match the data types of those contained on the actual file record.
- When RLP=YES, NOTAUTH conditions may have EIBRESP2=0000 instead of the expected EIBRESP2=101.

12.6 Administering Screen Level Protection (SLP)

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

- Define the screen (map) to CA-Top Secret Static Data Table (SDT). SDT record types of MAPREC and SELECT are required for SLP processing.

  Note: The SDT provides the framework on which field(s) within the terminal screen (map) are defined to SLP, and the selection criteria that identify if the map is allowed to be displayed.

- A SELECT clause must be specified on either the OTRAN or PPT permit of the transaction or program that is participating in SLP. The SELECT clause contains the name of the SDT SELECT record that is to be used as input into the SLP validation process.

  For example:

```
TSS PERMIT(userid) OTRAN(ABC) SELECT(MAPABC)
```

Refer to the *User Guide* to obtain more details on the SDT and SLP requirements.

12.7 Administering Terminal Security

The following sections explain how to:

- Preset terminal security
- Restrict terminal use
- Secure sequential terminals
- Secure VSE console terminals
- Control when inactive or unattended terminals are locked

12.7.1 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 below.

```
TSS ADD(CICSDEPT) SURROGAT(WAREHOUS.DFHINSTAL)
TSS PER(CICSADM) SURROGAT(WAREHOUS.DFHINSTAL) ACC(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.

12.7.2 Restricting Terminal Access

CA-Top Secret restricts selected VTAM 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 will be logged off after signon.

Refer to the TERMINAL keyword of the TSS ADD and PERMIT commands in the *Command Functions Guide* for details.

12.7.3 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.

12.7.4 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:

1. When you signon a terminal, CA-Top Secret begins to monitor LOCKTIME thresholds.
2. When the LOCKTIME threshold is expired, at the next action key (ENTER, CLEAR, PF key, etc.) the terminal screen is cleared and you are prompted for your password.

   You can enter your password, CESF, or press the CLEAR key.
3. 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.
4. 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.

12.8 Administering Transient Data Security

You can designate an ACID to be associated with transactions initiated by an intrapartition transient data queue with a trigger level greater than 1. The value may come from one of the following sources:

- If the userid is specified on the TYPE=INITIAL or TYPE=INTRA macro, it will be signed on and used for security.
- If the destination is associated with a terminal (DESTFAC=TERMINAL), the userid will be 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 will be used.

12.9 Administering CICS Command Security

CA-Top Secret provides the SPI resource for added security checking. With the CA-Top Secret SPI resource you can secure the following:

- CEMT commands
- EXEC CICS INQUIRE and SET commands
- EXEC CICS ENABLE, DISABLE, and EXTRACT commands
- EXEC CICS SPOOLOPEN command.

12.9.1 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.

Table 12-1 *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]
```

Table 12-2 *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 Table 12-3 *CEMT Secondary Resource Checks*.

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

Table 12-1. SPI Access Levels for CEMT

CEMT Action	**SPI Access Level**
ADD	SET
INQUIRE	INQUIRE
PERFORM	PERFORM
REMOVE	SET
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.

12.9.1.1 INQUIRE and SET Commands

The following table lists the CEMT command keywords and their associated SPI resource names. for specified resources, secondary resource checking applies. See the section Secondary Resource Checks later in this chapter for details.

Table 12-2. SPI Keywords for CEMT INQUIRE and SET Actions

Command Keyword	**SPI Keyword**
'Blanks' (default)	SPI(SYSTEM)
AUTINSTMODEL	SPI(AUTINSTM)
AUTOINSTALL	SPI(AUTOINST)
AUXTRACE	SPI(TRACEDES)
CONNECTION	SPI(CONNECTI)
DELETESHIPPED	SPI(DELETESH)
DLIDATABASE	SPI(DLIDATAB)
DSAS	SPI(SYSTEM)
DSNAME	SPI(DSNAME)
DUMP	SPI(DUMP)
DUMPDS	SPI(DUMPDS)
FECONNECTION	SPI(FEPIRESO)
FENODE	SPI(FEPIRESO)
FEPOOL	SPI(FEPIRESO)
FEPROPSET	SPI(FEPIRESO)
FETARGET	SPI(FEPIRESO)
FILE	SPI(FILE)
GTFTRACE	SPI(TRACEDES)
INTTRACE	SPI(TRACEDES)
IRBATCH	SPI(IRBATCH)
IRC	SPI(IRC)
JOURNALNUM	SPI(JOURNAL)

Table 12-3. SPI Keywords for CEMT INQUIRE and SET Actions

Command Keyword	SPI Keyword
LINE	SPI(LINE)
MODENAME	SPI(MODENAME)
MONITOR	SPI(MONITOR)
NETNAME	SPI(TERMINAL)
PARTNER	SPI(PARTNER)
PITRACE	SPI(PITRACE)
PROFILE	SPI(PROFILE)
PROGRAM	SPI(PROGRAM)
STATISTICS	SPI(STATISTI)
SYSDUMPCODE	SPI(SYSDUMPC)
SYSTEM	SPI(SYSTEM)
TASK	SPI(TASK)
TCLASS	SPI(TCLASS)
TDQUEUE	SPI(TDQUEUE)
TERMINAL	SPI(TERMINAL)
TRANSACTION	SPI(TRANSACT)
TRDUMPCODE	SPI(TRANDUMP)
TSQUEUE	SPI(TSQUEUE)
VOLUME	SPI(VOLUME)
VTAM	SPI(VTAM)

Examples for securing CEMT INQUIRE and SET commands appear next.

CEMT INQUIRE

Using the commands below, 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 ADD(deptacid) SPI(SYSTEM)

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

CEMT INQUIRE DUMP

Using the commands below, the user only has permission to execute CEMT INQUIRE DUMP commands.

```
TSS ADD(deptacid) SPI(DUMPDS)

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

CEMT INQUIRE AUTOINSTALL

Using the commands below, the user only has permission to execute CEMT INQUIRE AUTOINSTALL commands.

```
TSS ADD(deptacid) SPI(AUTOINST)

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

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

CEMT SET VTAM OPEN

Using the commands below, the user only has permission to execute CEMT SET VTAM OPEN commands.

```
TSS ADD(deptacid) SPI(VTAM)

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

12.9.1.2 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 Table 12-3 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 or not SPI access to INQUIRE or SET the CICS resource class has been granted.

Table 12-4. CEMT Secondary Resource Checks

CEMT Keyword	**Secondary Resource Type**
DSN	DATASET
FILE	FCT
JOURNAL	JCT
PROGRAM	PPT
QUEUE	DCT
TRANSACTIONS	OTRAN or LCF
VOLUMES	VOLUMES

Note:

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

```
TSS MODIFY FAC(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 FAC(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 for securing CEMT secondary resources appear next.

CEMT INQUIRE TRAN(CS*)

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

```
TSS ADD(deptacid) SPI(TRANSACT)
TSS ADD(deptacid) OTRAN(CS)
TSS PERMIT(acidname) SPI(TRANSACT) ACC(INQUIRE)
TSS PERMIT(acidname) OTRAN(CS) ACC(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.

12.9.1.3 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 table below lists the CEMT command keywords and their SPI equivalents for the CEMT PERFORM action.

Table 12-5. SPI Keywords for CEMT PERFORM

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

Examples for securing CEMT PERFORM commands appear next.

CEMT PERFORM SHUTDOWN

Using the commands below, the user only has permission to execute CEMT PERFORM SHUTDOWN commands.

```
TSS ADD(deptacid) SPI(SHUTDOWN)

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

12.9.1.4 ADD and REMOVE Commands

You can secure CEMT ADD and REMOVE commands for VOLUMEs only. The following access levels are valid:

Table 12-6. CEMT ADD and REMOVE Commands

Command Keyword	SPI Keyword
VOLUME	SPI(VOLUME)

Examples for securing CEMT ADD and REMOVE commands appear below.

Using the commands below, the user only has permission to execute CEMT ADD and REMOVE commands for VOLUMEs only.

```
TSS ADD(deptacid) SPI(VOLUME)

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

TSS REMOVE(acidname) SPI(VOLUME)

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

12.9.2 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* is equivalent to the CA-Top Secret SPI resource.
- *argument* is the data element being examined or modified.

For example:

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

Follow these steps to secure EXEC CICS commands:

1. TSS ADD the SPI resource to a department or division ACID.

2. TSS PERMIT the SPI resource to the user ACID and include the appropriate access level.

For example:

```
TSS ADD(divacid) SPI(FILE)

TSS PER(acid) SPI(FILE) ACC(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.

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

Table 12-7. SPI Access Levels for EXEC CICS

EXEC CICS Command	**SPI Access Level**
SET	SET

EXEC CICS INQUIRE and SET commands have related *options.* The next section describes how CA-Top Secret secures each option and its associated command.

12.9.2.1 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.

Table 12-8 (Page 1 of 2). SPI Keywords for EXEC CICS INQUIRE and SET Options

Command Option	**SPI Keyword**
AUTINSTMODEL	SPI(AUTINSTM)
AUTOINSTALL	SPI(AUTOINST)
CONNECTION	SPI(CONNECTI)
DELETESHIPPED	SPI(DELETESH)
DSNAME	SPI(DSNAME)
DUMPDS	SPI(DUMPDS)
EXITPROGRAM	SPI(EXITPROG)
FILE	SPI(FILE)
IRC	SPI(IRC)

Table 12-8 (Page 2 of 2). SPI Keywords for EXEC CICS INQUIRE and SET Options	
Command Option	**SPI Keyword**
JOURNALNUM	SPI(JOURNAL)
MODENAME	SPI(MODENAME)
MONITOR	SPI(MONITOR)
NETNAME	SPI(TERMINAL)
PARTNER	SPI(PARTNER)
PROFILE	SPI(PROFILE)
PROGRAM	SPI(PROGRAM)
REQID	SPI(REQID)
STATISTICS	SPI(STATISTI)
STORAGE	SPI(STORAGE)
SYSDUMPCODE	SPI(SYSDUMPC)
SYSTEM	SPI(SYSTEM)
TASK	SPI(TASK)
TCLASS	SPI(TCLASS)
TDQUEUE	SPI(TDQUEUE)
TERMINAL	SPI(TERMINAL)
TRACEDEST	SPI(TRACEDEST)
TRACEFLAG	SPI(TRACEFLAG)
TRACETYPE	SPI(TRACETYPE)
TRANCLASS	SPI(TCLASS)
TRANDUMPCODE	SPI(TRANDUMP)
TRANSACTION	SPI(TRANSACT)
TSQUEUE	SPI(TSQUEUE)
VOLUME	SPI(VOLUME)
VTAM	SPI(VTAM)

12.9.2.2 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.

Below is a table containing EXEC CICS keywords, the resource types called in the secondary CA-Top Secret security check, and the associated access levels.

Table 12-9. EXEC CICS Secondary Resource Checks

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

Examples for securing EXEC CICS INQUIRE and SET commands appear next.

EXEC CICS INQUIRE PROGRAM(TSSCAI)

Using the commands below, the user only has permission to execute EXEC CICS INQUIRE PROGRAM(TSSCAI) commands.

```
TSS ADD(deptacid) SPI(PROGRAM)

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

TSS ADD(deptacid) PPT(TSSCAI)

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

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

EXEC CICS SET TRANSACTION(TSS)

Using the commands below, the user only has permission to execute EXEC CICS SET TRANSACTION(TSS) commands.

```
TSS ADD(deptacid) SPI(TRANSACT)

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

TSS ADD(deptacid) OTRAN(TSS)

TSS PERMIT(acidname) OTRAN(TSS) ACC(INQUIRE)
```

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

12.9.2.3 ENABLE, DISABLE, and EXTRACT Commands

You can secure the ENABLE, DISABLE, and EXTRACT 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, and EXTRACT 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.

1. TSS PERMIT the user ACID and include the appropriate SPI resource access level.

The equivalent CA-Top Secret commands for the IBM EXEC CICS command shown above are:

```
TSS ADD(deptacid) SPI(ENABLE)

TSS PER(acid) SPI(ENABLE) ACC(SET)
```

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

Table 12-10. SPI Access Levels for EXEC CICS	
Command Function	**SPI Access Level**
ENABLE	SET
DISABLE	SET
EXTRACT	INQUIRE

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

12.9.2.4 Securing Functions

The table below lists the EXEC CICS command functions and their SPI equivalents for the EXEC CICS ENABLE, DISABLE, and EXTRACT commands.

Table 12-11. SPI Keywords for EXEC CICS ENABLE, DISABLE, and EXTRACT	
Command Function	**SPI Keyword**
ENABLE	SPI(EXITPROG)
DISABLE	SPI(EXITPROG)
EXTRACT	SPI(EXITPROG)

Examples for securing EXEC CICS ENABLE, DISABLE, and EXTRACT commands appear next.

EXEC CICS ENABLE

Using the commands below, the user only has permission to execute the EXEC CICS ENABLE commands.

```
TSS ADD(deptacid) SPI(EXITPROG)

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

EXEC CICS DISABLE

Using the commands below, the user only has permission to execute the EXEC CICS DISABLE commands.

```
TSS ADD(deptacid) SPI(EXITPROG)

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

EXEC CICS EXTRACT

Using the commands below, the user only has permission to execute the EXEC CICS EXTRACT commands.

```
TSS ADD(deptacid) SPI(EXITPROG)

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

12.9.2.5 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)
```

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

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

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

Table 12-12. SPI Keywords for EXEC CICS SPOOLOPEN

Command Function	**SPI Keyword**
SPOOLOPEN	SPI(PWRSPOOL)

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

Table 12-13. SPI Access Levels for EXEC CICS SPOOLOPEN	
Command Options	**SPI Access Level**
INPUT	SET
OUTPUT	SET

Examples for securing EXEC CICS SPOOLOPEN commands appear next.

EXEC CICS SPOOLOPEN INPUT

Using the commands below, the user only has permission to execute the EXEC CICS SPOOLOPEN INPUT commands.

```
TSS ADD(deptacid) SPI(PWRSPOOL)

TSS PERMIT(acidname) SPI(PWRSPOOL) ACC(SET)
```

EXEC CICS SPOOLOPEN OUTPUT

Using the commands below, the user only has permission to execute the EXEC CICS SPOOLOPEN INPUT commands.

```
TSS ADD(deptacid) SPI(PWRSPOOL)

TSS PERMIT(acidname) SPI(PWRSPOOL) ACC(SET)
```

12.9.2.6 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.

EXEC CICS SPOOLOPEN INPUT USERID(ext writer name)

Using the commands below, the user only has permission to execute the EXEC CICS SPOOLOPEN INPUT USERID commands.

```
TSS ADD(deptacid) ABSTRACT(ext writer name)

TSS PERMIT(acidname) ABSTRACT(ext writer name)
```

EXEC CICS SPOOLOPEN OUTPUT USERID(userid)

Using the commands below, the user only has permission to execute the EXEC CICS SPOOLOPEN OUTPUT USERID commands.

```
TSS ADD(deptacid) ABSTRACT(userid)

TSS PERMIT(acidname) ABSTRACT(userid)
```

12.9.2.7 QUERY SECURITY Command

The EXEC CICS QUERY SECURITY command and its functions are fully supported under CICS TS. Refer to the IBM *CICS 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.

12.10 Securing DL/I PSBs and DBDs

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.

12.11 Securing ICCF

This information only applies to running batch jobs in ICCF pseudo partitions.

CA-Top Secret protects VSE libraries, files, and program loads for jobs executing in ICCF pseudo partitions in the same way it supports any other VSE batch process.

Any jobs executing in an ICCF pseudo partition will execute under the facility name of the CICS region instead of the default BATCH facility.

To implement and support ICCF security checks, the RES facility option must be added for the facility in which ICCF resides:

```
TSS MODIFY(CICSPROD=RES)
```

Alternatively, it can be added permanently in the TSSPARM file.

12.12 Using Resource Caching

A resource cache is an in-storage list of secured resources that a user has been allowed to access. Its purpose is to reduce the overhead caused by repeated resource access validation.

CA-Top Secret/CICS typically determines whether access to a particular resource is allowed by submitting a resource validation call to the host CA-Top Secret subsystem. However, due to the nature of CICS, one terminal user may constantly repeat a set of transactions, resulting in multiple validation requests against the same resources. The resource cache reduces the number of host resource validation calls by capturing and saving "allowed" resources for each terminal in a separate storage buffer. Once a resource is stored in the resource cache, CA-Top Secret/CICS refers to the resource cache instead of issuing additional host resource validation calls.

12.12.1 Resource Cache Processing

Each terminal has its own resource cache buffer, which is created when a user signs on. When a resource validation is required, CA-Top Secret/CICS 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, 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/CICS 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 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/CICS assumes that access is allowed, but security processing (such as password verification) is still performed.
- 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.

12.12.2 Resource Cache Operation

Resource caching operates in one of two ways: transaction life or session life.

- **Transaction life** keeps resources in the cache buffer for the life of the transaction. Once the transaction ends, the cache buffer is cleared. Transaction life processing is the default for cache operation.
- **Session life** keeps the resources in the cache buffer for the life of the "signed on" session. The cache buffer is maintained until the user signs off.

There is a major performance benefit to using session life caching. Once the resource is stored in the cache buffer, subsequent accesses to that resource would avoid security host validation calls. Over time, all resource accesses would bypass the host validation path length, resulting in an overall performance gain.

How To Activate Session Life Caching: To activate session life caching, you must apply the optional APAR GS68190. However, there are a few security considerations you must be aware of if you are going to use session life caching:

- If auditing is in effect, only the first access for a particular resource will be logged to the Audit file. Once the resource is cached, no logging is performed.
- DATE and TIME permission will not be honored. Once the resource is cached, the user is allowed access to the resource for the life of the signed on session.

12.12.3 Tuning the Session Cache

The resource cache size is set to 512 bytes at initialization; currently there is no dynamic way to change the size. However, an optional APAR can be provided when a valid need exists to adjust the size of the cache. To help you tune the resource cache, CA-Top Secret/CICS provides functions to display cache utilization both 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 may 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.

The next sections explain how to display and interpret global and terminal cache status information.

12.12.3.1 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 = 768
Allocated session-related tokens = 2
Max concurrent sign-on/off requests = 50
Current no. of users signed on = 1
Active user-related sotrage = 1104 bytes
Total user-related storage = 61K bytes
Active user-related cache storage = 512 bytes
Total user-related cache sotrage = 384K bytes

CICS Maximum task value = 20
Current no. of tasks = 9
Active task-related storage = 36K bytes
Total task-related storage = 80K bytes
Total number task-related tokens = 20

Times resource found in cachce = 195
Times resource not found in cache = 28
Times cache overflowed = 3
Number entries added to cache = 28
Number entries excluded from cache = 0
Session cache box size = 512 bytes
```

The cache-related information contained on this screen is explained 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.

12.12.3.2 Displaying Terminal Cache Status

To display the cache information for a specific terminal, execute the following transaction from any CICS terminal:

```
TSEU=TERM=(termid | *)
```

Executing this transaction displays the information shown below.

```
ANALYSIS OF TERMINAL 2273

Address of security anchor is (0328F010)
The USER on this terminal is MASTER
The usr is defined and is running in WARN mode
The security record is located in ECSA

The CICS operator identifier is KOT
The CICS operator priority is 000
The attended bit is ON

-------------------------  User Session CACHE  -------------------------
Lookups   134       Hits   113          Inserts   21       Overflows   2
% Cache used   98.828
-----------------------------  CACHE Detail  ----------------------------

Resource     Resource      Number       Resource
 Class        Access        Hits          Name
PPT...Q%     READ             0        DFHEDAP
PPT...Q%     READ             0        DFHEITSP
PPT...Q%     READ             0        DFHEDAD
LCF...Q%     READ             2        CEMT
PPT...Q%     READ             2        DFHEMTP
```

```
PPT...Q%     READ             2        DFHEITMT
PPT...Q%     READ             2        DFHEMTD
SPI...s%     INQUIRE          1        FILE
FCT...F%     INQUIRE          1        CAIOPT
FCT...F%     INQUIRE          1        DFHCSD
SPI...s%     INQUIRE          1        CONNECTION
LCF....%     READ             1        ANSS
PPT...Q%     READ             1        PGMINSS
PPT...Q%     READ             0        TSSCAI
PPT...Q%     READ             0        TSSCAIN
LCF....%     READ             1        TSEU
PPT...Q%     READ             1        CAKSCHEK
PPT...Q%     READ             1        CAKSTERM
PPT...Q%     READ           101        CAKSWRIT
```

The cache related information displayed in the previous screens is explained next.

Global Terminal Cache Utilization

Lookups	Number of times cache was scanned.
Hits	Number of times a cache scan was successful.
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 13. Programmable Interfaces

This chapter discusses the Application Interface and the CA-Top Secret CICS exits.

13.1 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.

13.1.1 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 CICS LIBDEF search library, and must be defined to the CSD. The Application Interface can be invoked by either Command-Level 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 the section "Coding Samples", later in this chapter. Samples of PL/I, Assembler, and COBOL coding are located on the CA-Top Secret distribution tape, under File 11, TSSOPMAT.

13.1.2 Writing Requirements

Follow these guidelines when writing the CICS Application Interface:

- Use the CA-Top Secret Application Interface via an EXEC CICS 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: You must add a CSD entry for module 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(USER)
       RESIDENT(NO) USAGE(NORMAL) USELPACOPY(NO)
       STATUS(ENABLED) CEDF(NO) DATALOCATION(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) DATALOCATION(BELOW)
```

- You must pass the Application Interface a parameter list. The parameter list is passed by either 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 3.0 370 or 1138 bytes (The 1138 value refers to the FACLIST, RESLIST, and FLDXTR calls; all other calls use 370 bytes.)

13.1.3 Installation-defined Resources

Installation-defined resource classes (such as FIELD, UR1, UR2, and ABSTRACT) that are also predefined 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, refer to the TSS ADD(RDT) command function in the *Command Functions Guide.*

13.1.4 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. Refer to Chapter 12, "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.

13.1.5 Coding Samples

Use the coding examples provided in this section as customization samples.

13.1.5.1 Test TSSCAI Using Temporary Storage Record

```
      TITLE 'TESTCAI 1' --- TSS CICS APPLIATION INTERFACE
*************************************************************
*  NAME     - TESTCAI1                                      *
*  FUNCTION - COMMAND LEVEL ASSEMBLER CODE ......           *
*             TEST TSSCAI USING TEMPORARY STORAGE RECORD.   *
*  CALLS    - THE CICS APPLICATION INTERFACE PROGRAM        *
*************************************************************
           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   CL1138      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'TCPLV4L4'
        MVC   TSSCLASS,=CL8'FIELD   '
        MVC   TSSRNAME,=CL8'TSSFIELD'
        MVC   TSSPPGM,=CL8'        '
        XC    TSSACC,TSSACC
```

Figure 13-1. Customization Sample

```
*
* WRITE THE REQUEST RECORD TO TEMPORARY STORAGE.
*
        EXEC  CICS WRITEQ TS QUEUE(TSSQID)
              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)
              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'+1138'
         #TSSCPL               CICS PARAMETER LIST
         END
```

13.1.5.2 Test TSSCAI Using CICS COMMAREA

```
         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      CL1138               PARAMETER LIST LENGTH
         DS      CL1138               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'TCPLV4L4'
         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
```

Figure 13-2. Customization Sample

13.2 CA-Top Secret CICS Exits

CA-Top Secret provides two user exits: TSSPGM01 and TSSPGM02.

- TSSPGM01 is a message exit that lets you suppress or change the text of messages.
- TSSPGM02 is a message exit that lets you suppress or change the text of the locktime prompt.

The following sections explain how to modify these exits.

13.2.1 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    OH           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 will not be 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).

13.2.2 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:

```
WPARMLIST DS    0H           PROGRAM PARAMETER LIST
WMGAREA   DS    0XL79        MESSAGE AREA
WMSGLN    DS    H            MESSAGE LENGTH
          DS    H
WMESSAGE  DS    XL75         MESSAGE AREA
WPPWAREA  DS    0XL8         PASSWORD AREA
WPSWD     DS    XL8          PASSWORD
WPLISTLN  EQU   *-WPARMLST   PROGRAM P-LIST LENGTH
$TABEND   EQU   X'FF'        ABEND TASK
WMSGALN   EQU   *-WPARMLST   PARAMETER LIST LENGTH
```

Note: If running CICS/ESA, this exit must run AMODE(31).

WMGAREA Contains the password prompt message.

WPPWAREA Is the field that the TSSPGM02 Exit program places the user's password in for reverification.

13.2.3 Sample Program Definitions

Program definitions for the TSSPGM01 and TSSPGM02 message exits should look like the example shown below.

```
DEFINE  PROGNAM(TSSPGM01) GROUP(TOPGRP)
        DESCRIPTION(CA-SECURITY/CICS USER EXIT 01)
        LANGUAGE(ASSEMBLER) EXECKEY(CICS)
        RESIDENT(NO) USAGE(NORM) USELPACOPY(NO)
        STATUS(ENABLED) CEDF(NO) DATALOCATION (ANY)
```

Note: EXECKEY must be CICS, otherwise errors will result.

Chapter 14. CA-Top Secret Supplied Transactions

This chapter discusses the CA-Top Secret-supplied transactions and the CICS-supplied transactions.

14.1 LOCKTIME Logoff Feature Support (TSLA, TSLM)

The TSLA and TSLM transactions and their associated programs are required if you are specifying the LTLOGOFF FACILITY suboption.

14.1.1 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 or not 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.

14.1.2 TSLM Transaction

TSLM is a CA-Top Secret CICS transaction used with the TSLA transaction (described previously) to support LOCKTIME processing.

14.2 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.
TSEU=TRANS=(trans)	Gives information about a specified CICS transaction, where *trans* is the four-character CICS transaction ID specified in the PCT.

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 or not a user is signed on.
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.

14.2.1 Executing TSEU

This section contains a detailed description of the information returned by each of the transactions.

14.2.1.1 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.

Sample screens that appear when you issue TSEU=INSTALL are shown next.

```
SIT parameters in use, as defined in the FACILITY:
     SEC=YES       XAPPC=NO      XUSER=NO
     XCMD=NO       XDCT=NO       XFCT=CICSFCT  XJCT=NO       XPSB=NO
     XPCT=NO       XPPT=NO       XTST=NO       XTRAN=CICSTRN
     PCTCMDSEC=HONOR             PCTRESSEC=OVERRIDE

TSS application interface is installed
TSS command interface is NOT installed
This region is using CICS41T as a control ACID
The control ACID is defined with MASTFAC(CICSPROD)

DSNCHECK=YES    LTLOGOFF=NO

    Module location of CAKSLMT   is (83713418)
    Module location of CAKSMSGH  is (837BF408)
    Module location of CAKSPCH1  is (837D6F80)
    Module location of CAKSSIGN  is (837D8710)
    Module location of CADSSERV  is (837B25C8)
    Module location of CAKSRINT  is (837D6200)
    Module location of CAKSPVAL  is (837D80C8)
    Module location of CAKSRVAL  is (837DAC90)
    Module location of CAKSGLUE  is (837DA148)
```

```
    Module location of CAKSPWH   is (837B8128)
    Module location of CAKSCMIN  is (837C1920)
    Module location of CAKSHASH  is (836F2138)
    Module location of CAKSALOC  is (8371E0F0)
    Module location of CAKSTRPX  is (837C1020)
    Module location of CAKSGRSA  is (836F3D80)
    Module location of CAKSXCMD  is (837DC6D0)
    Module location of CAKSATS   is (837DD928)
    Module location of TSSCLMT   is (83713418)
    Module location of TSSCRVAL  is (837FDC88)
    Module location of TSSCCMDP  is (837FD0A0)
    Module location of TSSCRTYP  is (836F10F8)
    Module location of TSSCCTYP  is (837B20E0)
    Module location of TSSCRACC  is (836F10B8)
    Module location of TSSCTRKN  is (84C85980)
    Module location of CAKSCMGR  is (837B6D30)
    Module location of CAKSMSEV  is (8371EF40)
    Module location of CAKSPINT  is (837B1000)
    Module location of CAKSPMGR  is (837D1158)
    Module location of CAKSROUT  is (84C878C8)
    Module location of CAKSRD41  is (8372FF30)
    Module location of CAKSSMGR  is (837BFEA0)
```

14.2.1.2 TSEU=WHOSON

Indicates who is signed on to a particular region.

A sample screen appears below.

```
               <<<    List of USERS in Region: A15ICT03  >>>
                                                Signon    Signon     Signon
Userid      Netname     Applid     Mode     Type      Time       Date
-------------------------------------------------------------------------
CAKSDUSR    A51L90A     A15ICT03   Warn     User      10:57      12-21-94
CAKSDUSR    A15L904     A15ICT03   Warn     User      10:58      12-21-94
KOTPA01     A15L905     A15ICT03   Fail     User      11:18      12-21-94
MASTER      K18L2273    A15ICT03   Fail     User      11:13      12-21-94
```

14.2.1.3 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.

A sample screen for TSEU=TRAN=CEMT appears next.

```
ANALYSIS OF TRANSACTION CEMT
This is a LOCAL transaction
The transaction priority is 255
The transaction is defined with CMDSEC=YES and RESSEC=YES
Your region is defined with PCTCMDSEC=HONOR and PCTRESSEC=HONOR
The TRANID is NOT defined in the security bypass list
```

14.2.1.4 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.

14.2.1.5 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.

A sample screen for TSEU=TERM=2273 appears next.

```
ANALYSIS OF TERMINAL 2273

Address of security anchor is (0328F010)
The USER on this terminal is MASTER
The usr is defined and is running in WARN mode
The security record is located in ECSA

The CICS operator identifier is KOT
The CICS operator priority is 000
The attended bit is ON

-------------------------  User Session CACHE  -------------------------
Lookups   134       Hits   113          Inserts   21       Overflows   2
% Cache use   98.828
-----------------------------  CACHE Detail  -----------------------------

Resource     Resource      Number       Resource
 Class        Access        Hits          Name
PPT...Q%     READ             0        DFHEDAP
PPT...Q%     READ             0        DFHEITSP
PPT...Q%     READ             0        DFHEDAD
LCF...Q%     READ             2        CEMT
PPT...Q%     READ             2        DFHEMTP
```

```
PPT...Q%     READ             2        DFHEITMT
PPT...Q%     READ             2        DFHEMTD
SPI...s%     INQUIRE          1        FILE
FCT...F%     INQUIRE          1        CAIOPT
FCT...F%     INQUIRE          1        DFHCSD
SPI...s%     INQUIRE          1        CONNECTION
LCF....%     READ             1        ANSS
PPT...Q%     READ             1        PGMINSS
PPT...Q%     READ             0        TSSCAI
PPT...Q%     READ             0        TSSCAIN
LCF....%     READ             1        TSEU
PPT...Q%     READ             1        CAKSCHEK
PPT...Q%     READ             1        CAKSTERM
PPT...Q%     READ           101        CAKSWRIT
```

14.2.1.6 TSEU=MAXT=INQ

Has the ability to *inquire* about the maximum number and actions available for concurrent signon/signoff requests that are set.

Note: The SET option is not used with CA-Top Secret Release 4.4. Refer to the MAXSIGN FACILITY suboption in the *Control Options Guide* for details.

A sample screen for TSEU=MAXT=INQ appears next.

```
Maximum users = 3000
Total number session-related tokens = 768
Allocated session-related tokens = 2
Max concurrent sign-on/off requests = 50
Current no. of users signed on = 1
Active user-related sotrage = 1104 bytes
Total user-related storage = 61K bytes
Active user-related cache storage = 512 bytes
Total user-related cache sotrage = 384K bytes

CICS Maximum task value = 20
Current no. of tasks = 9
Active task-related storage = 36K bytes
Total task-related storage = 80K bytes
Total number task-related tokens = 20

Times resource found in cachce = 195
Times resource not found in cache = 28
Times cache overflowed = 3
Number entries added to cache = 28
Number entries excluded from cache = 0
Session cache box size = 512 bytes
```

14.2.1.7 TSEU=NEWC=(program)

Refreshes the running copy of a CA-Top Secret CICS module, allowing emergency maintenance to be applied to a single CICS region without recycling that specific region.

Note: The following TSS CICS modules are **not refreshable:** CAKSCMGR, CAKSSMGR, CAKSPMGR, CAKSLOCK, CAKSPINT, CAKSRINT, CAKSROUT, CAKSCINT, and CAKSSCAN.

The following CA-Top Secret CICS modules are refreshed through the CICS command CEMT SET PROGRAM(*name*) NEWCOPY: CAKSCHEK, CAKSINST, CAKSMAXT, CAKSNEWC, CAKSTERM, CAKSTRAC, CAKSTRAN, CAKSWRIT, and CAKSWHOS. However, any TSS or CAKS program with a CSD definition can be refreshed using CEMT NEWCOPY.

A sample screen for TSEU=NEWC=CAKSRVAL appears next.

```
Module refresh successful   - CAKSRVAL.
```

14.2.1.8 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.

Keyword	Description
LEvel=1|2|3	Controls the amount of trace data created. Select one of these levels: **1** 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. Level 1 is the default. **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 15. CICS Installation Checklist

Use this checklist when you are installing and implementing CA-Top Secret security using CICS TS 1.1 and above.

1. CAIENF Considerations

   The CA-Top Secret CICS interface requires the CA-CIS CAIENF (Event Notification Facility) 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:

   a. Check the LIBDEF to see if it contains the CA-Top Secret library. If it does not, add the library to the LIBDEF in the CICS JCL to properly load CAKSCINT. If CAKSCINT is not loaded, CA-Top Secret will not install into the CICS regions and you will not receive a Phase I message.

   b. Check the LIBDEF to see if it contains the CAIENF library. If it does not, add the library to the LIBDEF in the CICS JCL to properly load CAKSCINT. If CAKSCINT is not loaded, CAIENF will not install into the CICS regions and you will not receive a Phase I message.

   c. Ensure that EXTSEC is specified in either the CICS SIT or the Facility Matrix Table if the FACMATRX option has been set to YES.

   d. Ensure that one of the XPARMS (XFCT, XJCT, XPCT, etc) is set to YES.

   Refer to the *CA-CIS Installation Guide* for detailed information.

2. System Initialization Table (SIT)

   You must specify SEC=YES in either 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.

3. 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: You must make updates to 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 LIBDEF search library.

Refer to Chapter 16, "CSD TRANSACTION and PROGRAM Entries" for the updated definition statements.

4. TSSTRACK Utility

   Allocate the Audit Tracking file to the CICS region.

5. ISC/MRO Considerations

   Review Chapter 11, "Security for a Multi-System Environment" prior to implementing security under an MRO and/or ISC environment.

6. Starting Your CICS Region

   a. Verify that your CICS region ACIDs have either the NORESCHK or NOLCFCHK attributes or have been PERMITted to the appropriate transactions.

   b. At this point CA-Top Secret and CAIENF are installed and active.

   c. The following messages are displayed in succession. Refer to the *Messages and Codes Guide* for complete descriptions of reasons and actions associated with each message.

      Install Check Messages

```
TSS6093I - TSS/CICS Initialization Phase 0 started.
TSS6094I - TSS/CICS Initialization Phase 0 complete.
TSS6000I - TSS/CICS Initialization Phase 1 started.
TSS6099I - TSS/CICS Initialization Phase 1 complete.
TSS6002I - TSS/CICS Initialization Phase 2 started.
TSS6095I - TSS/CICS Signon Manager Subtask is active.
TSS6096I - TSS/CICS Attaching 005 Signon Server.
TSS6088I - TSS/CICS Core Manager Subtask is active.
TSS6088I - TSS/CICS Core Manager Subtask is active.
TSS6089I - TSS/CICS Program Manager Subtask is active.
TSS6003I - TSS/CICS Initialization Phase 2 complete.
TSS6007I - TSS/CICS Security Activated.
DFHXS0206 - CA-ENF Installing the CICS interface.
```

   d. Once the **TSS6007I Security Activated** message is displayed, the CA-Top Secret CICS interface is installed and active in the region.

7. 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.

Chapter 16. CSD PROGRAM and TRANSACTION Sample Entries

All PROGRAM and TRANSACTION sample entries are provided in the product install library as members of type 'Z'. The entries are contained in member TSSCSD41.Z, these entries can be used to update your CSD dataset.

16.1 Sample CSD 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.

```
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) PROTECT(NO)
         CHAINCONTROL(NO) DVSUPRT(ALL)
         INBFMH(EODS) RAQ(NO) LOGREC(NO)
         NEPCLASS(0) RTIMOUT(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)
         INDOUBT(BACKOUT) 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) USESVACOPY(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) USESVACOPY(NO)
         STATUS(ENABLED) CEDF(NO) DATALOCATION(BELOW)
DEFINE PROGRAM(TSSCAIO) GROUP(TOPSGRP)
         DESCRIPTION(CA-TOP SECRET Application Interface)
         LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(CICS)
         RESIDENT(NO) USAGE(NORMAL) USESVACOPY(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) USESVACOPY(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) USESVACOPY(NO)
         STATUS(ENABLED) CEDF(NO) DATALOCATION(BELOW)
DEFINE PROGRAM(TSSCICSO) GROUP(TOPSGRP)
         DESCRIPTION(CA-TOP SECRET Administration Interface)
         LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(CICS)
         RESIDENT(NO) USAGE(NORMAL) USESVACOPY(NO)
         STATUS(ENABLED) CEDF(NO) DATALOCATION(BELOW)
```

16.2 Sample CSD 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) PROTECT(NO)
         CHAINCONTROL(NO) DVSUPRT(ALL)
         INBFMH(EODS) RAQ(NO) LOGREC(NO)
         NEPCLASS(0) RTIMOUT(NO)
```

The following example shows how to define 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)
         INDOUBT(BACKOUT) 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)
         INDOUBT(BACKOUT) 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)
         INDOUBT(BACKOUT) RESTART(NO) SPURGE(NO)
         TASKDATALOC(ANY) TASKDATAKEY(USER)
         TPURGE(NO) DUMP(YES) TRACE(YES)
         RESSEC(NO) CMDSEC(NO)
```

The following example shows how to define PROGRAM entries for the CICS component.

```
DEFINE PROGRAM(CAKSCHEK) GROUP(CAKSGRP)
         DESCRIPTION(CA-SECURITY/CICS Environment Utility Driver)
         LANGUAGE(ASSEMBLER) RELOAD(NO) EXECKEY(CICS)
         RESIDENT(NO) USAGE(NORMAL) USESVACOPY(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) USESVACOPY(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) USESVACOPY(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) USESVACOPY(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) USESVACOPY(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) USESVACOPY(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) USESVACOPY(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) USESVACOPY(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) USESVACOPY(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) USESVACOPY(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) USESVACOPY(NO)
         STATUS(ENABLED) CEDF(NO) DATALOCATION(ANY)
```

Figure 16-1. Sample Program Entries for the CICS Component

Index

E

F

I

J

L

M

N

P

R

S

T

U

V

X

User Registration Form

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

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

Registration entitles you to receive such items as:

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

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

Product(s): _____

_____

_____

_____

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

Name: _____

Position: _____

Company or organization: _____

Address: _____

Address: _____

Phone No.: _____

Date: _____

I would like additional information on: _____

Demand Analysis Request Form

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

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

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

Name: _____

Position: _____

Company or organization: _____

Address: _____

Address: _____

Phone No.: _____

Date: _____

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

Priority: _____

Description of Requested Item:

_____

_____

_____

_____

_____

_____

_____

_____

_____

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

Reader Comment Form

CA-Top Secret Implementation: CICS Guide

3.0 VSE

Document Number: R101TS30ICE

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

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

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

Name: _____

Position: _____

Company or organization: _____

Address: _____

Address: _____

Phone No.: _____

Date: _____

Years of experience with this CA product: _____

Overall Rating

	Good	Fair	Poor	Why?
Accuracy				
Organization				
Clarity				

Reference Aids

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

Clarity

	Good	Fair	Poor	Why?
Instructions and Procedures				
Examples				
Graphics				

How Manual Is Used:

How do you use this manual in your job?

How often do you use this manual in a week?

Suggestions:

What do you like about this manual?

What do you dislike about this manual?

What would you like us to change?

Additional Comments: