INCONTROL for z/OS
9.0.00
Security Guide

July 2015

Contacting BMC Software

You can access the BMC Software website at http://www.bmc.com. From this website, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

United States and Canada

Address

BMC SOFTWARE INC
2101 CITYWEST BLVD
HOUSTON TX
77042-2827
USA

Telephone

- **713 918 8800**
- **800 841 2031**

Fax **713 918 8000**

Outside United States and Canada

Telephone **(01) 713 918 8800** **Fax** **(01) 713 918 8000**

Restricted rights legend

Customer support

You can obtain technical support by using the BMC Software Customer Support website or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, see “Before contacting BMC.”

Support website

You can obtain technical support from BMC 24 hours a day, 7 days a week at http://www.bmc.com/support. From this website, you can:

- Read overviews about support services and programs that BMC offers
- Find the most current information about BMC products
- Search a database for issues similar to yours and possible solutions
- Order or download product documentation
- Download products and maintenance
- Report an issue or ask a question
- Subscribe to receive proactive e-mail alerts when new product notices are released
- Find worldwide BMC support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

Support by telephone or e-mail

In the United States and Canada, if you need technical support and do not have access to the web, call 800 537 1813 or send an e-mail message to customer_support@bmc.com. (In the subject line, enter **SupID:<yourSupportContractID>**, such as SupID:12345). Outside the United States and Canada, contact your local support center for assistance.

Before contacting BMC

Have the following information available so that Customer Support can begin working on your issue immediately:

- Product information
  - Product name
  - Product version (release number)
  - License number and password (trial or permanent)
- Operating system and environment information
  - Machine type
  - Operating system type, version, and service pack or other maintenance level such as PUT or PTF

- System hardware configuration
  - Serial numbers
  - Related software (database, application, and communication) including type, version, and service pack or maintenance level
- Sequence of events leading to the issue
- Commands and options that you used
- Messages received (and the time and date that you received them)
  - Product error messages
  - Messages from the operating system, such as `file system full`
  - Messages from related software

License key and password information

If you have questions about your license key or password, contact BMC as follows:

- (USA or Canada) Contact the Order Services Password Team at 800 841 2031, or send an e-mail message to ContractsPasswordAdministration@bmc.com.
- (Europe, the Middle East, and Africa) Fax your questions to EMEA Contracts Administration at +31 20 354 8702, or send an e-mail message to password@bmc.com.
- (Asia-Pacific) Contact your BMC sales representative or your local BMC office.

Third party Software

For the provisions described in the BMC License Agreement and Order related to third party products or technologies included in the BMC Product, see https://docs.bmc.com/docs/display/workloadautomation/Control-M+Workload+Automation+Documentation and click **Third-party software (TPS).**

Contents

About this book

Conventions Used in This Guide

Notational conventions that may be used in this guide are explained below.

Standard Keyboard Keys

Keys that appear on the standard keyboard are identified in boldface, for example, Enter, Shift, Ctrl+S (a key combination), or Ctrl S (a key sequence).

The commands, instructions, procedures, and syntax illustrated in this guide presume that the keyboards at your site are mapped in accordance with the EBCDIC character set. Certain special characters are referred to in this documentation, and you must ensure that your keyboard enables you to generate accurate EBCDIC hex codes. This is particularly true on keyboards that have been adapted to show local or national symbols. You should verify that

$ is mapped to x'5B'
# is mapped to x'7B'
@ is mapped to x'7C'

If you have any questions about whether your keyboard is properly mapped, contact your system administrator.

Preconfigured PFKeys

Many commands are preconfigured to specific keys or key combinations. This is particularly true with regard to numbered PF keys, or pairs of numbered PFKeys. For example, the END command is preconfigured to, and indicated as, PF03/PF15. To execute the END command, press either the PF03 key or the PF15 key.

Instructions to enter commands may include

- only the name of the command, such as, enter the END command
- only the PF keys, such as, press PF03/PF15
- or both, such as, press PF03/PF15, or enter the END command

Command Lines and Option Fields

Most screens contain a command line, which is primarily used to identify a single field where commands, or options, or both, are to be entered. These fields are usually designated COMMAND, but they are occasionally identified as COMMAND/OPT or COMMAND/OPTION.

Option field headings appear in many screens. These headings sometimes appear in the screen examples as OPTION, or OPT, or O.

Names of Commands, Fields, Files, Functions, Jobs, Libraries, Members, Missions, Options, Parameters, Reports, Subparameters, and Users

The names of commands, fields, functions, jobs, libraries, members, missions, options, parameters, reports, subparameters, users, and most files, are shown in standard UPPERCASE font.

User Entries

In situations where you are instructed to enter characters using the keyboard, the specific characters to be entered are shown in this UPPERCASE BOLD text, for example, type EXITNAME.

Syntax Statements

In syntax, the following additional conventions apply:

- A vertical bar (|) separating items indicates that you must choose one item. In the following example, you would choose a, b, or c:

  a| b | c

- An ellipsis (. . .) indicates that you can repeat the preceding item or items as many times as necessary.
- Square brackets ([ ]) around an item indicate that the item is optional. If square brackets ([ ]) are around a group of items, this indicates that the item is optional, and you may choose to implement any single item in the group. Square brackets can open ([ ) and close ( ]) on the same line of text, or may begin on one line of text and end, with the choices being stacked, one or more lines later.
- Braces ({ }) around a group of items indicates that the item is mandatory, and you must choose to implement a single item in the group. Braces can open ({) and close (}) on the same line of text, or may begin on one line of text and end, with the choices being stacked, one or more lines later.

Screen Characters

All syntax, operating system terms, and literal examples are presented in this typeface. This includes JCL calls, code examples, control statements, and system messages. Examples of this are:

- calls, such as
- CALL ’CBLTDLI’
- code examples, such as
- FOR TABLE owner.name USE option, . . . ;
- control statements, such as
- //PRDSYSIN DD * USERLOAD PRD(2) PRINT
- system messages, both stand-alone, such as You are not logged on to database database_name, and those embedded in text, such as the message You are not logged on to database database_name, are displayed on the screen.

Variables

Variables are identified with italic text. Examples of this are:

- In syntax or message text, such as
  Specify database database_name
- In regular text, such as
  replace database database_name1 with database database_name2 for the current session
- In a version number, such as
  EXTENDED BUFFER MANAGER for IMS 4.1.xx

Special Elements

This book includes special elements called notes and warnings:

Notes provide additional information about the current subject.

Warnings alert you to situations that can cause problems, such as loss of data, if you do not follow instructions carefully.

Information New to This Version

Additional information that is new to this version is described in the *INCONTROL for z/OS Installation Guide: Upgrading* and What's New section of the INCONTROL for z/OS Release Notes.

Related Publications

INCONTROL for z/OS Installation Guide

A step-by-step guide to installing, customizing, maintaining, and upgrading INCONTROL products using the INCONTROL Customization and Installation Engine (ICE) application.

The guide is divided into the following volumes:

- *INCONTROL for z/OS Installation Guide: Installing*

  This guide describes the procedures and steps required for installing INCONTROL products.

  The following installation tracks are available:

  - express installation
  - customized installation
  - cloning

- *INCONTROL for z/OS Installation Guide: Customizing*

  This guide describes the procedures and steps required for customizing INCONTROL products.

- *INCONTROL for z/OS Installation Guide: Maintaining*

  This guide describes the procedures and steps required for maintaining INCONTRO products.

  The following maintenance procedures are described:

  - periodic maintenance
  - ad hoc maintenance

- change deployment

- *INCONTROL for z/OS Installation Guide: Upgrading*

  This guide contains instructions for upgrading to the current release from previous INCONTROL versions.

  The Express Upgrade is the recommended approach for upgrading INCONTROL products to the most recent major version.

INCONTROL for z/OS Messages Manual

A comprehensive listing and explanation of all IOA and INCONTROL messages.

INCONTROL for z/OS Administrator Guide

This guide provides information for system administrators about customizing and maintaining INCONTROL products.

INCONTROL for z/OS Utilities Guide

A detailed description of the utilities designed to perform specific administrative tasks that are available to INCONTROL products. The guide contains an alphabetized reference of all utilities for each INCONTROL product.

User Guides

Product-specific guides containing comprehensive information about the operation and implementation of each INCONTROL product.

IOA Security

The INCONTROL family of products can be protected like any other standard facility in a data center. INCONTROL has built-in security interfaces to RACF, CA-ACF2 and CA-TopSecret. In most cases, you are not required to customize the security modules.

This chapter describes the procedure used to install the security interface for the IOA component.

- **Security Modules:** Each IOA function invokes a security module that verifies if a specified user is authorized to perform a specific function (such as, add, modify, and delete) and determines if the action is permitted or denied. Security modules are used to control access to the various protected elements.
- **User Exits:** User exits are invoked before the security modules to allow the user to perform any required user functions that are not related to security. However, the user exits can be customized so that both user and security functions are performed by either or both modules. It is recommended that you separate the functions because some user exits cannot perform security functions. For more information, refer to the descriptions of each security module later in this guide.
- **IOASECUR Module:** When any interaction with the security product is required, the common security services module, IOASECUR, is invoked. This module is invoked each time an INCONTROL product requires a security service. For example, to create the security environment, check a user’s authority, extract user information, or delete a security environment.

  The IOASECUR module determines which security product the site is using (RACF, TopSecret, or ACF2), and invokes the relevant IOA security interface module (IOARACF, IOATSS, or IOAACF2).

Defining Security Modes

Security requirements often differ from site to site. The IOA security interface provides two modes for defining security: Basic Definition mode and Extended Definition mode. The difference between the two modes is not the degree of security, but rather ease of implementation (using Basic Definition mode) compared with the level of authorization that can be granted to the user (using Extended Definition mode).

The administrator can choose to use either Basic Definition mode or Extended Definition mode to control user authorizations to each INCONTROL security module.

Basic Definition Mode

Basic Definition mode authorizes the user access to an INCONTROL protected element. A user who is granted permission to an element is authorized to perform all the actions that are valid for this element (such as add, delete, change and update). The advantage of working in this mode is that it merges several different security events into a logically grouped resource structure. This structure simplifies the administration required to implement INCONTROL security.

Extended Definition Mode

Extended Definition mode grants each user access for a specific action within each INCONTROL protected element. Therefore, a user who is granted access to an INCONTROL element can be granted or denied any action (add, delete, change and update) within that element. This definition mode requires you to define several access rules. For each action there is an associated resource structure. However, Extended Definition mode provides maximum flexibility and accuracy for granting authorizations.

Conditional Definition Mode

When working in Conditional Definition mode (Fixed vs. Conditional Definition Mode (on page 12)), the security module first determines the definition mode to be used. To determine the mode, the security module issues an authority check for access to the entity $$IOAEDM.qname (or $$CTxEDM.qname) in the FACILITY class. If the user is authorized for the given entity, the security module operates in Extended Definition mode. Otherwise, the security module operates in Basic Definition mode.

During the installation process, a security module can be customized to operate in a single mode (meaning, either Extended Definition mode or Basic Definition mode). For more information, see Implementing IOA Security (on page 24).

Fixed vs. Conditional Definition Mode

Fixed Definition mode enables the administrator to determine the definition mode in advance. Conditional Definition mode first establishes the user’s definition mode (Basic or Extended) and then determines a user’s authority to access an entity. Conditional Definition mode provides more flexibility than Fixed Definition mode when determining a user’s authority. For example, certain users can be set to work in one mode, while others can be set to work in another mode. Using Fixed Definition mode improves the performance of the security checking process because the check for the mode is eliminated.

When Conditional Definition mode is used and a resource profile is defined to protect $$IOAEDM or $$CTxEDM entities, it is recommended that the security system not audit this resource to avoid generating failure audit records for these entities.

When Conditional Definition mode is used, one more call is made to the security product to determine the mode in which the user should operate.

Defining Security for Multiple Environments

The security definition scope is one IOA environment. If your site is running multiple IOA environments simultaneously (such as Production and Test), different security definitions can be specified for each environment. IOA installation parameter QNAME is used to distinguish between the IOA environments. For more information about this parameter, see the IOA installation chapter in the *INCONTROL for z/OS Installation Guide: Installing*.

Identifying Users

When planning INCONTROL security implementation, it is recommended that the security administrator identify the users or group of users for the test or production environment.

Typically, the IOA environment is intended for the following types of user categories:

- Users who install and customize INCONTROL products, and determine implementation and policy issues
- Users who create and update definitions of IOA entities, such as prerequisite conditions
- Users who use and operate INCONTROL products
- Users with special user IDs, such as long-running started tasks or specific system jobs. To determine the security related setup that these users require see Basic Definition Security Calls (on page 36), and Extended Definition Security Calls (on page 37).

Access to each protected element is controlled by verifying that the active user ID is authorized to perform each specific function. The specified user ID for authorization varies, depending on the environment being used. User IDs can be one of the following types:

Table 1 User IDs

Type	Description
Online User	User ID used to log in the IOA online environment
Execution User	User ID associated with a running job or a started task
Definition Owner	User ID specified in the owner field in the definition of the mission or job, and so on
JCL User	User ID associated with a batch job

When a security interface module performs a security authorization check, the user ID that is checked for the operation can be any of the user IDs described above, depending on the environment in that the operation is performed.

Implementation Overview

IOA security installation must be completed before proceeding with the security implementation of INCONTROL products.

It is recommended that the security administrator follow the implementation stages outlined below to verify that security of all INCONTROL products is successfully implemented:

- Plan the security implementation.

  Identify the types of users required for a test installation or production installation. Specify the types of security definitions required for each type of installation (for example, decide if Basic Definition mode is sufficient or if Extended Definition mode is required).

- Specify the required definitions and authorizations.

  Create the definitions required to permit the INCONTROL products installer to perform all actions.

- Specify security parameters using ICE.

  Follow the steps in ICE to specify the parameters required for the security implementation.

- Submit all security installation and customization jobs.

  Follow the steps in ICE to submit all the security installation jobs. Check all error messages produced by each job.

IOA Security Implementation

Each chapter in this guide contains a table that lists all the protected elements applicable to the INCONTROL product for that security is implemented. It is recommended that the security administrator review the table to determine the user categories and authorizations required for the protected elements in the INCONTROL products. Members (IOASRAC3 and CTxSRAC3, IOASTSS3 and CTxSTSS3, and IOASSAF3 and CTxSSAF3) in the IOA INSTWORK library contain examples that can be used as a basis for specifying the required authorizations. The job is created in the IOA INSTWORK library after selecting the "Functions Security Definitions" step in ICE.

The steps below describe the flow of the IOA security implementation process.

1. Determine the security environment for the user.

   IOA security interface modules verify that a specific user is authorized to perform a specific action. If the protected components belong to a multiuser address space (such as the Control-M monitor, the Control-D monitor, or the Online monitor) it is necessary to determine the security profile of the user within the address space. When an IOA security module is invoked under a multiuser address, the IOASECUR module determines the security profile. This service identifies the user requesting the action and builds the user’s security profile in storage. Otherwise, the user maintains the same security profile as that of the address space.

2. Determine definition mode.

   Once the security environment is determined for the user ID, the security interface determines which definition mode is required for the specified user ID or security module. The appropriate entity and CLASS are built for the user ID.

   If the chosen mode is conditional, a check for the Extended entity ($$xxxEDM) is performed and the appropriate entity and CLASS are built based on the check results. If the chosen mode is Fixed Definition mode, the appropriate entity and CLASS are built.

3. Issue security checks.

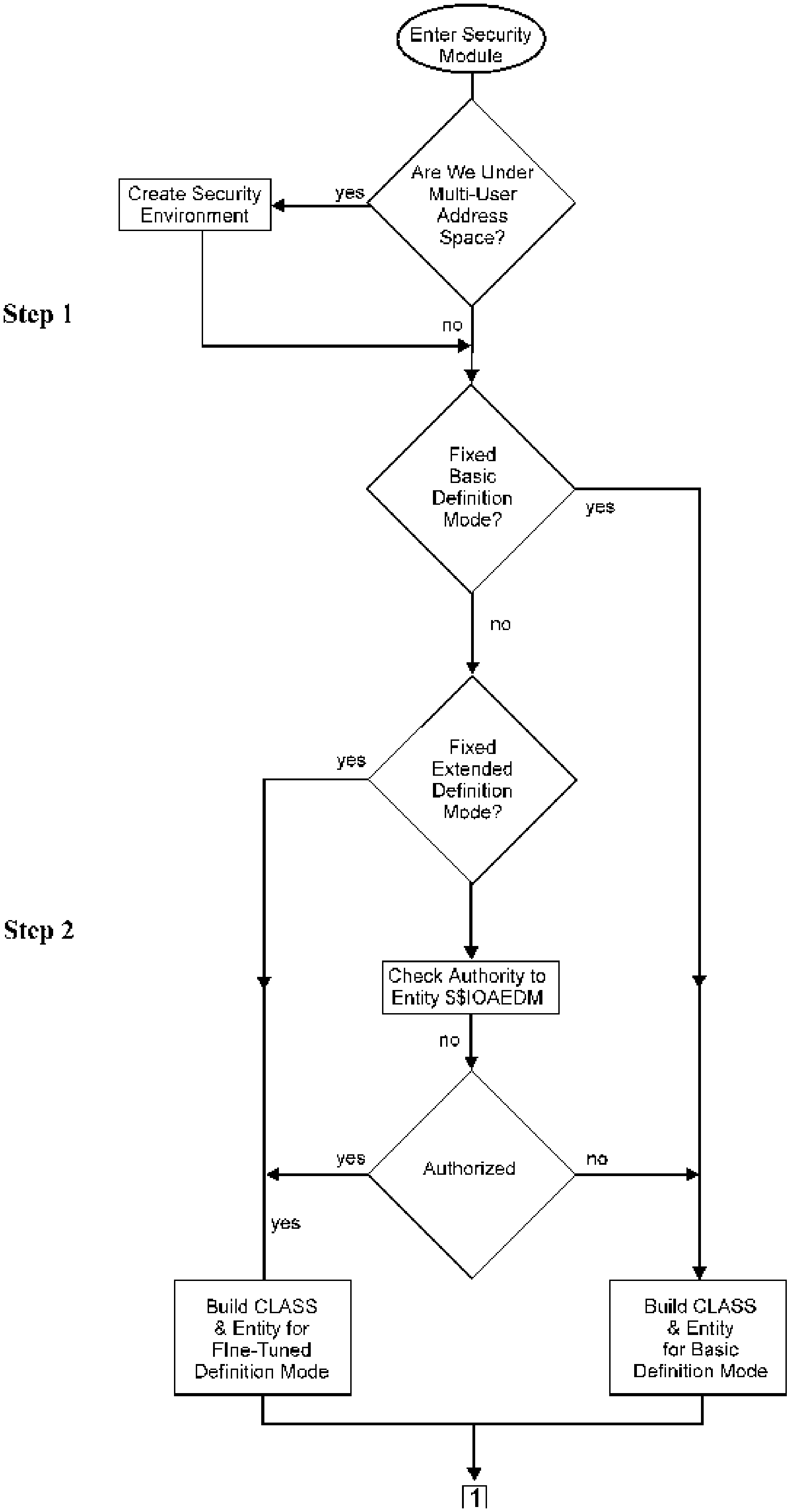
   The IOASECUR IOA security services module is invoked to perform the authority check.

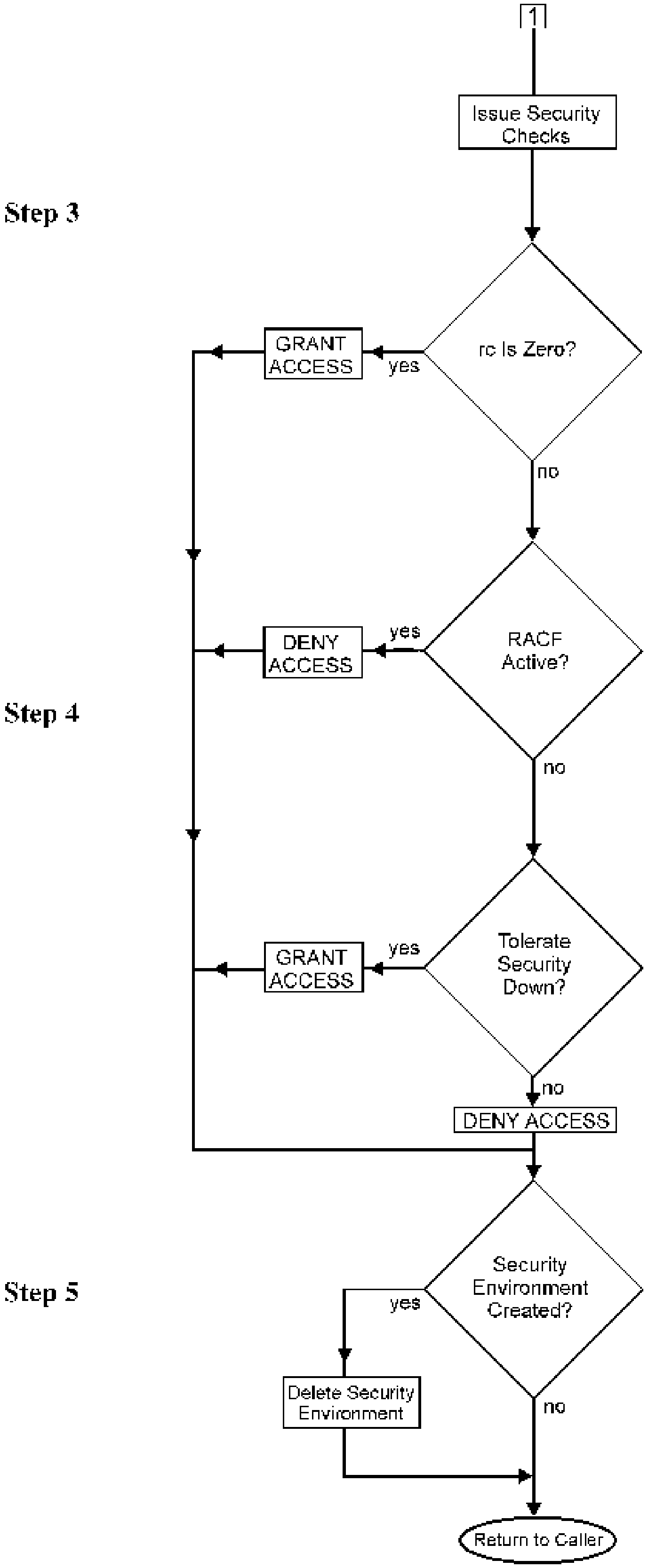
4. Determine security check results.

   If the user is authorized to the specified entity, an application return code zero (authorized) is returned. If the return code from the security product indicates that the security product is not active or the specific entity is not defined to the security product, the security interface module either grants or denies the request, according to the SECTOLx security installation parameter.

   Once the authorization process is complete, the security environment is cleaned up.

   The following diagrams illustrate the detailed flow of the INCONTROL security modules, representing each of the steps described above.


Enter Security Module
Are We Under Multi-User Address Space?
yes
Create Security Environment
Step 1
no
Fixed Basic Definition Mode?
yes
no
yes
Fixed Extended Definition Mode?
Step 2
Check Authority to Entity S$IOAEDM
no
yes
Authorized
no
yes
Build CLASS & Entity for FIne-Tuned Definition Mode
Build CLASS & Entity for Basic Definition Mode
1


1
Issue Security Checks
Step 3
rc Is Zero?
yes
GRANT ACCESS
no
Step 4
RACF Active?
yes
DENY ACCESS
no
Tolerate Security Down?
yes
GRANT ACCESS
no
DENY ACCESS
Step 5
Security Environment Created?
yes
Delete Security Environment
no
Return to Caller

Protecting IOA Elements

The following elements can be protected within the IOA environment:

- IOA datasets
- Login to IOA
- IOA Conditions
- IOA Manual Conditions
- Control-M Quantitative Resources
- Control-M Control Resources
- Access to the VTAM Monitor
- Operator Commands

Details of how access to each IOA element is controlled are provided below.

Dataset Protection

The following dataset types are protected under the IOA environment.

IOA Installation Libraries

INCONTROL product users must have read access for the following IOA installation libraries:

Table 2 IOA Installation Libraries

Suffix	Description
CLIST	IOA CLIST Library
ISMSGENG	IOA ISPF Message Library (English)
ISMSGFRA	IOA ISPF Message Library (French)
ISMSGGER	IOA ISPF Message Library (German)
ISMSGJPN	IOA ISPF Message Library (Japanese)
LOAD	IOA LOAD Library
CTRANS	"C" Language Run Time Library
MSGENG	IOA Messages, Screen and Help Screen Library (English)

Suffix	Description
MSGFRA	IOA Messages, Screen and Help Screen Library (French)
MSGGER	IOA Messages, Screen and Help Screen Library (German)
MSGJPN	IOA Messages, Screen and Help Screen Library (Japanese)
PANELENG	IOA ISPF Panel Library (English)
PANELFRA	IOA ISPF Panel Library (French)
PANELGER	IOA ISPF Panel Library (German)
PANELJPN	IOA ISPF Panel Library (Japanese)
ROSLB	IOA ROSCOE RPF and Panel Library
PARM	IOA Operational Parameters Library
IOAENV	IOA Operational Parameters Library
DOC	IOA Documentation
CUSTEXIT	IOA Customized User Exits Source Code

For the following IOA Installation libraries, it is recommended (but not mandatory) that IOA users be denied access and that read only access be granted to IOA administrators and maintenance personnel. Update access must be permitted only to security personnel (for example, the person installing or maintaining IOA and the IOA security administrator).

Table 3 IOA Restricted Access Libraries

Suffix	Description
CICSSAMP	IOA CICS Sample Library
CONV	IOA Conversion Tools Library
JCL	IOA Sample JCL Library
MAC	IOA Macro Library
PROCLIB	IOA JCL Procedure Library
PROCJCL	IOA Started Task JCL Library
KSL	IOA KSL Scripts

Suffix	Description
SAMPLE	IOA Sample Library
SAMPREPS	IOA Sample Reports Library
SAMPEXIT	IOA Security Exits Library
SIML	IOA Simulation LOAD Library

IOA Operations Libraries

INCONTROL product users must have read access to these libraries. INCONTROL product administrators or selected groups of users must have update access to these libraries. The PROF Library (see below) must be a public library with update access for all IOA users, but only using programs activated from the IOA LOAD library.

Table 4 IOA Operations Libraries

Suffix	Description
BANNERS	IOA Banner Library
CAL	IOA Calendar Library
PROF	IOA Default User Profile Library

IOA Files

INCONTROL product users must have update access to these datasets, but only through programs activated from the IOA LOAD library. The specific user authorization within the IOA is determined by the relevant IOA security interface module, as explained beginning on Basic Definition Security Calls (on page 36).

Table 5 IOA Files

Suffix	Description
ALTCND	IOA Mirror (dual) Conditions File
LOG	IOA Log File
NRS	IOA Manual Conditions File
NSN	IOA Manual Conditions Synchronization File
CND	IOA Conditions File

IOA Maintenance Library

It is recommended that update access to files in this library be granted only to security personnel (for example, the IOA security administrator or the person installing or maintaining IOA).

Table 6 IOA Maintenance Library

Suffix	Description
MAINTLIB	IOA Maintenance Library

Protecting Access to Datasets Using Specific Programs

For information about how to protect access to datasets through programs and program pathing, see Limiting Access to Specific Programs (on page 200).

Security Resource Classes

RACF Security

The IOA interface uses the general resource class. A different user-defined class can be used for IOA by updating the appropriate table (for example, the Class-Descriptor-Table) and setting the IOACLASS parameter.

The following resource classes are used within IOA security:

Table 7 RACF Resource Classes

Class	Description
FACILITY	Used to control a user’s access to INCONTROL protected elements
SURROGAT	Used to check authorization of users submitting jobs on behalf of other users A surrogate user is a user who is authorized to submit jobs on behalf of another user without having to specify user ID and password
DATASET	Used to control a specified user ID’s access to datasets

TopSecret Security

The IOA interface uses the general resource class. A different user-defined class can be used for IOA by updating the appropriate TopSecret table (meaning, the RDT, Resource-Descriptor-Table), and setting the IOACLASS parameter.

The following resource classes are used within IOA TopSecret security:

The FACILITY class is converted by TopSecret to IBMFAC.

Table 8 TopSecret Resource Classes

Class	Description
FACILITY	Controls a user's access to IOA protected elements
ACIDCHK	Checks authorization of users to submit jobs on behalf of other users. A user can be authorized to submit jobs on behalf of another user without having to specify its user ID and password using permission to an ACIDCHK Resource Class.
DATASET	Controls a specified user's ID access to datasets

ACF2/SAF Security

The IOA interface uses the general resource class. A different user-defined class can be used for IOA by updating the appropriate ACF2 table (meaning, the CLASMAP definition in ACF2 version 6.x, or the SAFMAPS definition in ACF2 version 5.2), and setting the IOACLASS parameter.

The following resource classes are used within IOA ACF2/SAF security:

Table 9 ACF2 / SAF Resource Classes

Class	Description
FACILITY	Controls a user's access to IOA protected elements. This class is mapped to the ACF2 resource type CMF or other defined user type.
DATASET	Controls a specified user's ID access to datasets

The SAF compatibility mode option must be used. The System Authorization Facility (SAF) is a basic component of MVS that enables resource managers of the operating system to request security services. The ACF2/SAF interface translates SAF security requests into ACF2 requests. SAF is invoked by issuing a RACROUTE request. For more information, see the *CA-ACF2 System Programmer's Guide*.

Access to IOA Datasets

Users require access to datasets within the IOA environment. Access to IOA datasets must be granted during IOA installation. For more information on how to grant permissions to the IOA datasets, refer to the *INCONTROL for z/OS Installation Guide: Installing*.

Sign on to the IOA VTAM Monitor

Whenever a user attempts to sign on to the IOA VTAM monitor, the user's ID and associated password or password phrase are verified.

The IOASE09 IOA security module verifies the user ID and password or password phrase used to sign on to the IOA VTAM monitor.

Access to the IOA Online Facility

Whenever a user attempts to access the IOA Online facility, the user's authority is checked. This authorization check is performed for all environments in which the user attempts to access IOA, including TSO, Online monitor, CICS, ROSCOE, and so on.

The authorization check verifies that the logged on user has at least read access to a given entity associated with IOA access ($$IOAONLINE.qname), as defined in your security product.

Authorization is also checked when the IOA Online facility is accessed in batch using the IOA KeyStroke Language (KSL).

The IOASE06 IOA Security module verifies the user's authority to use the IOA Online facility.

For details of the entity names to be defined to your security product, see Basic Definition Security Calls (on page 36) and Extended Definition Security Calls (on page 37).

Operations on IOA Conditions and Control-M Resources

Whenever a user attempts to perform an operation on IOA conditions or Control-M resources (for example, Add, Delete), or when the operation is performed automatically, the user's authority is checked. The following IOA conditions and Control-M resources are checked:

- Prerequisite conditions – common IOA facilities that control tracking of conditions that must be met before activities such as job submissions are performed.
- Control resources – common Control-M facilities that Control resource sharing between jobs.
- Quantitative resources – common Control-M facilities that control allocation of computer resources such as tapes, CPUs, and so on, to maximize system throughput.

The security interface checks every type of operation to verify that the specified user is authorized to access a specific entity, as defined in your security product. The name of the entity is derived from the name of the condition or resource being accessed, and optionally, from the type of operation requested by the user.

For example, to access a condition called SYSCOND1 in Basic Definition mode, the user must be authorized to access the entity $$IOARES.qname.SYSCOND1. To delete a resource named SYSTAPE1 in Extended Definition mode, the user must be authorized to access the entity $$DELRES.qname.SYSTAPE1.

The IOASE07 IOA security module is invoked to verify the user's authorization to

- Add or delete a prerequisite condition to or from the IOA Conditions file.
- Add or delete a Control resource to or from the Control-M Resources file.
- Add, delete, or change a Quantitative resource to, from, or in the Control-M Resources file.
- Add a prerequisite condition to the IOA Manual Conditions file.

For more information about these elements, see Basic Definition Security Calls (on page 36) and Extended Definition Security Calls (on page 37).

User Authorization to Issue Operator Commands

Several IOA facilities (such as IOAOPR, Control-O rules) enable users to issue operator commands, including JES2 and JES3 commands.

The IOA security interface checks that the user is authorized to issue an operator command prior to executing the user's operator command request.

The authorization check is done by verifying that the specified user has access to the entity $$IOACMD.qname.command, where command is derived from the command text.

For example, to authorize processing of requests to issue all operator commands, users must be authorized to access entity $$IOACMD.qname.* or $$IOACMD.qname.**

To allow a user's request to issue the operator command 'D J,L', the user must be authorized to access entity $$IOACMD.qname.D.J.L

The CLASS checked is FACILITY; the entity used to check authorization is: $$IOACMD.qname.command-text, where command text is the first 20 characters of the operator command, compressed according to the following rules:

- Multiple non-alphanumeric characters are replaced with one period.
- A dot at the end of a command is dropped.
- All non-alphanumeric characters (blanks, commas, apostrophes, and so on) are replaced with dots.

Table 10 Operator Command Checking

Operator Command	Resulting ENTITY Checked
D J,L	$$IOACMD.qname.D.J.L
D J,L	$$IOACMD.qname.D.J.L
SE 'THIS IS A MESSAGE	$$IOACMD.qname.SE.THIS.IS.A.MESSAGE

The IOASE12 IOA security module verifies the user's authority to issue operator commands. This module functions in a similar manner under both Basic and Extended Definition modes.

Access to User Datasets

Whenever a user attempts to perform an operation on a user dataset, the user's authority to access the dataset and member is checked.

The IOASE32 IOA Security module verifies the user's authority to perform an edit, view or save action on JCL members, documentation members, definition tables or calendars from the IOA Online facility.

For more details about the definitions required in your security product, see Basic Definition Security Calls (on page 36) and Extended Definition Security Calls (on page 37).

Access to Control-D through the IOA Gateway

When a user logs on to Control-D through the IOA Gateway from the Control-D/Page On Demand feature, the IOASE16 security module is invoked to check the user's user ID and password. The security module also allows the user to optionally change their own password for MVS logon procedures (for example, TSO, CICS).

Invoking Utilities

When a particular batch utility is invoked, the IOASE40 security module checks to see if the user is authorized to run the utility.

Setting Global Variables

When a user attempts to set a global variable in a directory that is not the current directory, the IOASE42 security module is invoked to check the user's authorization to set global variables.

IOA XBM interface

Note: To activate the IOA to XBM (Extended Buffer Manager) interface, XBM must be active and at least one of the following parameters: ZIIPXBMO, ZIIPXBMP, or ZIIPXBMA must be set to Y.

A RACF call is made by XBM on the initial request to determine if a user is authorized to perform the requested function. The following RACF profile is used:

```
BMCXBM.<XBM_SSID>.ZIIP
```

If this profile is not defined, permission will be granted. More detailed information can be found in the XBM documentation.

Implementing IOA Security

This section describes the steps required to install the IOA security interface.

The IOA security interface can be installed either as part of the customized installation path, or during the Customization process after installation. Both options use the INCONTROL Installation and Customization Engine (ICE) application. If you are not familiar with the ICE interface, see the *INCONTROL for z/OS Installation Guide: Installing*.

During run time, the IOASECUR module determines which security product the site is using, and invokes the proper interface module (IOARACF for RACF, IOATSS for TopSecret, or IOAACF2 for ACF2/SAF) to create the security environment or clean up the security environment, and to make all necessary authorization checking. Therefore, one installation supports all three security products.

The IOA administrator must make several changes when TopSecret or ACF2 are the security products used by the site.

- To install the IOA TopSecret interface, do the following:

  Edit the IOASMP member in the IOA PROCLIB library.

  Add the following line, ensuring that the DSN parameters points to the correct library:
  ```
  //TSSMOD DD DISP=SHR,DSN=TSS.LOAD.LIBRARY
  ```

  Change the library name of the SECMTSS parameter to point to the TopSecret macro library.

- Save the member.

- To install the IOA ACF2 interface, do the following:

  Edit the IOASMP member in the IOA PROCLIB library.

  Add the following line, ensuring that the DSN parameters points to the correct library:
  //ACFMOD DD DISP=SHR,DSN=ACF2.LOAD.LIBRARY

  Change the library name of the SECMACF parameter to point to the ACF2 macro library.

  Save the member.

➢ To install the IOA security interface

1. Enter the main ICE screen.
2. Select Customization.
3. Enter IOA in the Product field.
4. Select Security Customization.
5. Perform all major and minor steps required to install the security product.

Step 1. Implement IOA Security

Use the following steps that correspond to the installation steps in ICE, to implement IOA security.

Step 1.1 Grant Access Permissions

Authorizations to access IOA datasets are optionally defined during IOA installation. It is recommended that this step be completed before proceeding with security implementation. For information about how to grant users access to IOA datasets, see the IOA installation chapter in the *INCONTROL for z/OS Installation Guide: Installing*.

Step 1.2 Customize Security Parameters

The following table describes the required parameters and their values:

Table 11 Customized Security Parameters

Parameter	Description
DEFMCHKI	When choosing a definition mode as COND to any of the IOA security modules, use qname together with the value given to this parameter as the high level qualifier, to determine the real definition mode to be used.
SECTOLI	Determines the IOA action to perform if your security product is inactive or a specific resource is not defined to the security product. Valid values are: ▪ YES – Perform the action. ▪ NO – Do not perform the action.
IOACLASS	Indicates the FACILITY class or a user-defined class. This is the class used by the IOA security interface for authority checks of IOA protected elements.
	Note: To use a user-defined class, you must define the class to your security product. For more information on how to define a resource class, see the *IBM RACF:SPL Guide*, *TopSecret Implementation Guide*, or *CA-ACF2 Administrator's Guide*.
IOAXCLAS	Indicates the extended FACILITY class or a user-defined class. This class must support long entity names and used by the IOA security interface for authority checks of IOA protected elements.
	Note: To use a user-defined class, you must define the class to your security product. For more information on how to define a resource class, see the *IBM RACF:SPL Guide*, *TopSecret Implementation Guide*, or *CA-ACF2 Administrator's Guide*.
IOATCBS	IOA Online monitor task level security ▪ NO – Operate all tasks in the IOA Online monitor with the IOAOMON or CTDOMON user ID authority. ▪ YES – Set each task in the IOA Online monitor to be activated with the logged on user authorizations.

Parameter	Description
IOACLNT	Determines the type of Login to the mainframe for the CONTROL-D/Web Access Server Communication users. ▪ NO – Single Login ▪ YES – Automatic Login For more information see the *Control-D WebAccess Server Administrator's Guide*.
Mode Definition	Specify one of the following values to determine the definition mode for the IOA security modules: ▪ COND – Conditional Definition mode. Default. ▪ BASIC – Basic Definition mode. ▪ EXTEND – Extended Definition mode.
DFMI06	Definition mode for the IOASE06 IOA security module
DFMI07	Definition mode for the IOASE07 IOA security module
DFMI09	Definition mode for the IOASE09 IOA security module
DFMI12	Definition mode for the IOASE12 IOA security module
DFMI16	Definition mode for the IOASE16 IOA security module
DFMI32	Definition mode for the IOASE32 IOA security module
DFMI40	Definition mode for the IOASE40 IOA security module
DFMI42	Definition mode for the IOASE42 IOA security module. This module is always in EXTEND mode.
ASAPPL	Name of Control-D Application Server started task. Add this parameter if you want to use the PassTicket feature for Control-D Application Server users.
VMONAPPL	Name of IOAVMON started task. Add this parameter if you want to use the PassTicket feature for VMON users.

Parameter	Description
UNIQID	Specify one of the following values to determine the uniqueness of the user ID: ▪ Y – User ID is unique, so that no more than one user can logon to an IOA application server with the same user ID. ▪ N, or left blank – User ID is not unique, so that more than one user can logon to an IOA application server with the same user ID. Default.

Step 1.3 Save Security Parameters into Product

This step saves all the security parameters specified for IOA.

When this step is completed, the Status column is automatically updated to COMPLETE.

Step 1.4 Select Security Product Interface

The IOA installation supports RACF, TopSecret, and ACF2. The IOA security interface automatically determines which security product is being used at runtime. More than one security product can be selected.

Specify the security products you want to support, and follow the instructions in this step.

Step 1.5 Build IOA RACF Interface

The IOASRACF job in the IOA INSTWORK library creates the IOA security interface module for RACF.

Submit the job. All steps of this job must end with a condition code of 4 or less.

Step 1.6 Build IOA TopSecret Interface

The IOASTSS job in the IOA INSTWORK library creates the IOA security interface module for TopSecret.

Submit the job. All steps of this job must end with a condition code of 4 or less.

Check all lines that are marked with "MODIFY" and ensure that the proper dataset names are set.

Step 1.7 Build IOA ACF2 Interface

The IOASSAF job in the IOA INSTWORK library creates the IOA security interface module for ACF2.

Submit the job. All steps of this job must end with a condition code of 4 or less.

Check all lines that are marked with "MODIFY" and ensure that the proper dataset names are set.

Step 1.8 Activating IOA and INCONTROL Product Security

To activate the IOA security or any INCONTROL product security, the security administrator must define the following entities in the relevant security product:

For IOA, define entity $$SECIOA.qname

For Control-M, define entity $$SECCTM.qname

For Control-D, define entity $$SECCTD.qname

For Control-O, define entity $$SECCTO.qname

For Control-M/Analyzer, define entity $$SECCTB.qname

For Control-M/Tape, define entity $$SECCTT.qname

where qname is defined during the IOA installation process.

BMC recommends that you define the entities as fully qualified in order to distinguish between different environments, and define the entities with universal READ access.

The class which these entities must be defined are as follows:

- RACF — The class is FACILITY
- TopSecret — The class is IBMFAC
- ACF2 — The resource type is CMF

Step 2. Link IOA Security Modules (Optional)

- **Step 2.1 Link Security Modules to IOA COMPLETE Online Interface Modules (Optional)**

  The IOASCPL job in the IOA INSTWORK library is optional. This job is used only if COM-PLETE is installed.

  Submit this job to set SMP/E to link the security modules into the IOA COM-PLETE online interface modules.

  This job must end with a condition code of 4 or less.

Check all lines that are marked with "MODIFY" and make sure that the proper dataset names are specified.

- **Step 2.2 Link Security Modules into IOA IDMS Online Interface Modules (Optional)**

  The IOASIDM job in the IOA INSTWORK library is optional. This job is used only if IDMS is installed.

  Submit this job to set SMP/E to link the security modules to the IOA IDMS online interface modules.

  This job must end with a condition code of 4 or less.

Check all lines that are marked with "MODIFY" and make sure that the proper dataset names are specified.

- **Step 2.3 Link Security Modules into IOA IMS Online Interface Modules (Optional)**

  he IOASIMS job in the IOA INSTWORK library is optional. This job is used only if IMS is installed.

  Submit this job to set SMP/E to link the security modules to the IOA IMS online interface modules.

  This job must end with a condition code of 4 or less.

Check all lines that are marked with "MODIFY" and make sure that the proper dataset names are specified.

Step 3. RACF Security Definition Samples (Optional)

Note: To activate the IOA to XBM interface, XBM must be active and at least one of the following parameters: ZIIPXBMO, ZIIPXBMP, or ZIIPXBMA must be set to Y.

A RACF call is made by XBM on the initial request to determine if a user is authorized to perform the requested function.   The following RACF profile is used:

BMCXBM.<*XBM_SSID*>.ZIIP

If this profile is not defined, permission will be granted. More detailed information can be found in the XBM documentation.

Step 3.1 IOA Security Definitions (Optional)

IOA security definition samples are found in the IOASRAC2 member of the IOA INSTWORK library. This member is created in the IOA INSTWORK library after selecting this step.

1. Associate users with extended definition mode.
   1. When using Conditional Definition mode, define the entity $$IOAEDM.qname using the following command:

      ```
      RDEFINE FACILITY $$IOAEDM.qname UACC(NONE)
      ```

      Force USERA to work in the Extended Definition mode by using the following command:

      PERMIT $$IOAEDM.qname ID(USERA) CLASS(FACILITY) ACCESS(READ)

      Users who have read authority to this entity will work in the Extended Definition mode. Users who are not authorized to access this entity work in the Basic Definition mode.
   2. Submit the job for execution.
2. Verify that the SURROGAT class is active.
   1. Use the following command to list the active classes:

      ```
      SETROPTS LIST
      ```
   2. To activate the SURROGAT class, use the following command:

      ```
      SETROPTS CLASSACT(SURROGAT)
      ```

Step 3.2 Function Security Definitions (Optional)

The IOASRAC3 job in the IOA INSTWORK library is optional. It contains some definition samples for various entities. Customize this job according to your requirements and submit the job.

Define entities and user authorizations.

For information about defining IOA entities and user authorizations, see Basic Definition Security Calls (on page 36), and Extended Definition Security Calls (on page 37).

Examples

The IOASRAC4 job in the IOA INSTWORK library contains a sample of the definitions required to define Program Pathing access authorizations to IOA datasets. Review the definitions and modify them according to the requirements of your site.

Before submitting this job, BMC Software recommends that the security administrator read Limiting Access to Specific Programs (on page 200) and read about protecting entities through Program Pathing in the manual of your security product.

- To control access to the IOA Online facility, specify the following command:

  ```
  RDEFINE FACILITY $$IOAONLINE.qname
  ```

  where qname is used to assign different authorizations to different IOA environments (such as Test and Production). This parameter is specified during IOA installation.

- To define and authorize all conditions beginning with SYS, use the following command:

  ```
  RDEFINE FACILITY $$IOARES.qname.SYS*
  PERMIT $$IOARES.qname.SYS* CLASS(FACILITY) ID(USERA) ACCESS(READ)
  ```

- To authorize USERA access to a given IOA entity, use the following command:

  ```
  PERMIT $$IOAnnn.qname CLASS(FACILITY) ID(USERA) ACCESS(READ)
  ```

  All entity names for each IOA protected element appear in Basic Definition Security Calls (on page 36) for Basic Definition mode and Extended Definition Security Calls (on page 37), for Extended Definition mode.

Step 3.3 Control Program Access to IOA Datasets (Optional)

The IOASRAC4 job in the IOA INSTWORK library contains a sample of the definitions required to define Program Pathing access authorizations to IOA datasets. Review the definitions and modify them according to the requirements of your site.

Before submitting this job, BMC Software recommends that the security administrator read Limiting Access to Specific Programs (on page 200) and read about protecting entities through Program Pathing in the manual of your security product.

Step 4. TopSecret Security Definition Samples (Optional)

Step 4.1 IOA Security Definitions (Optional)

IOA security definition samples are found in the IOASTSS2 member of the IOA INSTWORK library. The IOASTSS2 member is created in the IOA INSTWORK library after selecting this step.

1. Define IOA monitors in the TopSecret Facility Matrix

   IOASTSS2 contains the necessary command to dynamically define IOA in TopSecret Facility Matrix.

   a. Modify USER1 in the facility definition command to a free entry in the Facility Matrix, as follows:

      ```
      TSS MODIFY FAC(USER1=NAME=IOA)
      ```

      This command defines IOA in the Facility Matrix until the next IPL.

   b. Update the TopSecret parameter member, usually called TSSPARM0, to permanently define the facility.

   c. Copy the IOA facility definition from the IOASTSS5 member in the IOA BASE INSTALL library to the TSSPARM0 member.

   d. Update the Facility Matrix entry name to the same name that is specified in the TSS MODIFY command.

2. Define IOA ACID to TopSecret

   Change the DEPT parameter value from security administrator, department to the appropriate ACID:

   ```
   TSS CRE(IOA) NAME (...) DEPT(sec-administrator-dept)
   ```

3. Define IOA procedures (started tasks) to TopSecret

   Change the ACID definition in the following commands to the appropriate ACID:

```
TSS ADD(STC) PROC(IOAOMON1) ACID(IOA)
TSS ADD(STC) PROC(IOAVMON) ACID(IOA)
```

4. Connect the appropriate profile to the IOA ACID in the following command:

   ```
   TSS ADD(IOA) PROF(profile-name)
   ```

   IOAOMON must be authorized to any datasets that are accessed by online users.

5. Connect usera to the IOA ACID in the following command:

   ```
   TSS ADD(usera) FAC(IOA)
   ```

6. Define IOA entities and user authorizations to TopSecret

For information about how to define IOA entities and user authorizations to TopSecret, see Basic Definition Security Calls (on page 36), and Extended Definition Security Calls (on page 37).

   Modify the following command to establish resource ownership in TopSecret to the appropriate owner:

   TSS ADD(sec-administrator-dept) IBMFAC($$IOA)

   For samples of user authorizations, review member IOASTSS3 in the IOA INSTWORK library.

   All entity names for each IOA protected element appear in Basic Definition Security Calls (on page 36), for Basic Definition mode and Extended Definition Security Calls (on page 37) , for Extended Definition mode.

7. Associate users with Extended Definition modes

   Customize the following TopSecret command to establish Extended Definition mode for the IOA installer.

   ```
   TSS PERMIT (USERA) IBMFAC($$IOAEDM.qname) ACC(READ)
   ```

   Change USERA to the UID of IOA installer.

   When an IOA security module is customized to CONDitional mode without access to this entity, the user works in Basic Definition mode. With access, the user works in Extended Definition mode.

Do not define the $$IOAEDM entity to operate in warning mode since this causes all users to operate in Extended Definition mode.

8. Authorize the IOA installer to use IOA facilities.

   a. Customize the following command to authorize USERA access to the Online monitor:

      ```
      TSS ADD(USERA) FACILITY(IOA)
      ```

   b. Change USERA to the user ID of the IOA installer.

   c. Customize the following command to authorize the IOA installer to use IOA facilities:

      ```
      TSS PERMIT(USERA) IBMFAC($$IOA) ACC(READ)
      ```

   d. Submit the job.

      This job must be run under the ACID of the general security administrator (SCA) who has authorization to enter TopSecret commands.

      All job steps must end with a condition code of zero.

Step 4.2 Function Security Definitions (Optional)

The IOASTSS3 job in the IOA INSTWORK library is optional. It contains some definition samples for various entities. Customize this job according to your requirements and submit this job.

Step 4.3 Control Program Access to IOA Datasets (Optional)

The IOASTSS4 job in the IOA INSTWORK library contains a sample of the definitions required to define Program Pathing access authorizations to IOA datasets. Review the definitions and modify them according to the requirements of your site.

Before submitting this job, BMC Software recommends that the security administrator read Limiting Access to Specific Programs (on page 200) and read about protecting entities through Program Pathing in the manual of your security product.

Step 4.4 Define IOA to TopSecret Facility Matrix (Optional)

The IOASTSS5 member in the IOA INSTWORK library contains the necessary definitions to define IOA in TopSecret Facility Matrix.

Modify USER1 in the facility definition command to a free entry in the Facility Matrix, by copying the IOA facility definition from this member (IOASTSS5) member in the IOA BASE INSTALL library to the TSSPARM0 member.

Step 5. ACF2 Security Definition Samples (Optional)

Step 5.1 IOA Security Definitions (Optional)

IOA security definition samples are found in the IOASSAF2 member of the IOA INSTWORK library. The IOASSAF2 member is created in the IOA INSTWORK library after selecting this step.

Review this member and make all required changes according to your site.

1. Define IOA to CA-ACF2/SAF interface.

   The IOASSAF2 member contains ACF2 commands that translate the SAF class FACILITY to the ACF2 resource type CMF.

   If the SAF class FACILITY is already in use, check if this class is translated to another ACF resource type. Specify the following command:

   ```
   SET CONTROL(GSO) SYSID(****)
   SHOW CLASMAP
   ```

   For versions prior to ACF2 version 6.0, specify the following command:

   ```
   SET CONTROL(GSO) SYSID(****)
   LIST SAFMAPS
   ```

   This command determines the resource type to which the class FACILITY is translated. If the resource type is not CMF, change all occurrences of resource type "CMF" in this member to the resource type in use.

2. Associating users with Extended Definition mode

   Add the following ACF2 commands to define the $$IOAEDM entity to ACF2 and authorize users to this entity.

   ```
   SET RESOURCE(CMF)
   COMP
   $KEY($$IOAEDM.qname)
    UID(USERA) ALLOW
   ```

   If the user does not have access to this entity, the user is set to work in Basic Definition mode. Otherwise, the user is set to work in Extended Definition mode.

3. Submit the job.

   Verify that all job steps end with a condition code of 0.

   This job must be run under a user of an ACF2 administrator who is authorized to enter these ACF2 commands.

4. Refresh ACF2 definitions.

   Issue the following MVS commands to refresh the definitions of the ACF2:

   ```
   F ACF2,REFRESH(SAFDEF)
   F ACF2,REFRESH(CLASMAP)
   ```

   For versions prior to ACF2 version 6.0, run the commands:

   ```
   F ACF2,REFRESH (OPTS)
   F ACF2,REFRESH (SAFMAPS)
   ```

5. Rebuild the resource type CMF rules.

   Issue the following MVS command:

   ```
   F ACF2,REBUILD(CMF)
   ```

   Add the ACF2 commands to the GSO member in the ACF2 parameter library. These commands are specified in the IOASSAF2 member.

   This job must end with a condition code of 4 or less.

Step 5.2 Function Security Definitions (Optional)

Job IOASSAF3 in the IOA INSTWORK library is optional. It contains some definition samples for various entities. Customize this job according to your use and submit this job.

1. Define entities and user authorizations to ACF2.

   For information about how to define IOA entities and user authorizations to ACF2, see Basic Definition Security Calls (on page 36) and Extended Definition Security Calls (on page 37).

To permit USERA to use the IOA Online facility, specify the following command:

```
SET RESOURCE(CMF)
COMP
$KEY($$IOAEDM.qname)
 UID(USERA) ALLOW
```

where qname is the name used to assign different authorizations to different IOA environments (such as Test and Production). This parameter is specified during IOA installation.

To define and authorize all conditions beginning with SYS, use the following command:

```
SET RESOURCE(CMF)
COMP
$KEY($$IOARES.qname.SYS************************)
 UID(USERA) ALLOW
```

To authorize USERA (the user ID of the IOA installer) access to a given IOA entity, use the following command:

```
SET RESOURCE(CMF)
COMP
$KEY($$IOAnnn.qname)
 UID(USERA) ALLOW
```

All entity names for each IOA protected element appear in Basic Definition Security Calls (on page 36) and Extended Definition Security Calls (on page 37).

2. Submit the job.

Verify that all job steps end with a condition code of 0.

This job must be run under a user of an ACF2 administrator who is authorized to enter these ACF2 commands.

Step 5.3 Control Program Access to IOA Datasets (Optional)

The IOASSAF4 job in the IOA INSTWORK library contains a sample of the definitions required to define Program Pathing access authorizations to IOA datasets. Review the definitions and modify them according to the requirements of your site.

Before submitting this job, BMC Software recommends that the security administrator read Limiting Access to Specific Programs (on page 200) and read about protecting entities through Program Pathing in the manual of your security product.

Step 5.4 Define IOA to CA-ACF2/SAF Interface (Optional)

The IOASSAF5 job in the IOA INSTWORK library contains a sample of the definitions to ACF2. Review this job and make all required modifications according to the requirements of your site, and submit the job.

1. Rebuild the resource type CMF rules.

a. Issue the following MVS command:

F ACF2,REBUILD(CMF)

b. Add the ACF2 commands to the GSO member in the ACF2 parameter library. These commands are specified in member IOASSAF5. This job must end with a condition code of 4 or less.

IOA Security Interface Modules

This section provides details on each IOA security interface module and user exit.

Basic Definition Security Calls

Table 12 IOA Basic Definition Security Calls

Protected Element	Type	Class Entity Name	Explanation	Security Module
IOA Online facility	all	FACILITY $$IOAONLINE.*qname*	*qname* is the name used to assign different authorizations to various IOA environments (such as Test and Production).	IOASE06
VTAM monitor	all		Verifies user password or password phrase. Allows user to reenter password or password phrase.	IOASE09
Controlling IOA Conditions and Control-M Resources				
IOA Condition	all	FACILITY $$IOARES.*qname.cond*	*cond* is the condition name.	IOASE07
Control-M Quantitative Resource	all	FACILITY $$IOARES.*qname.res*	*res* is the Quantitative resource name.	IOASE07
Control-M Control Resource	all	FACILITY $$IOARES.*qname.cntl*	*cntl* is the Control resource name.	IOASE07
IOA Manual Condition	all	FACILITY $$IOARES.*qname.mancnd*	*mancnd* is the manual condition name.	IOASE07
Entering Operator Command	all	FACILITY $$IOACMD.*qname.cmndtext*	*cmndtext* is the first 20 characters of the operator command.	IOASE12
Control-D Application Servera using IOAGATE	all	FACILITY $$IOAAS.*qname*		IOASE16
User Dataset	all	DATASET*dataset-name*		IOASE32

Protected Element	Type	Class Entity Name	Explanation	Security Module
Running Batch Utilities	all	FACILITY $$IOAUTL.*qname*.*utility-name*	*utility-name* is the name of the utility being invoked.	IOASE40
Variable database	all	FACILITY ADMIN mode $$IOAGL.*qname*.ADMINDB.*database* $$IOAGL.*qname*.SETDBVAR.*database* $$IOAVD.*qname*.M.*application* NON-ADMIN mode $$IOAGL.*qname*.SETPLVAR.*database* $$IOAVP.*qname*.M.*application*	*database* is the database name. *application* is the application name in the variable name.	IOASE42

Extended Definition Security Calls

Table 13 IOA Extended Definition Security Calls

Protected Element	Type	Class Entity Name	Explanation	Security Module
Controlling Access to the IOA Online Facility				
IOA Online facility	all	FACILITY $$IOAONLINE.*qname*	*qname* is the name used to assign different authorizations to various IOA environments (such as Test and Production).	IOASE06
VTAM monitor	all		Verifies user password or password phrase. Allows user to reenter password or password phrase.	IOASE09

Protected Element	**Type**	**Class Entity Name**	**Explanation**	**Security Module**
Controlling IOA Conditions and Control-M Resources				
IOA Condition	all	FACILITY Add: $$ADDCND.*qname.cond* Delete: $$DELCND.*qname.cond* Check: $$CHKCND.*qname.conda*	*cond* is the condition name.	IOASE07
Control-M Quantitative Resource	all	FACILITY Add: $$ADDRES.*qname.res* Delete: $$DELRES.*qname.res* Change: $$CHARES.*qname.res* Check: $$CHKRES.*qname.res*	*res* is the Quantitative resource name.	IOASE07
Control-M Control Resource	all	FACILITY Add: $$ADDCTL.*qname.cntl* Delete: $$DELCTL.*qname.cntl* Check: $$CHKCTL.*qname.cntl*	*cntl* is the Control resource name.	IOASE07
IOA Manual Condition	all	FACILITY Define: $$NEWCND.*qname.mancnd* Erase: $$ERACND.*qname.mancnd*	*mancnd* is the manual condition name.	IOASE07
Entering Operator Command	all	FACILITY $$IOACMD.*qname.cmndtext*	*cmndtext* is the first 20 characters of the operator command.	IOASE12
Control-D Application Serverb using IOAGATE	all	FACILITY $$IOAAS.*qname*		IOASE16

Protected Element	Type	Class Entity Name	Explanation	Security Module
User Dataset	all	DATASET *dataset-name* FACILITY Dir:$$IOADIR.*qname*. View: $$IOAVIW.*qname.mem* Edit: $$IOAEDT.*qname.mem* Save: $$IOASAV.*qname.mem* Del: $$IOADEL.*qname.mem*	*mem* is the member name.	IOASE32
Running Batch Utilities	all	FACILITY $$IOAUTL.*qname.utility-name*	*utility-name* is the name of the utility being invoked.	IOASE40
Variable database	all	FACILITY ADMIN mode $$IOAGL.*qname*.ADMINDB. *database* $$IOAGL.*qname*.SETDBVAR. *database* $$IOAVD.*qname*.M. *application*. NON-ADMIN mode $$IOAGL.*qname*.SETPLVAR. *database* $$IOAVP.*qname*.M.*application*	*database* is the database name. *application* is the application name in the variable name.	IOASE42

Module IOASE06

The IOASE06 module is the security module of IOA Exit IOAX006. This module checks a user's authority to enter the IOA Online environment. Optionally, an IOA Entry Panel is displayed requiring the user to specify user ID and password or password phrase.

The module does not display the IOA Entry Panel as a default because it is assumed that the user has already passed through a logon screen when entering the current online environment (for example, session manager, TSO entry screen, CICS sign on screen).

When the module is invoked under the IOA Online monitor, it creates the User Security Profile (ACEE). The Online monitor is a multiuser address space, and different users can have different access authorizations. Under a single user address space such as a TSO user, or a batch job, the security profile initialization has already been performed by the operating system.

Use User Exit IOAX006 to perform user functions, and the IOASE06 module to perform security functions.

User Exit IOAX006 does not return to the IOA entry panel. The IOA entry panel can only be displayed by the IOASE06 security module.

The IOASE06 security module and user Exit IOAX006 are also invoked when using the IOA KeyStroke Reporting Language (KSL), that performs IOA Online functions in batch. Therefore, the same security functions are performed under all IOA environments.

Module IOASE07

The IOASE07 module is the security module of IOA Exit IOAX007. This module verifies the user’s authorization to add, delete, or change prerequisite conditions, Control resources and Quantitative resources.

The CLASS checked is FACILITY; the entity used to check authorization depends on whether Basic or Extended definition modes are used:

CMEM Default User ID

This module checks the user's authorization to perform a CMEM DO COND statement when runtime security is not active (the rule’s RUNTSEC parameter is set to NONE). If the rule’s OWNER parameter is set to CTMCTLM, user ID CTMCTLM is checked. Otherwise, user ID CTOCTLO is checked.

Basic Definition Mode

The entity used to check authorization is $$IOARES.qname.name , where name is the resource name or the condition name.

Extended Definition Mode

The entity built for verification depends on what screen is being used:

Screen 4 – Conditions/Resources Screen

For prerequisite conditions, use the actions listed in the following table:

Table 14 Prerequisite Condition Actions

Action	Entity
ADD	$$ADDCND.qname.condition-name
DEL	$$DELCND.qname.condition-name

For quantitative resources, use the actions listed in the following table:

Table 15 Quantitative Resource Actions

Action	Entity
ADD	$$ADDRES.qname.resource-name
DEL	$$DELRES.qname.resource-name
CHANGE	$$CHARES.qname.resource-name

For control resources, use the actions listed in the following table:

Table 16 Control Resource Actions

Action	Entity
ADD	$$ADDCTL.qname.control-name
DEL	$$DELCTL.qname.control-name

Screen 7 – Manual Conditions Screen

Table 17 Manual Condition Actions

Action	Entity
NEW (add)	$$NEWCND.qname.condition-name
ERASE (delete)	$$ERACND.qname.condition-name

Module IOASE09

The IOASE09 module is the security module of IOA Exit IOAX009. This module checks the user's authority to enter the IOA Online environment under CICS, IMS/DC, VTAM, IDMS/DC, COM-PLETE, ROSCOE and TSO cross-memory options. It is not invoked under TSO (native or ISPF), or native ROSCOE.

Optionally, a site can choose to display an IOA entry panel, requiring the user to specify user ID and password or password phrase.

The IOASE09 module is invoked under the user environment (for example, CICS). When a check is successfully passed, the IOA Online monitor is invoked. The IOASE06 module is also invoked using the user ID passed from the user environment; it performs an additional security check.

The IOASE09 module displays the IOA entry panel as a default only by IOAVMON. Otherwise, it is assumed that the user has already passed through an entry or password screen when entering the current online environment. However, the IOA entry panel is displayed under IOAVMON. The user ID and password or password phrase is verified.

Use Exit IOAX009 to perform any required user functions, while IOASE09 should be used for security functions. Note that user Exit IOAX009 cannot display the IOA entry panel. This can only be done through the IOASE09 security module.

Module IOASE12

The IOASE12 module is the security module of IOA Exit IOAX012. This module verifies that a user is authorized to issue an operator command from within the IOA environment as a result of a user action or request. The IOA utilities IOAOPR, CTMOPR, CTDOPR allow users to issue operator commands, including JES2 and JES3 commands. These utilities are also activated under CTMTDAY, CTDNDAY, the Control-O monitor and whenever the user activates the program CTM34F.

For additional information, see User Authorization to Issue Operator Commands (on page 23).

Module IOASE16

The IOASE16 module is the security module of IOA Exit IOAX016. This module is called when a logon request is made by Control-D/Page On Demand using the IOA Gateway (IOAGATE). This module determines and builds the user's identity for all subsequent actions. The ACEE control block is built and its address is saved in the OCT (Online Control Table). All authorization checks are performed using the user's ACEE control block. If the ACEE control block is not built, the security interface will probably not perform the authorization checks correctly. To build the ACEE control block, the user must be defined to your security product and must have READ access to entity $$IOAAS.qname.

Module IOASE32

The IOASE32 module is the security module of IOA user Exit IOAX032. This module verifies that the user is authorized to Edit or View JCL members, documentation members, tables or calendars from the CTMAS workstation.

Basic Definition Mode

The CLASS checked is DATASET and the entity built is the dataset name of the library.

The access level used to check this authorization depends on user request:

Table 18 Basic Definition Request Authorization

Request	Authorization
EDIT request	read
SAVE request	update
VIEW request	read
DIR request	read
DEL request	update

Extended Definition Mode

Two checks are performed:

1. Dataset access

   The CLASS checked is DATASET. The entity used to check authorization is: the dataset name of the library. The access level used to check this authorization depends on the user request:

Table 19 Extended Definition Request Authorization

Request	Authorization
EDIT request	read
SAVE request	update
VIEW request	read
DIR request	read
DEL request	update

2. Operations

   Under IOA, the CLASS checked is FACILITY. The entity used to check authorization depends on the user request:

Table 20 Operation Access

Request	Authorization
DIR	$$IOADIR.qname
EDIT	$$IOAEDT.qname.member
VIEW	$$IOAVIW.qname.member
SAVE	$$IOASAV.qname.member
DEL	$$IOADEL.qname.member

Module IOASE40

The IOASE40 module verifies that the user is authorized to invoke batch utilities. This module is activated when the following utilities are invoked:

- IOADDC
- IOAVERFY
- CTTPTI

The CLASS checked is FACILITY and the entity used to check authorization is

$$IOAUTL.qname.utility-name

where utility-name is the name of the utility being invoked.

Module IOASE42

The IOASE42 module verifies that the user is authorized to create and/or update the IOA Global variable database. This module is always in the EXTEND mode.

IOASE042 protects the IOA Global variable database in ADMIN and non-ADMIN mode.

The CLASS checked is FACILITY and the entity used to check authorization depends on the user request, as described in the following tables:

Table 21 Operation Access non-IOAVAR database

Request	Authorization	Mode
Insert a new database	$$IOAGL.qname.ADMINDB.database	ADMIN
Update the description of the database	$$IOAGL.qname.ADMINDB.database	ADMIN
Insert a new column in the database	$$IOAGL.qname.ADMINDB.database	ADMIN
Update the information in the selected column	$$IOAGL.qname.ADMINDB.database	ADMIN
Delete the selected column	$$IOAGL.qname.ADMINDB.database	ADMIN
Zoom - show and edit the variables of the row	$$IOAGL.qname.SETDBVAR.database	ADMIN
Repeat the row	$$IOAGL.qname.SETDBVAR.database	ADMIN
Delete the current row	$$IOAGL.qname.SETDBVAR.database	ADMIN
Zoom - show and edit the variables of the row	$$IOAGL.qname.SETPLVAR.database	Non-ADMIN

(where database is the database name)

Table 22 Operation Access IOAVAR database

Request	Authorization	Mode
Update the description of the database	$$IOAGL.qname.ADMINDB.IOAVAR	ADMIN
Zoom - show and edit the variables of the row	$$IOAVD.qname.M.application $$IOAVP.qname.M.application	ADMIN Non-ADMIN
Insert a new row in the database	$$IOAVD.qname.M.application	ADMIN
Repeat the row	$$IOAVD.qname.M.application	ADMIN
Delete the current row	$$IOAVD.qname.M.application	ADMIN

(where application is the application name in the variable name)

Installing Control-M Application Server Security

The Control-M Application Server (CTMAS) security interface uses the security mechanisms provided for IOA and Control-M. For each Control-M/Enterprise Manager user operation (for example, Hold, Rerun), the workstation gateway transfers the requesting Control-M/Enterprise Manager user ID to the application server. The application server checks the authorization of this user using the IOA and Control-M security modules. If the user is not authorized to perform this operation, the operation is rejected and the workstation issues an error message.

User IDs are always defined in uppercase letters on the mainframe. User IDs are usually defined in lowercase under Unix. Therefore, the application server automatically converts the workstation user IDs to uppercase for compatibility with the mainframe definitions.

Implementing CTMAS Security

Before implementing CTMAS security, the security administrator should read this section, and be familiar with ICE.

Protecting CTMAS Elements

CTMAS security uses the standard IOA and Control-M security interfaces to protect the following elements:

- IOA conditions (the IOASE07 IOA security module)
- Jobs displayed in the Active Environment screen (the CTMSE08 Control-M security module)
- JCL members, documentation members, job orders in the Active Jobs file, tables, and calendars (the IOASE32 IOA security module)

Controlling Operations on IOA Conditions and Control-M Resources

The IOASE07 IOA security module is invoked to verify the user's authorization to

- Add or delete a prerequisite condition to or from the IOA Conditions file.
- Add or delete a Control resource to or from the Control-M Resources file.
- Add, delete, or change a Quantitative resource to, from, or in the Control-M Resources file.
- Add or erase a prerequisite condition from the IOA Manual Conditions file.

For more information, see Module IOASE07 (on page 40).

Controlling Access to Edit, Save, or View JCL, Documentation, Table and Calendar Members

Although Control-M/Enterprise Manager users do not operate directly in the mainframe environment, they can issue a request to edit, save, or view user datasets, JCL members, documentation members, tables, calendars, and so on through the Control-M Application Server (CTMAS). The IOASE32 IOA security module is invoked to verify the user's authority to perform such operations. For more information, see Module IOASE32 (on page 42).

Controlling Access to Jobs in the Active Jobs File

User actions performed in the Control-M/Enterprise Manager application are similar to the same user actions performed under Screen 3 in Control-M. Therefore, when a Control-M/Enterprise Manager user issues inquiries about a job within the Active Jobs file, or attempts to change a job's status through Screen 3, the CTMSE08 security module is invoked to verify that the user is authorized to perform the attempted action. For more information, see Module CTMSE08 (on page 56), and Control-M Security (on page 59).

Access to the M2G file

The M2G dataset is shared by CTMAS, Control-M and (optionally) other INCONTROL products that can issue SHOUT messages to Control-M/Enterprise Manager (such as Control-O and Control-M/Analyzer). Therefore, the M2G file requires update authority for these products.

CTMAS Basic Definition Security Calls

Table 23 CTMAS Basic Definition Security Calls

Protected Element	Class Entry Name	Explanation	Security Module
Controlling Operations on IOA Conditions and Resources			
Access an IOA Condition	FACILITY $$IOARES.qname.cond	qname is the name used to assign different authorizations to various environments (such as Test and Production). cond is the condition name.	IOASE07
Access IOA Quantitative Resource	FACILITY $$IOARES.qname.res	res is the Quantitative resource name.	IOASE07
Access an IOA Control Resource	FACILITY $$IOARES.qname.cntl	cntl is the Control resource name.	IOASE07
Access an IOA Manual Condition	FACILITY $$IOARES.qname.mancnd	mancnd is the manual condition name.	IOASE07
Access JCL library members	DATASET dataset-name		IOASE32
Access Documentation library members	DATASET dataset-name		IOASE32
Access Table/Calendar library members	DATASET dataset-name		IOASE32
Access library members	DATASET dataset-name		IOASE32

Protected Element	Class Entry Name	Explanation	Security Module
Controlling Access to the Active Environment Screen			
Access the Active Environment Screen (Screen 3)	FACILITY $$CTMPNL3.qname		CTMSE08
Perform Operations in the Active Environment Screen	SURROGAT owner.SUBMIT ACIDCHK owner FACILITY $SUBMIT.owner	owner is the name of the user specified in the job order definition.	CTMSE08

CTMAS Extended Definition Security Calls

Table 24 CTMAS Extended Definition Security Calls

Protected Element	Class Entry Name	Explanation	Security Module
Controlling Operations on IOA Conditions and Resources			
Access IOA Conditions	FACILITY Add: $$ADDCND.qname.cond Delete: $$DELCND.qname.cond	qname is the name used to assign different authorizations to various environments (such as Test and Production). cond is the condition name.	IOASE07
Access an IOA Quantitative Resource	FACILITY Add: $$ADDRES.qname.res Delete: $$DELRES.qname.res Change: $$CHARES.qname.res Check: $$CHKRES.qname.res	res is the Quantitative resource name.	IOASE07
Access an IOA Control Resource	FACILITY Add: $$ADDCTL.qname.cntl Delete: $$DELCTL.qname.cntl Check: $$CHKCTL.qname.cntl	cntl is the Control resource name.	IOASE07
Access Manual Conditions	FACILITY Define: $$NEWCND.qname.mancnd Erase: $$ERACND.qname.mancnd	mancnd is the manual condition name.	IOASE07

Protected Element	Class Entry Name	Explanation	Security Module
Access JCL library members	DATASET dataset-name FACILITY Edit: $$ECSEDJ.qname.mem Save: $$ECSSVJ.qname.mem View: $$ECSVWJ.qname.mem	mem is the member name.	IOASE32
Access Documentation library members	DATASET dataset-name FACILITY Edit: $$ECSEDD.qname.mem Save: $$ECSSVD.qname.mem View: $$ECSVWD.qname.mem	mem is the member name.	IOASE32
Access Table or Calendar library members	DATASET dataset-name FACILITY Edit: $$ECSEDF.qname.mem Save: $$ECSSVF.qname.mem View: $$ECSVWF.qname.mem	mem is the member name.	IOASE32
Delete a Table	$$ECSTTB.qname.table	table is the name of the table to be deleted.	IOASE32
Delete a Calendar	$$ECSTCL.qname.cal	cal is the name of the calendar to be deleted.	IOASE32

Protected Element	Class Entry Name	Explanation	Security Module
Control Access to the Environment Screen			
Access the Active Environment Screen (Screen 3)	FACILITY $$CTMPNL3.qname		CTMSE08
Perform Operations in the Active Environment Screen	FACILITY React: $$JOB1ACT.qname.owner Browse: $$JOB1SYS.qname.owner Stats: $$JOB1STA.qname.owner Zoom: $$JOB1ZOO.qname.owner Log: $$JOB1LOG.qname.owner Hold: $$JOB2HLD.qname.owner Free: $$JOB2FRE.qname.owner Rerun: $$JOB2RRN.qname.owner Restore :$$JOB2RRN.qname.owner Confirm: $$JOB2CNF.qname.owner Change: $$JOB2CHA.qname.owner Priority: $$JOB3PRI.qname.owner Delete: $$JOB3DEL.qname.owner EditJCL: $$JOB3EDI.qname.owner	owner is the name of the user specified in the job order definition.	CTMSE08

Step 6. Control-M Application Server — RACF

Step 6.1 Grant Access Permission

Collect the required data in order to define the INCONTROL entities and user authorizations to the security product.

You can use this data in the sample jobs provided in the subsequent steps "Control-M Application Server Security Definitions (Sample)" and "Functions Security Definitions (Sample)".

Select the appropriate step to create the sample job by ICE. After the job is created, you can enter and save your definitions in the INSTWORK library.

Step 6.2 Security Definitions (Sample)

To define CTMAS security, edit the ECSSRAC2 member in the IOA INSTWORK library to perform the following actions.

CTMAS security uses the IOASE07 and IOASE32 IOA security modules, as well as the CTMSE08 Control-M security module, when IOA and Control-M security interfaces are installed. Therefore, to complete CTMAS security, only the required definitions are necessary.

1. Define entities and user authorizations to RACF.

   For details about entities and user authorizations, see the Protected Elements tables (Table 23 on page 56, and Table 24 on page 57).

2. To authorize USERA (the user ID of the Control-M/Enterprise Manager installer) access to a given CTMAS entity, use the following command:

   PERMIT $$ECSnnn.qname CLASS(FACILITY) ID(USERA) ACCESS(READ)

   where ECSnnn is the name of the CTMAS entity to be accessed.

3. Change USERA to the user ID of the CTMAS installer.

   All entity names for each CTMAS protected element are described in Table 23 on page 56, for Basic Definition mode, and in Table 24 on page 57, for Extended Definition mode.

4. Submit the job for execution.

   This job must be run under a user of a RACF administrator who has authorization to enter these RACF commands.

5. Scan the output of the job for information and error messages produced by RACF.

For samples of user authorizations, review members ECSSRAC3, IOASRAC3 and CTMSRAC3 in the IOA INSTWORK library.

Step 6.3 Functions Security Definitions (Sample)

Select this step to edit the ECSSRAC3 member in the IOA INSTWORK library. This member contains a sample of the various definitions required to define access authorizations to various CTMAS entities. Review the definitions and modify to meet your site's requirements.

Step 7. Control-M Application Server - TopSecret

Step 7.1 Grant Access Permission

Collect the required data in order to define the INCONTROL entities and user authorizations to the security product.

You can use this data in the sample jobs provided in the subsequent steps "Control-M Application Server Security Definitions (Sample)" and "Functions Security Definitions (Sample)".

Select the appropriate step to create the sample job by ICE. After the job is created, you can enter and save your definitions in the INSTWORK library.

Step 7.2 Security Definitions (Sample)

To define CTMAS security, edit the ECSSTSS2 member in the IOA INSTWORK library and customize it as follows.

CTMAS security uses the IOASE07 and IOASE32 IOA security modules, as well as the CTMSE08 Control-M security module, when IOA and Control-M security interfaces are installed. Therefore, to complete CTMAS security, only the required definitions are necessary.

1. Define CTMAS ACID to TopSecret.
   - **a.** Change the value of parameter DEPT from sec-administrator-dept to the appropriate ACID:

     TSS CRE (CTMAS) NAME (...) DEPT(sec-administrator-dept)
   - **b.** Define the CTMAS started task to TopSecret.

     Change the ACID definition in the following command to the appropriate ACID:

     TSS ADD(STC) PROC(CTMAS) ACID(CTMAS)
   - **c.** Allow CTMAS ACID to access IOA datasets.

     Authorizations to access IOA datasets are optionally defined during the IOA installation process. This step must be completed before proceeding with security implementation. For information about how to grant users access to IOA datasets, see the IOA Installation chapter of the *INCONTROL for z/OS Installation Guide: Installing*.
   - **d.** Connect the appropriate profile to the CTMAS ACID in the following command:

     TSS ADD (CTMAS) PROF (profile-name)
2. Give CTMAS READ access authority to any datasets that are accessed by workstation users.
   - **a.** Define IOA entities and user authorizations to TopSecret

     For information about how to define Control-M/Enterprise Manager entities and user authorizations to TopSecret, see Basic Definition Security Calls (on page 36), and Extended Definition Security Calls (on page 37).
   - **b.** Modify the following command to establish ownership of the resources in TopSecret to the appropriate owner:

     TSS ADD(sec-administrator-dept) IBMFAC($$ECS)

   For samples of user authorizations, see member ECSSTSS3 in the IOA INSTWORK library.

   All entity names for each CTMAS protected element are described in CTMAS Basic Definition Security Calls (on page 47) , for Basic Definition mode and in CTMAS Extended Definition Security Calls (on page 48), for Extended Definition mode.
   - **c.** Authorize the CTMAS installer to use CTMAS facilities.

     Customize the following command to authorize USERA access to the Online monitor:

     TSS ADD(USERA) FACILITY(CTW)
   - **d.** Modify USERA to the user ID of the CTMAS installer.

     Customize the following command to authorize the CTMAS installer to use CTMAS facilities:

     TSS PERMIT(USERA) IBMFAC($$ECS) ACC(READ)
3. Submit the job

   Submit the job and verify that all steps complete with a condition code of zero. Run this job under the ACID of the general security administrator (SCA) who has authorization to enter these TopSecret commands.

4. Verify that all job steps end with a condition code of 4 or less.

Step 7.3 Functions Security Definitions (Sample)

Select this step to edit the ECSSTSS3 member in the IOA INSTWORK library. This member contains a sample of the various definitions required to define access authorizations to various CTMAS entities. Review the definitions and modify to meet your site's requirements.

Step 8. Control-M Application Server - ACF2

Step 8.1 Grant Access Permission

Collect the required data in order to define the INCONTROL entities and user authorizations to the security product.

You can use this data in the sample jobs provided in the subsequent steps "Control-M Application Server Security Definitions (Sample)" and "Functions Security Definitions (Sample)".

Select the appropriate step to create the sample job by ICE. After the job is created, you can enter and save your definitions in the INSTWORK library.

Step 8.2 Security Definitions (Sample)

Define CTMAS security in the following steps.

CTMAS security uses the IOASE07 and IOASE32 IOA security modules, as well as the CTMSE08 Control-M security module, when IOA and Control-M security interfaces are installed. Therefore, to complete CTMAS security, only the required definitions are necessary.

1. Define a CTMAS started task

   Define a Logon ID for the CTMAS started task with a multi-user address space (MUSSAS) parameter.

2. Define entities and user authorizations to ACF2/SAF

   a. Edit member ECSSSAF2 in the IOA INSTWORK library. For details about entities and user authorizations, see CTMAS Basic Definition Security Calls (on page 47), and CTMAS Extended Definition Security Calls (on page 48).

   b. Authorize USERA (the user ID of the Control-M installer) access to a given CTMAS entity, use the following command:

      ```
      SET RESOURCE(CMF)
      COMP
      $KEY($$ECSnnn.qname)
       UID(USERA) ALLOW
      ```

      where ECSnnn is the name of the CTMAS entity to be accessed.

   c. Change USERA to the UID string of the CTMAS installer.

      All entity names for each CTMAS protected element are described in CTMAS Basic Definition Security Calls (on page 47), for Basic Definition mode, and in CTMAS Extended Definition Security Calls (on page 48), for Extended Definition mode.

   For samples of user authorizations, review member ECSSSAF3 in the IOA INSTWORK library.

3. Submit the job

Verify that all job steps complete with a condition code of zero.

This job must be run under a user of an ACF2 administrator who is authorized to enter these ACF2 commands.

Scan the output of the job for information and error messages produced by ACF2. All job steps must end with a condition code of 4 or less.

Step 8.3 Functions Security Definitions (Sample)

Select this step to edit the ECSSSAF3 member in the IOA INSTWORK library. This member contains a sample of the various definitions required to define access authorizations to various CTMAS entities. Review the definitions and modify to meet your site's requirements.

Security Interface Modules

This section describes the IOASE07, IOASE32, and CTMSE08 security modules in detail.

Module IOASE07

The IOASE07 module is the security module of IOA Exit IOAX007. This module verifies the user's authorization to add, delete, or change prerequisite conditions, Control resources and Quantitative resources.

The CLASS checked is FACILITY; the entity used to check authorization depends on whether Basic or Extended definition modes are used:

Basic Definition Mode

The entity used to check authorization is $$IOARES.qname.name, where name is the resource name or the condition name.

Extended Definition Mode

The entity built for verification depends on what screen is being used.

Table 25 Screen 4—IOA Conditions/Resources Screen

	Action	Entity
For prerequisite conditions:	ADD	$$ADDCND.qname.condition-name
	DEL	$$DELCND.qname.condition-name

	Action	Entity
For quantitative resources:	ADD	$$ADDRES.qname.resource-name
	DEL	$$DELRES.qname.resource-name
	CHANGE	$$CHARES.qname.resource-name
	CHECK	$$CHKRES.qname.resource-name
For control resources:	ADD	$$ADDCTL.qname.control-name
	DEL	$$DELCTL.qname.control-name
	CHECK	$$CHKCTL.qname.control-name

Table 26 Screen 7 – IOA Manual Conditions Screen

Action	Entity
NEW (add)	$$NEWCND.qname.condition-name
ERASE (delete)	$$ERACND.qname.condition-name

Module IOASE32

The IOASE32 module is the security module of IOA user Exit IOAX032. This module verifies that the user is authorized to Edit or View JCL members, documentation members, tables or calendars from the CTMAS workstation.

Basic Definition Mode

The CLASS checked is DATASET; the entity built is the dataset name of the library.

The access level used to check this authorization depends on user request:

- SAVE request: update
- VIEW request: read
- EDIT request: read

Extended Definition Mode

Two checks are performed:

1. Dataset access

   The CLASS checked is DATASET. The entity used to check authorization is: the dataset name of the library. The access level used to check this authorization depends on the user request:

   SAVE request: update

   VIEW request: read

   EDIT request: read

2. Operations

   Under IOA, the CLASS checked is FACILITY. The entity used to check authorization depends on the user request:

Table 27 Request Authorization Entity

Request	Entity
EDITJCL	$$ECSEDJ.qname.member
SAVEJCL	$$ECSSVJ.qname.member
VIEWJCL	$$ECSVWJ.qname.member
EDITDOC	$$ECSEDD.qname.member
SAVEDOC	$$ECSSVD.qname.member
VIEWDOC	$$ECSVWD.qname.member
EDITDEF	$$ECSEDF.qname.member
SAVEDEF	$$ECSSVF.qname.member
VIEF	$$ECSVWF.qname.member
DELETTB	$$ECSTTB.qname.table
DELETCL	$$ECSTCL.qname.cal

Module CTMSE08

The CTMSE08 Control-M security module verifies that the user is authorized to perform actions (for example, hold, delete, rerun) on jobs displayed in the Active Environment screen (Screen 3).

Basic Definition Mode

Initial Access to Screen 3

IOASECUR is called to issue a security check for authorization. The CLASS checked is FACILITY; the entity checked is $$CTMPNL3.qname.

Subsequent Operations in Screen 3

For all actions (for example, hold, delete, rerun) that are performed on this screen, the IOASECUR module is called to issue a security check for authorization. The CLASS and entity checked is:

For RACF:

```
SURROGAT
 owner.SUBMIT
```

For TopSecret:

```
ACIDCHK
 owner
```

For ACF2/SAF:

```
FACILITY
 $SUBMIT.owner
```

The check verifies that the current user who has the authority to submit jobs with a USER parameter is equal to that of the specific job being accessed. A user who is authorized to submit a job on behalf of others is also authorized to perform the specific action (for example, hold, delete, rerun) on jobs belonging to other users.

Extended Definition Mode

Initial access to Screen 3

IOASECUR is called to issue a security check for authorization. The CLASS checked is FACILITY; the entity checked is $$CTMPNL3.qname.

Subsequent Operations in Screen 3

The actions (for example, hold, delete, rerun, and so on) are separated into different categories of access authority to the Active Environment screen (Screen 3).

The CLASS checked is FACILITY. The entity checked is $$JOBxrrr.qname.owner

where

- owner is the owner specified in the Job Scheduling Definition screen (Screen 2)
- x is the one-digit action identifier
- rrr is the three-character identifier for each action, described in the following table:

Table 28 Action Identifiers

Action Identifier	Action	Description
1	ACT LOG SYS STA ZOO AES	Activate Log View sysout View statistics Zoom AutoEdit simulation
2	CNF FOK FRE HLD RRN RRN	Confirm Force OK Free Hold Rerun Restore
3	CHA PRI DEL EDI KIL	Change Change priority Delete, Undelete Edit JCL Kill an executing job

To permit USERA to hold jobs with an owner of USERB, use the following command:

RACF Security

```
PERMIT $$JOB2HLD.qname.USERB ACCESS(READ) ID(USERA) CLASS(FACILITY)
```

TopSecret Security

```
TSS PERMIT(USERA) ISMFAC($$JOB2HLD.qname.UDERB) ACC(READ)
```

ACF2/SAF Security

```
SET RESOURCE(CMF)
COMP
$KEY($$JOB2HLD.qname.USERB) TYPE(CMF)
 UID(USERA) ALLOW
```

2

Control-M Security

This chapter describes the procedure used to implement the Control-M security interface. Review the explanations below about the elements that are protected in Control-M and then proceed to the step-by-step instructions.

Protecting Control-M Elements

The Control-M security interface protects the following Control-M elements:

- Job ordering.
- Access to JCL libraries.
- Job submission.
- Access to and use of the Job Status screen (Screen 3).
- Ordering of Control-M Event Manager (CMEM) rules, and use of DO commands and ON blocks.

To control user authorizations to each Control-M protected element, choose either the Basic Definition mode or the Extended Definition mode. For more information on Basic and Extended Definition modes, see 1 IOA Security.

Job Ordering

Each Control-M job definition contains an OWNER parameter. This parameter, which must be defined as a valid security product user ID, specifies the user ID to which the definition belongs. When ordering a job, the user must have the authorization to access the owner specified in the job definition. The CTMSE01 Control-M security module verifies that the user who orders a job is authorized to do so, using the owner parameter of the job. If the user who orders a job is the owner specified in the job definition, no security checks are performed.

Access to JCL Libraries

Before Control-M submits a job, the CTMSE02 security module verifies the job definition owner’s authority to read the JCL library that is specified in the job definition.

In addition, during OPEN processing, the operating system data management routines check whether or not the user ID of the address space is authorized to read the JCL library. BMC therefore recommends adding the RACF OPERATIONS attribute (or equivalent, for other security products) to the Control-M monitor user ID to reduce security checking overhead.

Job Submission

When Control-M submits a job, the CTMSE02 Control-M security module performs a check to verify that the job is submitted with a valid USER parameter in the job statement or a valid //*JOBFROM statement for ACF2/SAF.

If the job statement does not contain a USER parameter or a valid //*JOBFROM statement for ACF2/SAF, the USER parameter (set to the value owner), or a valid //*JOBFROM statement for ACF2/SAF, is added to the job statement, if required.

If the job statement contains a USER parameter or a valid //*JOBFROM statement for ACF2/SAF, the CTMSE02 Control-M security module either allows or fails the submission of a job, depending on implementation options and the owner authority of the user ID defined for the user who submitted the job specified in the JCL.

In addition, the CTMSE02 security module determines whether the user ID assigned to the Control-M monitor is authorized to submit jobs on behalf of the user ID assigned to the submitted job. If it is not authorized, the submission fails.

For more information, see the description of the CTMSE02 security module, in Module CTMSE02 (on page 77).

Access to and Use of the Status Screen

The Status screen (screen 3) lists the active jobs currently handled by Control-M and their status. The user can issue inquiries about a job within the list or change a job's status. The CTMSE08 Control-M security module verifies the user's authorization to enter Screen 3 and perform actions (for example, hold, delete) on jobs displayed in the Status screen.

For more information, see Control-M Basic Definition Security Calls (on page 60), Control-M Extended Definition Security Calls (on page 62), and Control-M Security Modules (on page 76).

Control-M Basic Definition Security Calls

Table 29 Control-M Basic Definition Security Calls

Protected Element	Type	Class Entity Name	Explanation	Security Module
Controlling Job Orders				
Order a job	all	SURROGAT owner.SUBMIT ACIDCHK owner FACILITY $SUBMIT.owner	owner is the name of the user specified in the job scheduling definition.	CTMSE01

Protected Element	Type	Class Entity Name	Explanation	Security Module
JOBDSNa security check	all	FACILITY $$REGSTR.qname	qname is the name used to assign different authorizations to various Control-M environments (for example, Test and Production).	CTMSE01
Order a started task	all	FACILITY $$CTMSTC.qname.stcname	stcname is the name of the started task.	CTMSE01
Controlling Job Submissions				
Access JCL library	all	DATASET dataset	dataset is the name of the JCL library.	CTMSE02
Starting a started task	all	No check is performed in Basic Definition mode		CTMSE02
Submitting a job	all	■ If the job statement does not contain parameter USER, parameter USER is added to the job statement and set to the value of owner.b ■ The submission fails if all the following statements are true: ■ the JCL job statement contains the USER= parameter ■ the owner ID of the job definition is not the same as the value specified in the USER= parameter or //*JOBFROM (for ACF2 users) ■ the MSUBCHK is set to No ■ If the USER= parameter exists and parameter MSUBCHK is set to Y (Yes), the class checked is [SURROGAT | ACIDCHK | FACILITY] and the entity checked is [userid.SUBMIT | | $SUBMIT.userid]c	owner is the name of the user specified in the job scheduling definition.	CTMSE02

Protected Element	Type	Class Entity Name	Explanation	Security Module
Controlling Access to the Active Environment Screen				
Accessing the Active Environment Screen	all	FACILITY $$CTMPNL3.qname		CTMSE08
Performing operations in the Active Environment screen	all	SURROGAT owner.SUBMIT ACIDCHK owner FACILITY $SUBMIT.owner	owner is the owner specified in the job scheduling definition.	CTMSE08
Performing Refresh commands in the Job Dependency Network screen	all	FACILITY REFRESH NET $$REFNET.qname REFRESH PROPAGATE $$REFPROP.qname REFRESH DEADLINE $$REFDEAD.qname REFRESH ALL $$REFALL.qname		CTMSE08

Control-M Extended Definition Security Calls

Table 30 Control-M Extended Definition Security Calls

Protected Element	Type	Class Entity Name	Explanation	Security Module
Controlling Job Orders				
Order a job	all	FACILITY $$JOBORD.qname.owner	qname is the name used to assign different authorizations to various Control-M environments (for example, Test and Production). owner is the name of the user specified in the job scheduling definition.	CTMSE01

Protected Element	Type	Class Entity Name	Explanation	Security Module
JOBDSNa security check	all	FACILITY $$REGSTR.qname		CTMSE01
Order a started task	all	FACILITY $$STCORD.qname.stcname	stcname is the name of the started task.	CTMSE01
Controlling Job Submissions				
Access JCL library	all	DATASET dataset	dataset is the name of the JCL library.	CTMSE02
Starting a started task	all	FACILITY $$STRSTC.qname.stcname	stcname is the name of the started task.	CTMSE02
Submitting a job	all	▪ If the job statement does not contain parameter USER, parameter USER is added to the job statement and set to value owner.b ▪ The submission fails if all the following statements are true: ▪ the JCL job statement contains the USER= parameter ▪ the owner ID of the job definition is not the same as the value specified in the USER= parameter or //*JOBFROM (for ACF2 users) ▪ the MSUBCHK is set to No ▪ If parameter USER exists and parameter MSUBCHK is set to Y (Yes), the class checked is [SURROGAT | ACIDCHK | FACILITY] and the entity checked is [userid.SUBMIT | | $SUBMIT.userid]c	owner is the name of the user specified in the job scheduling definition.	CTMSE02
Controlling Access to the Active Environment Screen				
Accessing the Active Environment Screen	all	FACILITY $$CTMPNL3.qname		CTMSE08

Protected Element	Type	Class Entity Name	Explanation	Security Module
Performing Operations in the Active Environment Screen	all	FACILITY React: $$JOB1ACT.qname.owner Browse: $$JOB1SYS.qname.owner Stats: $$JOB1STA.qname.owner Zoom: $$JOB1ZOO.qname.owner Log: $$JOB1LOG.qname.owner AES: $$JOB1AES.qname.owner Hold: $$JOB2HLD.qname.owner Free: $$JOB2FRE.qname.owner Force: $$JOB2FOK.qname.owner Rerun: $$JOB2RRN.qname.owner Restore: $$JOB2RST.qname.owner Confirm: $$JOB2CNF.qname.owner Change: $$JOB3CHA.qname.owner Bypass: $$JOB3BYP.qname.owner Priority: $$JOB3PRI.qname.owner Delete: $$JOB3DEL.qname.owner Undelete: $$JOB3DEL.qname.owner Edit JCL: $$JOB3EDI.qname.owner Kill: $$JOB3KIL.qname.owner	owner is the owner specified in the job scheduling definition.	CTMSE08
Performing Refresh commands in the Job Dependency Network screen	all	FACILITY REFRESH NET $$REFNET.qname Refresh Propagate: $$REFPROP.qname REFRESH DEADLINE: $$REFDEAD.qname REFRESH ALL: $$REFALL.qname		

Implementing Control-M Security

This section describes the steps required to implement the Control-M security interface.

The Control-M security interface can be installed either as part of the customized installation path, or during the Customization process after installation. Both options use the INCONTROL Installation and Customization Engine (ICE) application. If you are not familiar with the ICE interface, see the *INCONTROL for z/OS Installation Guide: Installing*. Perform all the steps required to implement Control-M security at your site.

If CMEM is installed at your site, perform all the steps required to implement CMEM security at your site.

➢ To install the Control-M security interface

1. Enter the main ICE screen.
2. Select Customization.
3. Enter CTM in the Product field.
4. Select Security Customization.
5. Perform all major and minor steps required to install the security product.

Step 1. Implement Control-M Security

Use the following steps that correspond to the installation steps in ICE, to implement Control-M security.

Step 1.1 Grant Access Permissions

Collect the required data to define the INCONTROL entities and user authorizations to the security product.

You can use this data in the sample jobs provided in subsequent steps "Control-M Security Definitions (Sample)" and "Functions Security Definitions (Sample)".

Select the appropriate step to create the sample job by ICE. After the job is created, enter your definitions and save them in the INSTWORK library.

Step 1.2 Customize Security Parameters

Use ICE to define the following parameters:

Table 31 Parameter Definitions

Parameter	Description
DEFMCHKM	When choosing a definition mode as COND to any of the Control-M security modules, use qname together with the value given to this parameter as the high level qualifier, to determine the real definition mode to be used.
LIFETIME	This parameter determines whether a security cache is used during the job submission process. This cache allows checking the authorization against a cache instead of the security system, resulting in improved performance. The cache is created by the exit CTMSE02. The value specified for this parameter defines how frequent the cache is refreshed. The value is specified in minutes. The valid range of values is from 0 to 1440. Default: 0 - meaning that no cache is used.
SECTOLM	This parameter determines the action to perform if your security product is inactive or a specific resource is not defined in the security product. Valid values are: ▪ YES — Perform the action. ▪ NO — Do not perform the action.

Parameter	Description
MSUBCHK	This parameter determines whether Control-M submits jobs that already contain the USER parameter or //*JOBFROM statement in the job card. Valid values are: ▪ YES — If Control-M attempts to submit a job and the job statement already contains the USER parameter or //*JOBFROM, check the job definition owner's authority to the JCL USER. Default. ▪ NO — Reject the submission if the JCL JOB statement contains the USER parameter, and the owner ID of the job definition is not the same as the value specified in parameter USER or //*JOBFROM (for ACF2 users).
PROTAUTO	This parameter protects the AUTO command. Valid values are: ▪ YES — Users need permission to use the AUTO command. ▪ NO — The AUTO command is unrestricted. Default. The AUTO command allows you to put certain screens into 'AutoRefresh Mode'. If you set PROTAUTO=Y, then Users need permission ($$CTMAUTO.qname) to enter AutoRefresh Mode and CTM Security Exit 8 (CTMSE08) will check for it. Otherwise, the AUTO command is unrestricted. Some customers prefer to protect it, since AutoRefresh can use a lot of cycles, and some Users have a tendency to leave it active.

Table 32 Job Card Parameters

Parameter	Description
RACJCARD	For RACF. This parameter determines whether Control-M adds USER and GROUP parameters to submitted jobs if they do not exist. Valid values are: ▪ U — Add a USER parameter to the submitted job card. ▪ G — Add both USER and GROUP parameters to submitted jobs, where the GROUP is the RACF default group of the user. ▪ N — Do not add USER or GROUP parameters.
TSSJCARD	For TopSecret. This parameter determines whether Control-M adds the USER parameter to submitted jobs if it does not exist. Valid values are: ▪ U — Add the USER parameter to the submitted job card. ▪ N — Do not add the USER parameter.

Parameter	Description
SAFJCARD	For ACF2. This parameter determines whether Control-M adds the USER parameter or //*JOBFROM statement to submitted jobs if they do not exist. Valid values are: ▪ U — Add the USER parameter to the submitted job statement. ▪ J — Add a //*JOBFROM statement to the submitted job. ▪ L – Add a //*LOGONID statement to the submitted job. ▪ S — Add a //*JOBFROM userid/ctm-stc-name statement to the submitted job. ▪ N — Do not add the USER parameter or //*JOBFROM statement.

Table 33 Mode Definition

Mode	Description
Mode Definition	Definition mode for the Control-M security modules. Valid values are: ▪ COND — Conditional Definition mode. Default. ▪ BASIC — Basic Definition mode. ▪ EXTEND — Extended Definition mode.
DFMM01	Definition mode for the CTMSE01 Control-M security module.
DFMM02	Definition mode for the CTMSE02 Control-M security module.
DFMM08	Definition mode for the CTMSE08 Control-M security module.

Step 1.3 Save Security Parameters into Product

This step saves all the security parameters specified for Control-M. When completed, the Status column is automatically updated to COMPLETE.

Step 2. RACF Security Definition Samples

Step 2.1 Control-M Security Definitions

Select this step to edit the CTMSRAC2 member in the IOA INSTWORK library.

Perform the following steps to define the required users permissions:

1. To define the entity $$CTMEDM.qname to RACF, use the following RACF command:

   ```
   RDEFINE FACILITY $$CTMEDM.qname UACC(NONE)
   ```

2. To associate USERA with Extended Definition mode, use the following RACF command:

   PERMIT $$CTMEDM.qname ID(USERA) CLASS(FACILITY) ACCESS(READ)

   If the definition mode to a Control-M security module was defined as conditional mode (COND), and a user does not have access to this entity, the user is set to work in Basic Definition mode. Otherwise, the user is set to work in Extended Definition mode.

3. Submit the job for execution.

   This job must be run under a user who has authorization to enter these RACF commands.

4. Scan the output of the job for information and error messages produced by RACF.

Step 2.2 Function Security Definitions (Optional)

Select this step to edit the CTMSRAC3 member in the IOA INSTWORK library. This member contains a sample of the various definitions required to define access authorizations to various Control-M entities. Review the definitions and modify to meet your site's requirements.

Step 2.3 Control Program Access to Datasets (Optional)

BMC Software recommends that, before selecting this step, the security administrator first read Limiting Access to Specific Programs (on page 200) and the *IBM Resource Access Control Facility Security Administrator's Guide*.

Select this step to edit the CTMSRAC4 member in the IOA INSTWORK library. This member contains a sample of the definitions required to define Program Pathing access authorizations to Control-M datasets. Review the definitions and modify to meet your site's requirements.

Step 3. TopSecret Security Definition Samples

Step 3.1 Control-M Security Definitions

Select this step to edit the CTMSTSS2 member in the IOA INSTWORK library.

Perform the following steps to define the required permissions:

1. Define Control-M in the TopSecret Facility Matrix.

   a. Modify USER2 in the Facility definition command to a free entry in the Facility Matrix, as follows:

      TSS MODIFY FAC(USER2=NAME=CTM)

      This command defines Control-M in the Facility Matrix until the next IPL.

   b. Update the TopSecret parameter member (usually called TSSPARM0) to permanently define Control-M.

2. Define Control-M ACID in TopSecret.

   Change the DEPT parameter value from sec-administrator-dept to the appropriate ACID:

   ```
   TSS CRE(CONTROLM) NAME (...) DEPT(sec-administrator-dept)
   ```

3. Define Control-M started tasks in TopSecret.

   Change the ACID definition in the following commands to the appropriate ACID:

   ```
   TSS ADD(STC) PROC(CONTROLM) ACID(CONTROLM)
   TSS ADD(STC) PROC(CONTDAY) ACID(CONTROLM)
   ```

4. Allow Control-M ACID to access Control-M datasets.

   Optionally, you can define authorizations to access Control-M datasets during Control-M installation. Complete this step before proceeding with security implementation. For information about how to grant users access to Control-M datasets, see the IOA Installation chapter in the *INCONTROL for z/OS Installation Guide: Installing*.

   Connect the appropriate profile to the Control-M ACID in the following command:

   ```
   TSS ADD(CONTROLM) PROF (profile-name)
   ```

Allow READ access authorization to any Control-M JCL libraries used to submit jobs.

5. Authorize Control-M ACID to submit jobs for other users, with the following command:

   ```
   TSS ADD(CONTROLM) NOSUBCHK
   ```

6. Define Control-M entities and user authorizations to TopSecret.

   For information about how to define Control-M entities and user authorizations to TopSecret, see Control-M Basic Definition Security Calls (on page 60), and Control-M Extended Definition Security Calls (on page 62).

   Modify the following command to establish ownership of the resources in TopSecret to the appropriate owner:

   ```
   TSS ADD(sec-administrator-dept) IBMFAC($$CTM)
   ```

For samples of user authorizations, see member CTMSTSS3 in the IOA INSTWORK library.

Entity names for Control-M protected elements appear in Control-M Basic Definition Security Calls (on page 60) for Basic Definition mode and in Control-M Extended Definition Security Calls (on page 62) for Extended Definition mode.

7. Associate users with Extended Definition modes.

   a. Modify the following TopSecret command to establish Extended Definition mode for the Control-M installer.

      ```
      TSS PERMIT (USERA) IBMFAC($$CTMEDM.qname) ACC(READ)
      ```

   b. Change USERA to the UID of Control-M installer.

      A user with access to this entity is set to work in Extended Definition mode. The user without access is set to work in Basic Definition mode.

If the definition mode to a Control-M security module was defined as COND, and does not have access to this entity, the user is set to work in Basic Definition mode. Otherwise, the user is set to work in Extended Definition mode.

8. Authorize the Control-M installer to use Control-M facilities.

   a. Customize the following command to authorize USERA access to Control-M:

      ```
      TSS ADD(USERA) IBMFAC($$CTM)
      ```

   b. Change USERA to the user ID of the Control-M installer.

   c. Customize the following command to authorize the Control-M installer to use Control-M facilities:

      ```
      TSS PERMIT(USERA) IBMFAC($$CTM) ACC(READ)
      ```

9. Submit the job.

Run this job under the ACID of the general security administrator (SCA) who has authorization to enter TopSecret commands.

Step 3.2 Function Security Definitions (Optional)

The IOASRAC3 job in the IOA INSTWORK library is optional. It contains some definition samples for various entities. Customize this job according to your requirements and submit the job.

Define entities and user authorizations.

For information about defining IOA entities and user authorizations, see Control-M Basic Definition Security Calls (on page 60), and Control-M Extended Definition Security Calls (on page 62).

To control access to the IOA Online facility, specify the following command:

```
RDEFINE FACILITY $$IOAONLINE.qname
```

where qname is used to assign different authorizations to different IOA environments (such as Test and Production). This parameter is specified during IOA installation.

To define and authorize all conditions beginning with SYS, use the following command:

```
RDEFINE FACILITY $$IOARES.qname.SYS*
PERMIT $$IOARES.qname.SYS* CLASS(FACILITY) ID(USERA) ACCESS(READ)
```

To authorize USERA access to a given IOA entity, use the following command:

```
PERMIT $$IOAnnn.qname CLASS(FACILITY) ID(USERA) ACCESS(READ)
```

All entity names for each IOA protected element appear in Control-M Basic Definition Security Calls (on page 60) for Basic Definition mode and Control-M Extended Definition Security Calls (on page 62), for Extended Definition mode.

Step 3.3 Control Program Access to Datasets (Optional)

BMC Software recommends that, before selecting this step, the security administrator first read Limiting Access to Specific Programs (on page 200) and the *IBM Resource Access Control Facility Security Administrator's Guide.*

Select this step to edit the CTMSTSS4 member in the IOA INSTWORK library. This member contains a sample of the definitions required to define Program Pathing access authorizations to Control-M datasets. Review the definitions and modify to meet your site's requirements.

Step 3.4 Define CTM to TopSecret Facility Matrix (Optional)

Select this step to edit the CTMSTSS5 member in the IOA INSTWORK library.

Perform the following steps to define Control-M in the TopSecret Facility Matrix:

10. Modify USER2 in the Facility definition command to a free entry in the Facility Matrix, with the following command:

TSS MODIFY FAC(USER2=NAME=CTM)

11. Copy modified member CTMSTSS5 into TSSPARM0.

Step 4. ACF2 Security Definition Samples

Step 4.1 Control-M Security Definitions

Select this step to edit the CTMSSAF2 member in the IOA INSTWORK library.

1. Associating users with Extended Definition mode.

   Add the following ACF2 commands to define the $$CTMEDM.qname entity to ACF2/SAF and authorize users to this entity:

   ```
   SET RESOURCE(CMF)
   COMP
   $KEY($$CTMEDM.qname) TYPE(CMF)
   UID(USERA) ALLOW
   ```

   If the definition mode to a Control-M security module was defined as COND, and does not have access to this entity, the user is set to work in Basic Definition mode. Otherwise, the user is set to work in Extended Definition mode.

2. Define entities and user authorizations to CA-ACF2/SAF.

   For more information about entities and user authorizations, see Control-M Basic Definition Security Calls (on page 60) and Control-M Extended Definition Security Calls (on page 62).

To define and authorize the resource profile in Basic Definition mode to protect ordering of STCs beginning with SYS, specify the following command:

```
SET RESOURCE(CMF)
COMP
$KEY($$CTMSTC.qname.SYS**************************)
 UID(USERA) ALLOW
```

where qname is the name used to assign different authorizations to different IOA environments (such as Test and Production). This parameter is specified during IOA installation.

To authorize USERA access to a given Control-M entity, use the following command:

```
SET RESOURCE(CMF)
COMP
$KEY($$CTMnnn.qname)
 UID(USERA) ALLOW
```

where CTMnnn is the entity name of the Control-M protected element described in Control-M Basic Definition Security Calls (on page 60) for Basic Definition mode and in Control-M Extended Definition Security Calls (on page 62)for Extended Definition mode.

For samples of user authorizations, review member CTMSSAF3 in the IOA INSTWORK library.

3. Submit the job.

   Run this job with a user who has authorization to enter these ACF2 commands.

   Scan the job output for information and error messages produced by ACF2.

4. Rebuild resource type CMF rules.

   Rebuild the resource type CMF rules by issuing the following MVS command:

   F ACF2,REBUILD(CMF)

Step 4.2 Function Security Definitions (Optional)

The IOASTSS3 job in the IOA INSTWORK library is optional. It contains some definition samples for various entities. Customize this job according to your requirements and submit this job.

Step 4.3 Control Program Access to Datasets (Optional)

BMC Software recommends that, before selecting this step, the security administrator first read Limiting Access to Specific Programs (on page 200) and the *IBM Resource Access Control Facility Security Administrator's Guide.*

Select this step to edit the CTMSSAF4 member in the IOA INSTWORK library. This member contains a sample of the definitions required to define Program Pathing access authorizations to Control-M datasets. Review the definitions and modify to meet your site's requirements.

Control-M Event Manager Security

The Control-O security interface protects the following Control-M Event Manager (CMEM) elements:

- The CTOSE01 security module protects CMEM rule ordering.
- The CTOSE02 security module protects the use of DO statements and ON blocks that access or modify restricted IOA prerequisite conditions.

When Control-O is installed, the Control-O monitor assumes that a CMEM monitor is functional. Control-O security modules provide a migration path from a CMEM monitor to a Control-O monitor to implement CMEM security.

If Control-O security is already implemented or is going to be implemented, do not implement CMEM security.

Rule Ordering

Each CMEM rule is defined with an owner, which is the name of a user ID to which this rule belongs. To order a rule, the user must have the authorization to access the owner of the rule. The CTOSE01 security module verifies that the current user has the authorization to order the rule, using the OWNER field of the rule.

The CMEM default rules in the IOACMEMR table are provided with the OWNER user ID of IOADMIN.

You must grant the user who orders these rules (either CTMCMEM or CONTROLO) permission to load the rules on behalf of the IOADMIN user ID, and grant the IOADMIN user permissions to the perform ON and DO statements in these rules.

Authority to Use Rule Functions (DO Statements and ON Statements)

CMEM rules react to events defined in the ON statements of the rule and actions defined in the DO statement of the rule. The security interface verifies if these actions are permitted to the owner of the rule. Before the rule is loaded, the CTOSE02 security module performs an authority check for each rule statement. If one of the authority checks fails, the entire rule load is canceled.

CMEM Basic Definition Security Calls

Table 34 CMEM Basic Definition Security Calls

Protected Element	Type	Class Entity Name	Explanation	Security Module
Controlling Rule Ordering	all	SURROGAT *owner*.SUBMIT ACIDCHK *owner* FACILITY $SUBMIT.*owner*	*owner* is the owner of the rule.	CTOSE01
Controlling use of Control-O commands	all	FACILITY ON JOBARRIV $$CTOJAR.*qname.jobname* ON JOBEND $$CTOJED.*qname.jobname* ON DSNEVENT $$CTODSN.*qname.jobname* ON STEP $$$CTOSTP.*qname.jobname* DO COND or RESOURCE $$IOARES.*qname.resource-name* DO FORCEJOB $$CTOCMO.*qname.lib-name.table*	*jobname* is the name of the job specified in the ON statement. *resource-name* is the name of the resource specified in the DO statement. *lib-name* is the first 21 characters of the Control-M schedule library. *table* is the member name in the Control-M schedule library. The whole entity name is truncated by RACF to 39. This means that *table* will be entirely truncated unless *lib-name* is less than 21.	CTOSE02
	all	DO STOPJOB $$CTOJST.*qname*		CTOSE02
	all	RUNSTEC *field* $$CTORTS.*qname.runtime-sec*	*runtime-sec* is the value of the rule RUNTSEC parameter.	CTOSE02

CMEM Extended Definition Security Calls

Table 35 CMEM Extended Definition Security Calls

Protected Element	Type	Class Entity Name	Explanation	Security Module
Controlling Rule Ordering		FACILITY $$CTOORD.*qname.owner*	*owner* is the owner of the rule.	CTOSE01
Controlling use of Control-O commands		FACILITY ON JOBARRIV $$CTOJAR.*qname.jobname* ON JOBEND $$CTOJED.*qname.jobname* ON DSNEVENT $$CTODSN.*qname.jobname* ON STEP $$$CTOSTP.*qname.jobname* DO COND or RESOURCE $$CTORES.*qname.resource-name* DO FORCEJOB $$CTOCMO.*qname.lib-name.table*	*jobname* is the name of the job specified in the ON statement. *resource-name* is the name of the resource specified in the DO statement. *lib-name* is the first 21 characters of the Control-M schedule library, and *table* is the member name in the Control-M schedule library. The whole entity name is truncated by RACF to 39. This means that *table* will be entirely truncated unless *lib-name* is less than 21.	CTOSE02
		DO STOPJOB $$CTOJST.*qname*		CTOSE02
		RUNSTEC *field* $$CTORTS.*qname.runtime-sec*	*runtime-sec* is the value of the rule RUNTSEC parameter.	CTOSE02

Implementing CMEM Security

CMEM security implementation is an optional step performed during Control-M security implementation.

If CMEM is installed, you must implement CMEM security after completing the Control-M security implementation steps.

This section details the steps required to implement the CMEM security interface using the ICE application. If you are not familiar with the ICE interface, see the ICE chapter in the *INCONTROL for z/OS Installation Guide: Installing*.

CMEM security implementation consists of the following ICE steps:

Step 5. Implement CMEM Security (Optional)

Perform the following steps to implement CMEM security.

Step 5.1 Grant Access Permissions

Collect the data you need to define the INCONTROL entities and user authorizations in the security product.

RACF Security

1. Add the following commands to define the $$CTOEDM entity to RACF, and authorize users to this entity.
2. To define the entity $$CTOEDM.qname, use the following command:

   ```
   RDEFINE FACILITY $$CTOEDM.qname UACC(NONE)
   ```

3. To authorize USERA to Extended Definition mode, use the following command:

   ```
   PERMIT $$CTOEDM.qname ID(USERA) CLASS(FACILITY) ACCESS(READ)
   ```

   Basic Definition mode is set if the user does not have access to this entity. If the user does have access to this entity, Extended Definition mode is set.

TopSecret Security

4. Define Control-O entities and user authorizations to TopSecret

   For information about how to define Control-O entities and user authorizations to TopSecret, see CMEM Basic Definition Security Calls (on page 73), and CMEM Extended Definition Security Calls (on page 74).

   Modify the following command to establish ownership of the resources in TopSecret to the appropriate owner:

   ```
   TSS ADD(sec-administrator-dept) IBMFAC($$CTO)
   ```

All entity names for each Control-O protected element appear in CMEM Basic Definition Security Calls (on page 73) for Basic Definition mode and CMEM Extended Definition Security Calls (on page 74) for Extended Definition mode.

5. Associate users with definition modes
   1. Customize the following TopSecret command to establish Extended Definition mode for the Control-O installer.

      ```
      TSS PERMIT(USERA) IBMFAC($$CTOEDM.qname) ACC(NONE)
      ```

   2. Modify USERA to the UID of Control-O installer.

      If the user does not have access to this entity, the user is set to work in Basic Definition mode. Otherwise, the user is set to work in Extended Definition mode.

6. Authorize the Control-O installer to use Control-O facilities
   1. Customize the following command to authorize USERA access to Control-O:

      ```
      TSS ADD(USERA) IBMFAC($$CTO)
      ```

   2. Modify USERA to the user ID of the Control-O installer.
   3. Customize the following command to authorize the Control-O installer to use Control-O facilities:

```
TSS PERMIT(USERA) IBMFAC($$CTO) ACC(READ)
```

ACF2/SAF Security

To associate users with Extended Definition Mode, define and authorize the entity $$CTOEDM.qname to ACF2 using the following command:

```
SET RESOURCE(CMF)

COMP

$KEY($$CTOEDM.qname)

 UID(USERA) ALLOW
```

Step 5.2 Customize Security Parameters

Table 36 Security Definition Modes

Mode	Description
Mode Definition	The Definition Mode for the CMEM security modules. Valid values are: ▪ COND — Conditional Definition mode. Default. ▪ BASIC — Basic Definition mode. ▪ EXTEND — Extended Definition mode.
DFMO01	Definition mode for the CTOSE01 security module.
DFMO02	Definition mode for the CTOSE02 security module.

Step 5.3 Save Security Parameters into the Product

This step saves all the security parameters specified for CMEM. When this step is completed, the Status column is automatically updated to COMPLETE.

Control-M Security Modules

This section describes the Control-M security modules in detail.

Module CTMSE01

The CTMSE01 module is the security module of Control-M user Exit CTMX001. It verifies that the user is authorized to order a job. A check is performed to verify if the logged on user is authorized to order jobs on behalf of the user ID as specified in the owner field of the job definition.

Basic Definition Mode

Basic Definition mode authorizes the user access to an INCONTROL protected element. A user who is granted permission to an element is authorized to perform all the actions that are valid for this element (such as add, delete, change and update). The advantage of working in this mode is that it merges several different security events into a logically grouped resource structure. This structure simplifies the administration required to implement INCONTROL security.

Extended Definition Mode

Extended Definition mode grants each user access for a specific action within each INCONTROL protected element. Therefore, a user who is granted access to an INCONTROL element can be granted or denied any action (add, delete, change and update) within that element. This definition mode requires you to define several access rules. For each action there is an associated resource structure. However, Extended Definition mode provides maximum flexibility and accuracy for granting authorizations.

Module CTMSE02

The CTMSE02 module is the security module of Exit CTMX002. It verifies that the owner of a job is allowed to read the JCL library specified in the job definition, and enforces the USER parameter to match the specification made on the job order.

To reduce the amount of resources required for verifying the owner against the security system, the CTMSE02 module can use the internal cache for keeping results of the security requests. These results are refreshed according to the time specified by the LIFETIME parameter. Verification against the security system is performed only for those requests that are not found in cache or if the information in cache has expired. The entire cache is refreshed after NEWDAY process and according to the modify command for refreshing Control-M parameters.

Basic Definition Mode

When Control-M submits a job, the following checks are made:

1. The user ID specified in the owner field of the job definition is authorized to read the JCL library. The CLASS checked is DATASET; the entity checked is the JCL library name. To allow a user to access a JCL library, use one of the following commands, as appropriate:

   For RACF:

   ```
   PERMIT jcl-library-name ACC(READ) ID(USERA)
   ```

   For TSS:

   ```
   TSS PERMIT (USERA) DSN(jcl-library-name) ACC(READ)
   ```

   For ACF2/SAF:

   ```
   COMP
   $KEY(jcl-library-name)
   UID(USERA) ALLOW
   ```

2. If the job statement does not contain parameter USER (or the JCL does not contain a //*JOBFROM statement when ACF2/SAF is in use), parameter USER is added to the job statement and set to owner.

For RACF, parameter GROUP can optionally be added to the job statement and set to the RACF default group.

If the USER parameter exists in the JCL job statement, and the user ID or //*JOBFROM value (for ACF2 users) specified is not same as the owner of the job definition, and the MSUBCHK parameter is set to N (No), the job submission is canceled.

If the USER parameter exists, the user ID specified is not the same as the owner, and parameter MSUBCHK is set to Y (Yes), the class checked is
[SURROGAT | ACIDCHK | CMF] and the entity checked is
[cl-userid.SUBMIT | the JCL user ID | $SUBMIT.cl-userid].

userid is the user ID assigned to the job being submitted.

For started tasks, no security checks are performed, because no distinction is made between the authority to start a started task and the authority to order a started task. The user’s authority is already verified by the CTMSE01 module.

Extended Definition Mode

When Control-M submits a job the following checks are made:

1. The user ID specified in the owner field of the job definition is authorized to read the JCL library. The CLASS checked is DATASET; the entity checked is the JCL library name. To allow a user to access a JCL library, use one of the following commands, as appropriate:

   For RACF:

   ```
   PERMIT jcl-library-name ACC(READ) ID(USERA)
   ```

   For TSS:

   ```
   TSS PERMIT (USERA) DSN(jcl-library-name) ACC(READ)
   ```

   For ACF2/SAF:

   ```
   COMP
   $KEY(jcl-library-name)
   UID(USERA) ALLOW
   ```

2. If the job statement does not contain parameter USER, or the JCL does not contain a //*JOBFROM statement when ACF2/SAF is in use, parameter USER is added to the job statement and set to owner.

   For RACF security, parameter GROUP is optionally added to the job statement and set to the RACF default group.

   If the USER parameter exists, the user ID or //*JOBFROM value (for ACF2 users) specified is not the same as the owner of the job definition, and parameter MSUBCHK is set to N (No), the job submission is cancelled.

   If the USER parameter exists, the user ID specified is not the same as the owner, and parameter MSUBCHK is set to Y (Yes), the class checked is
   [SURROGAT | ACIDCHK | CMF] and the entity checked is
   cl-userid.SUBMIT | the JCL user ID | $SUBMIT.cl-userid].

userid is the user ID assigned to the job being submitted.

   For a started task, the CLASS checked is FACILITY. The entity checked is $$STRSTC.qname.stcname

3. To permit USERA to start a started task named SYSMON, use the following command:

For RACF:

```
PERMIT $$STRSTC.qname.SYSMON ACCESS(READ) ID(USERA) CLASS(FACILITY)
```

For TopSecret:

```
TSS PERMIT(USERA) IBMFAC($$STRSTC.qname.SYSMON) ACC(READ)
```

For ACF2/SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($$STRSTC.qname.STC1) TYPE(CMF)
 UID(USERA) ALLOW
```

Module CTMSE08

The CTMSE08 Control-M security module verifies that the user is authorized to perform actions (for example, hold, delete, rerun) on jobs displayed in the Active Environment screen (Screen 3).

Basic Definition Mode

1. Initial access to Screen 3

   Perform a check to determine if the user is authorized to access Screen 3. The CLASS checked is FACILITY. The entity checked is $$CTMPNL3.qname

2. Refresh commands in the Job Dependency Network screen (Screen 3.N).

   The following entities are checked to verify user authorization for the various REFRESH command options in the Control-M Job Dependency Network screen.

Table 37 Refresh Commands

Command	Entity
REFRESH NET	$$REFNET.qname
REFRESH PROPAGATE	$$REFPROP.qname
REFRESH DEADLINE	$$REFDEAD.qname
REFRESH ALL	$$REFALL.qname
AUTO	$$CTMAUTO.qname

For more information about command REFRESH, see the Online Facilities chapter in the *Control-M for z/OS User Guide*.

3. Subsequent operations in Screen 3

   For all actions (hold, rerun, delete, and so on) that are performed on this screen, an authorization check is made.

The check verifies that the authority of the current user to submit jobs with a USER parameter is equal to that of the specific job being submitted. A user who is authorized to submit a job on behalf of others is also authorized to perform the specific action (hold, rerun, delete, and so on) on jobs belonging to other users.

RACF Security

The CLASS checked is SURROGAT. The entity checked is owner.SUBMIT.

TopSecret Security

The TopSecret Application Interface module (TSSAI) is called with the following parameters:

Resource Class: ACIDCHK

Resource Name: owner

ACF2/SAF Security

The CLASS checked is FACILITY. The entity checked is $SUBMIT.owner.

Extended Definition Mode

1. Initial Access to Screen 3

   Check if the user is authorized to enter screen 3. Check the CLASS, FACILITY and the entity, $$CTMPNL3.qname.

2. Refresh commands in the Job Dependency Network screen (Screen 3.N).

   The following entities are checked to verify user authorization for the various REFRESH command options in the Control-M Job Dependency Network screen.

Table 38 Refresh Commands

Command	Entity
REFRESH NET	$$REFNET.qname
REFRESH PROPAGATE	$$REFPROP.qname
REFRESH DEADLINE	$$REFDEAD.qname
REFRESH ALL	$$REFALL.qname
AUTO	$$CTMAUTO.qname

For more information about command REFRESH, see the online facilities chapter in the *Control-M for z/OS User Guide*.

3. Subsequent operations in Screen 3

   Actions (hold, delete, rerun, and so on) in the Active Environment screen (Screen 3) are separated into different categories of access authority.

   The CLASS checked is FACILITY. The entity checked is:

   $$JOBxrrr.qname.owner

where

- owner is the owner specified in the Job Scheduling Definition screen (Screen 2).
- x is the 1-digit action identifier.
- rrr is the 3-character identifier for each action described in the following table.

Table 39 Active Environment Actions

Action Indentifier	Action	Description
1	ACT LOG SYS STA ZOO	Activate Log Viewsys Veiwstat Zoom
2	CNF FOK FRE HLD RRN RRN	Confirm Force OK Free Hold Return Restore
3	CHA PRI DEL EDI KIL	Change Change priority Delete, Undelete Edit JCL Cancel an executing job

To permit USERA to hold jobs with an owner of USERB, use the following command:

For RACF:

```
PERMIT $$JOB2HLD.qname.USERB ACCESS(READ) ID(USERA) CLASS(FACILITY)
```

For TopSecret:

```
TSS PERMIT(USERA) ISMFAC($$JOB2HLD.qname.UDERB) ACC(READ)
```

For ACF2/SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($$JOB2HLD.qname.USERB) TYPE(CMF)
 UID(USERA) ALLOW
```

CMEM Security Interface Modules

Module CTOSE01

The CTOSE01 module is the security module of Exit CTOX001. It is used to verify that the user is authorized to order the rule. A security check is performed to verify if the logged on (current) user is allowed to order rules on behalf of the user ID as specified in the owner field of the rule definition.

Basic Definition Mode

Verify that the user is authorized to use the user ID (owner) in the rule definition. It is assumed that if the logged on user is allowed to submit jobs on behalf of another user, the logged on user is also allowed to order CMEM rules owned by that user.

RACF Security

For this verification:

Entity Built: owner.SUBMIT

CLASS checked: SURROGAT

where owner is the user ID specified as the owner of the CMEM rule.

TopSecret Security

The Application Interface module (TSSAI) is called with the following parameters:

Resource Class: ACIDCHK

Resource Name: owner

where owner is the user ID specified as the owner of the CMEM rule.

ACF2/SAF Security

For this verification:

Entity Built: $SUBMIT.owner

CLASS checked: FACILITY

where owner is the user ID specified as the owner of the CMEM rule.

Extended Definition Mode

Verify that the user is authorized to specify the user ID (owner) in the rule definition.

RACF Security

For this verification:

Entity Built: $$CTOORD.qname.owner

where owner is the user ID specified as the owner of the CMEM rule.

TopSecret Security

For this verification:

Entity Built: $$CTOORD.qname.owner

where owner is the user ID specified as the owner of the CMEM rule.

ACF2/SAF Security

For this verification:

Entity Built: $$CTOORD.qname.owner

where owner is the user ID specified as the owner of the Control-O rule.

Module CTOSE02

The CTOSE02 module is the security module of Exit CTOX002. It is used to verify that the owner of a rule is allowed to specify the DO statements or ON statements as specified in the rule definition. The module builds a list of security calls, one call for each DO statement and for certain ON statements.

For the rule to be loaded, the owner of the rule must have the authority to request all of the statements specified in the rule definition. If authorization fails for one of the calls, the entire rule load is canceled.

The CLASS checked is always FACILITY. The entity built for each DO statement depends on if Basic Definition mode or Extended Definition mode is used.

Basic Definition Mode

The structure of the entity is as follows:

Table 40 CTOSE02 Basic Definition Entity Structure

Statement	Entity
ON JOBARRIV	$$CTOJAR.qname.jobname
ON JOBEND	$$CTOJED.qname.jobname
ON DSNEVENT	$$CTODSN.qname.jobname
ON STEP	$$CTOSTP.qname.jobname
DO COND or DO RESOURCE	$IOARES.qname.resource-name This is the same structure that the IOASE07 security module builds to verify the user’s authorization to access IOA prerequisite conditions. If a user is allowed to access a Control-M resource, the user is also allowed to access that resource through a CMEM rule execution.

Statement	Entity
DO FORCEJOB	$$CTOCMO.qname.lib-name.table - lib-name is the first 21 characters of the Control-M schedule library. - table is the member name in the Control-M schedule library. The whole entity name is truncated by RACF to 39. This means that table will be entirely truncated unless lib-name is less than 21.
DO STOPJOB	$$CTOJST.qname
For runtime security setting	$$CTORTS.qname.runtime-sec Valid values are: - TRIGGER - OWNER - NONE as specified in rule parameter RUNTSEC.

Extended Definition Mode

The structure of the entity is as follows:

Table 41 CTOSE02 Extended Definition Entity Structure

Statement	Entity
ON JOBARRIV	$$CTOJAR.qname.jobname
ON JOBEND	$$CTOJED.qname.jobname
ON DSNEVENT	$$CTODSN.qname.jobname
ON STEP	$$CTOSTP.qname.jobname
DO COND or DO RESOURCE	$$CTORES.qname.resource-name

Statement	**Entity**
DO FORCEJOB	$$CTOCMO.qname.lib-name.table - lib-name is the first 21 characters of the Control-M schedule library. - table is the member name in the Control-M schedule library. The whole entity name is truncated by RACF to 39. This means that table will be entirely truncated unless lib-name is less than 21.
DO STOPJOB	$$CTOJST.qname
For runtime security setting	$$CTORTS.qname.runtime-sec Valid values are: - TRIGGER - OWNER - NONE as specified in rule parameter RUNTSEC.

3

Control-D and Control-V Security

This chapter describes the procedure used to implement the Control-D and Control-V security interface. Review the explanations below on the elements that are protected in Control-D and Control-V, and then proceed to the step-by-step instructions.

Protecting Control-D and Control-V Elements:

The Control-D and Control-V security interface protects the following Control-D and Control-V elements.

- Ordering missions to the Active Missions file
- Access to decollated sysouts
- Access to and use of the Active Missions file
- Access to packets on Control-D/WebAccess Server Active Transfer file
- Controlling the printing of reports by immediate print requests
- Use of Control-V Quick Access features
- Use of Control-V Indexing features
- Use of Control-D Delivery
- Controlling Online Viewing:
    - Use of the Recipient Tree definition
    - Using various rulers
    - Filter the list of reports
    - Access to reports

Ordering Missions

Each Control-D mission is defined with an OWNER parameter. OWNER is the user ID to which this mission belongs. If a user requests to order a mission, the user must have the authorization to access the owner of the mission. The CTDSE01 Control-D security module verifies that the current user has the authorization to order the mission, using the owner field of the mission.

Accessing Sysouts that Are Decollated

When a report decollating mission is ordered, the CTDSE01 Control-D security module verifies that the user who ordered the mission is authorized to access the sysouts of the jobs that are decollated by this mission.

Accessing and Using of the Mission Status Screen (Screen A)

The Mission Status screen lists the active missions currently handled by Control-D and their status. The user can issue inquiries about a mission within the list or change its status. the CTDSE08 Control-D security module verifies the user’s authorization to perform actions (delete, rerun, zoom, and so on) on missions displayed in the Mission Status screen.

Updating Mission Status in Batch Mode

A user’s authority is verified when the user requests to run missions in batch mode. The CTDSE08 Control-D security module verifies the user’s authority to change the mission status (RESTORE or BACKUP) in the Active Mission file.

Accessing and Using the Control-D/WebAccess Server Active Transfer File

If Control-D/WebAccess Server is installed, the CTDSE19 Control-D security module verifies that the user is authorized to access the Active Transfer File (ATF), transfer reports from the mainframe to the PC, retransmit a packet, delete a packet, and so on.

Filtering the List of Reports in the User Screen (Screen U)

The User screen (Screen U) of Control-D enables the user to view reports online. When the user enters Screen U, only the reports for which the user has access are listed.

The CTDSE04 Control-D security module controls access to User Report List screens. When a user specifies selection criteria for reports, the list of reports that the user is allowed to see is displayed. The list can contain reports that belong to the user and reports of other users that this user is allowed to see. A user can view only the decollated portion of any report that the user is authorized to view.

For information about setting up security definitions, see the description of Exit CTDX004 in the *INCONTROL for z/OS Administrator Guide*.

Using Recipient Tree Definitions

The Recipient Tree is a major security mechanism in Control-D that defines how reports are distributed to users. The tree is defined in one or more library members, and allocated by DD statement DATREE. The current user’s authority is checked to determine if the user is allowed to use the tree definition. If a user defines a tree with multiple members, the security module checks that each member is authorized for use. The tree is protected by IOASE32 and the user must have authorization to it to update it.

Accessing Reports

The CTDSE04 Control-D security module verifies that the user is authorized to perform an operation on a report, such as viewing the report, printing the report, and so on. Although online users can only access reports for which they are authorized, access to a specific report is also verified by checking the user’s authority to print the report, change the ruler, delete the report, and so on.

Limiting the Number of Pages Sent to Spool on Immediate Print Requests

Immediate printing requests can be restricted such that users are authorized to print reports with a specified number of pages. A check is performed to control the size of the report that a user is authorized to print using the immediate print request. There are three ranges of page numbers (defined as MIN, MID and MAX) that verify a user's authority to print a report. All users are authorized to print a report if the number of pages to be printed is less the MIN value. Users can be authorized to print according to a page range between MIN and MID, MID and MAX, or above MAX.

Accessing CDAM Files

When a CDAM file is accessed by a user in Screen U, the user's authority to access the CDAM file is checked by the CTDSE04 security module. This check is performed only if the DCDAMCHK installation parameter is set to YES during the implementation of Control-D security.

Accessing Reports by Control-D/Page On Demand

The CTDSE24 Security module is called to control access to the Control-D Active User Report file and the Control-V Migrated User Report file from Control-D/Page On Demand. The mainframe logon user ID specified in the Control-D/WebAccess Server Communication Setup menu is passed to this module. The associated user exit is CTDX024.

Using Control-D Delivery

If Control-D Delivery is installed, the CTDSE26 Control-D security module verifies if the user is authorized to it as well as its various functions.

Control-D and Control-V Basic Definition Security Calls

Table 42 Control-D and Control-V Basic Definition Security Calls

Protected Element	Type	Class Entity Name	Explanation	Security Module
Controlling Mission Scheduling				
Order a Decollating mission		SURROGAT *owner*.SUBMIT ACIDCHK *owner* FACILITY $SUBMIT.*owner*	*owner* is the name of the user specified in the decollating mission definition.	CTDSE01
Order a Print mission		SURROGAT *owner*.SUBMIT ACIDCHK *owner* FACILITY $SUBMIT.*owner*	*owner* is the name of the user specified in the print mission definition.	CTDSE01

Protected Element	**Type**	**Class** **Entity Name**	**Explanation**	**Security Module**
Order a Backup Mission		SURROGAT *owner*.SUBMIT ACIDCHK *owner* FACILITY $SUBMIT.*owner*	*owner* is the name of the user specified in the backup mission definition.	CTDSE01
Order a Restore Mission		SURROGAT *owner*.SUBMIT ACIDCHK *owner* FACILITY $SUBMIT.*owner*	*owner* is the name of the user specified in the restore mission definition.	CTDSE01
Order a Migration Mission		SURROGAT *owner*.SUBMIT ACIDCHK *owner* FACILITY $SUBMIT.*owner*	*owner* is the name of the user specified in the migration mission definition.	CTDSE01
Controlling Access to PREFIX Parameter				
When using parameter ON PREFIX, two checks are performed. If the user has read authority to entity $$CTDPREFIX.qname, then a second check is made to entity $$CTDPRF.qname.prefix. qname is the name used to assign different authorizations to various Control-D and Control-V environments (such as Test and Production).				CTDSE01
Controlling Ordering of Decollations				
Controlling Access to Sysouts		FACILITY $$CTDJOB.*qname.jobname*	*jobname* is the name of the sysout name to be decollated.	CTDSE01

Protected Element	Type	Class Entity Name	Explanation	Security Module
Controlling Access and Use of the Active Missions File (Screen A)				
Access to the Active Mission Status screen		FACILITY $$CTDPNLA.*qname*		CTDSE08
Use of the Mission Status screen (screen A)		SURROGAT *owner*.SUBMIT ACIDCHK *owner* FACILITY $SUBMIT.*owner*	*owner* is the name of the user specified in the mission definition.	CTDSE08
If the user runs in batch mode, the entity checked is $$CTDRRST.*qname.owner* Batch includes backup jobs, restore jobs, and Control-V migration jobs.				CTDSE08
Controlling Access and Use of the Active Missions File (Screen F)				
Access Active Transfer screen		FACILITY $$CTDPNLF.*qname*		CTDSE19
Use of the File Transfer facility		SURROGAT *owner*.SUBMIT ACIDCHK *owner* FACILITY $SUBMIT.*owner*	*owner* is the name of the user specified in the packet definition.	CTDSE19
Controlling Online Viewing				
Use of Recipient Tree Definitions by Online Users		FACILITY $$TREE.*dsn.member*	*dsn* is the first 23 chars of the *dsname* allocated to DD statement DATREE. member is the member name allocated.	CTDSE04
Filtering the List of Reports in Screen U		The user list is created by scanning the Recipient Tree using the USERLIST program (CTDUSR).	See the “Filtering the List of Reports in the User Screen (Screen U)” section in Control-D and Control-V Security (on page 86).	CTDSE04

Protected Element	Type	Class Entity Name	Explanation	Security Module
Controlling Access to CDAM files		DATASET *dsn*	*dsn* is the CDAM dsname. This check is performed only if parameter DCDAMCHK is set to YES. See Step 1. Implement Control-D Security (on page 99).	CTDSE04 CTDSE24
Controlling Access to Reports		FACILITY $$CTDACT.*qname.userid*	*userid* is the user ID to whom the report belongs (the recipient of the report). If parameter DGLBRULR is set to YES, the ruler name is used instead of the Global Ruler owner. See Step 1. Implement Control-D Security (on page 99).	CTDSE04
Controlling Access to Reports by Control-D/Page On Demand and Control-D/ WebAccess		FACILITY $$CTDASR.*qname.userid* Logical View: $$CTDASR.*qname.report-name* $$CTDASR.*qname* All other requests: $$CTDASR.*qname.userid*	*report-name* is name of the report for which a Logical View request is executed. *report-name* is * or ? or blank. *userid* is the user ID to whom the report belongs (the recipient of the report).	CTDSE24
Using Control-D Delivery				
Controlling Control-D Delivery functions		FACILITY $$CTDCDD.*qname.userid*	All Control-D Delivery functions.	CTDSE26
Controlling Immediate Printing of Reports by Reports Size				
Printing a report within MIN-MID number of pages		FACILITY $$PAGIII		CTDSE04
Printing a report within MIN-MAX number of pages		FACILITY $$PAGII		CTDSE04

Protected Element	Type	Class Entity Name	Explanation	Security Module
Printing a report of more than MAX number of pages		FACILITY $$PAGI		CTDSE28
Printing a report within MIN-MID number of pages		FACILITY $$PGASRIII		CTDSE28
Printing a report within MIN-MAX number of pages		FACILITY $$PGASRII		CTDSE28
Printing a report of more than MAX number of pages		FACILITY $$PGASRI		CTDSE28 CTDSE24
Entering to screen DO option 1 Report Clique		FACILITY $$CTDOBJ.*qname*.ENTRY.REPCLQ	*qname* is the name used to assign different authorizations to various Control-D environments (for example, Test and Production).	
Entering to screen DO option 2 Resource Set		FACILITY $$CTDOBJ.*qname*.ENTRY.RESSET	*qname* is the name used to assign different authorizations to various Control-D environments (for example, Test and Production).	
Saving a new or a modified report clique name		FACILITY $$CTDOBJ.*qname*.SAVE.*clique-name*	*qname* is the name used to assign different authorizations to various Control-D environments (for example, Test and Production). *clique-name* is the name of the report clique name to be created or saved.	

Protected Element	Type	Class Entity Name	Explanation	Security Module
Deleting a report clique name or a resource set name		FACILITY $$CTDOBJ.*qname*.DELETE. *clique-name* FACILITY $$CTDOBJ.*qname*.DELETE. *resource-set-name*	*qname* is the name used to assign different authorizations to various Control-D environments (for example, Test and Production). *clique-name* is the name of the report clique name to be deleted. *resource-set*-name is the name of the resource set to be deleted.	
Accessing the Global Index Path that is included in the list of paths in the Control-D/Web Access Index box or specified in the Control-D/Web Access filter manually by the user		FACILITY $$CTDASR.*qname.#nn*	*qname* is the name used to assign different authorizations to various Control-D environments (for example, Test and Production). *nn* is the path number	

Control-D and Control-V Extended Definition Security Calls

Table 43 Control-D and Control-V Extended Definition Security Calls

Protected Element	Type	Class Entity Name	Explanation	Security Module
Controlling Mission Scheduling				
Order a Decollating mission		FACILITY $$REPORD.*qname.owner*	*owner* is the name of the user specified in the decollating mission definition. *qname* is the name used to assign different authorizations to various Control-D and Control-V environments (for example, Test and Production).	CTDSE01

Protected Element	Type	Class Entity Name	Explanation	Security Module
Order a Print mission		FACILITY $$PRTORD.*qname.owner*	*owner* is the name of the user specified in the printing mission definition.	CTDSE01
Order a Backup Mission		FACILITY $$BKPORD.*qname.owner*	*owner* is the name of the user specified in the backup mission definition.	CTDSE01
Order a Restore Mission		FACILITY $$RSTORD.*qname.owner*	*owner* is the name of the user specified in the restore mission definition.	CTDSE01
Controlling Access to PREFIX				
When using parameter ON PREFIX, two checks are performed. If the user has read authority to entity $$CTDPREFIX.*qname*, a second check is made to entity $$CTDPRF.*qname.prefix*.				CTDSE01

Protected Element	**Type**	**Class Entity Name**	**Explanation**	**Security Module**
Controlling Access to Sysouts				
Controlling Access to Sysouts		FACILITY $$CTDJOB.*qname.jobname*	*jobname* is the name of the sysout name to be decollated.	CTDSE01
Controlling Access and Use of the Active Missions File (Screen A)				
Access to the Active Mission Status screen		FACILITY $$CTDPNLA.*qname*		CTDSE08
Use of the Mission Status screen (screen A)		FACILITY Hold: $$MIS2HLD.*qname.owner* Free: $$MIS2FRE.*qname.owner* Rerun: $$MIS2RRN.*qname.owner* Change: $$MIS3CHA.*qname.* *owner* Delete: $$MIS3DEL.*qname.owner* Print: $$MIS3PPL.*qname.owner* Update: $$MIS3UPD.*qname.owner*	*owner* is the name of the user specified in the mission definition.	CTDSE08
If the user runs in batch mode, the entity checked is $$CTDRRST.*qname.owner*. Batch includes backup jobs, restore jobs, and Control-V migration jobs.				CTDSE08
Controlling Access and Use of the Active Missions File (Screen F)				
Access the Active Transfer screen		FACILITY $$CTDPNLF.*qname*		CTDSE19

Protected Element	Type	Class Entity Name	Explanation	Security Module
Use of the File Transfer facility		FACILITY Read: $$DPC1VIE.*qname.owner* Hold: $$DPC2HLD.*qname.owner* Free: $$DPC2FRE.*qname.owner* Print: $$DPC3PRN.*qname.owner* Delete: $$DPC3DEL.*qname.owner* Retransmit: $$DPC4TRN.*qname.owner* Modify: $$DPC4TRN.*qname.owner*	*owner* is the name of the user specified in the packet definition.	CTDSE19
Controlling Online Viewing				
Controlling the use of Recipient Tree Definitions by Online Users		FACILITY $$TREE.*dsn.member*	*dsn* if the first 23 characters of the dsname allocated to the DATREE DD card. member is the *member* name allocated.	CTDSE04
Filtering the List of Reports in Screen Ua		The user list is created by scanning the Recipient Tree using the USERLIST program (CTDUSR).		CTDSE04
Controlling Access to CDAM filesb		DATASET *dsn*	*dsn* is the CDAM dsname.	CTDSE04 CTDSE24
Controlling usage of parameter DREPLST when set to YES		FACILITY $$REPLST.*qname.rec-name*	*rec-name* is the recipient name.	CTDSE04 CTDSE24

Protected Element	Type	Class Entity Name	Explanation	Security Module
Controlling Access to Reportsc		FACILITY Update: $$RECUPD.*qname.userid* Insert: $$RECINS.*qname.userid* Delete: $$RECDEL.*qname.userid* Reprint: $$RECRPR.*qname.userid* Restore: $$RECRST.*qname.userid* View without Recipient Tree: $$REPLST.*qname.userid* Immediate print: $$RECIPR.*qname.userid* Define ruler: $$EXTENT.*qname.userid* Ruler on/off: $$RULONF.*qname.userid* Use a global ruler: $$RULONF.*qname.userid* Save a ruler: $$RULSAV.*qname.userid* Give to: $$GIVETO.*qname.userid* Browse NOTES: $$VIEWNO.*qname.userid* Add/Update NOTES: $$EDITNO.*qname.userid* Delete NOTES: $$DELNOT.*qname.userid* Add NOTES: $$ADDNOT.*qname.userid* Update NOTES: $$UPDNOT.*qname.userid* FACILITY Update Report View Indicator: $$VEWUPD.*qname.userid* Cancel Restore for History Report: $$UNRSTR.*qname.userid* Perform a recall of a migrated CDAM file: $$CHKRCL.*qname.userid* Submit a job to perform recall of a migrated CDAM file: $$RECALL.*qname.userid* View the report in hexadecimal format: $$RECHEX.*qname.userid* Control-V Use Control-V Quick Access features: $$CTVQAC.*qname.userid* Use Control-V Indexing features: $$CTVINX.*qname.userid*	*userid* is the user ID to whom the report belongs (the recipient of the report).	CTDSE04

Protected Element	Type	Class Entity Name	Explanation	Security Module
Entering to screen DO option 1 Report Clique		FACILITY $$CTDOBJ.*qname*.ENTRY. REPCLQ	*qname* is the name used to assign different authorizations to various Control-D environments (for example, Test and Production).	CTDSE28
Entering to screen DO option 2 Resource Set		FACILITY $$CTDOBJ.*qname*.ENTRY. RESSET	*qname* is the name used to assign different authorizations to various Control-D environments (for example, Test and Production).	CTDSE24
Saving a new or a modified report clique name		FACILITY $$CTDOBJ.*qname*.SAVE.*clique-name*	*qname* is the name used to assign different authorizations to various Control-D environments (for example, Test and Production). *clique-name* is the name of the report clique name to be created or saved.	CTDSE28
Deleting a report clique name or a resource set name		FACILITY $$CTDOBJ.*qname*.DELETE. *clique-name* FACILITY $$CTDOBJ.qname.DELETE. *resource-set-name*	*qname* is the name used to assign different authorizations to various Control-D environments (for example, Test and Production). *clique-name* is the name of the report clique name to be deleted. *resource-set-name* is the name of the resource set to be deleted.	CTDSE28

Implementing Control-D and Control-V Security

This section details the steps required to implement the Control-D security interface.

The Control-D security interface can be installed either as part of the customized installation path, or during the Customization process after installation. Both options use the INCONTROL Installation and Customization Engine (ICE) application. If you are not familiar with the ICE interface, see the *INCONTROL for z/OS Installation Guide: Installing*.

The Control-D security interface cannot be implemented until IOA security is installed. Verify that IOA security is implemented before implementing Control-D security.

➢ To install the Control-D security interface

- **a.** Enter the main ICE screen.
- **b.** Select Customization.
- **c.** Enter CTD in the Product field.
- **d.** Select Security Customization.
- **e.** Perform all major and minor steps required to install the security product.

Step 1. Implement Control-D Security

Perform the following steps to implement Control-D security.

Step 1.1. Grant Access Permissions

1. Collect the data you need to define the INCONTROL entities and user authorizations to the security product.
2. In ICE run the steps "Control-D Security Definitions" and "Functions Security Definitions" to create two sample jobs.
3. Enter the above security definitions into the sample jobs just created and save the jobs in the INSTWORK library.
4. Submit the jobs to define security to IOA and Control-D.

Step 1.2. Customize Security Parameters

Perform the steps in ICE for all the required parameters, as follows:

Table 44 ICE Parameters

Parameter	Description
DEFMCHKD	When choosing a definition mode as COND to any of the Control-D security modules, use qname together with the value given to this parameter as the high level qualifier, to determine the real definition mode to be used.
SECTOLD	Determine the action to perform if your security product is inactive or a specific resource is not defined in the security product. Valid values are: ▪ YES — Perform the action. ▪ NO — Do not perform the action.

Parameter	Description
DCDAMCHK	Specify CDAM CHECK option. Valid values are: ■ YES — Check online users for authorization access to CDAM datasets. ■ NO — Do not check online users authorization to CDAM datasets. Default.
DGLBRULR	Specify Global Ruler option. Valid values are: ■ YES — For ruler operations check authority to Global Ruler name instead of owner. ■ NO — For ruler operations, use Global Ruler owner (that is fixed as "MASTER" for authority checks). Default.
SYSDCHK	Specify SYSDATA viewing option. Valid values are: ■ YES — Enable SYSDATA viewing by adding the jobname to the entity name. ■ NO — Do not enable SYSTDATA viewing.

The following parameters determine the number of pages a user is authorized to print using the immediate print request:

Table 45 Page Authorization Parameters

Parameter	Quantity	
DPAGMIN	10	>0
DPAGMID	100	>min
DPAGMAX	200	>mid

Table 46 ICE Definition Parameters

Parameter	Description
RACULIST	RACF USERLIST options. Valid values are: ▪ STD — Authorize a user to view a report if the recipient authorized the user with the AUTHORIZE statement. Default. ▪ ALL — Authorize a user to view a report if the recipient used the AUTHORIZE statement to authorized any of the RACF groups to which the user is connected. ▪ MIXED — Combines STD and ALL. ▪ GRP — Authorize a user to view all the reports if the recipient authorized the user's default group with the AUTHORIZE statement. ▪ NO — Do not check authorization through the Recipient Tree.
TSSULIST	TopSecret USERLIST option. Valid values are: ▪ STD — Authorize a user to view a report if the recipient authorized the user with the AUTHORIZE statement. Default. ▪ NO — Do not check authorization through the Recipient Tree.
SAFULIST	ACF2 USERLIST option. Valid values are: ▪ STD — Authorize a user to view a report if the recipient authorized the user with the AUTHORIZE statement. Default. ▪ NO — Do not check authorization through the Recipient Tree. ▪ UID — Authorize a user to view reports based on the first 19 characters of the user's UID string instead of the logon ID, using the SYNONYM statement. The UID string in the SYNONYM statement must have a leading "-" character followed by the 19 characters of the UID string.

Parameter	Description
DREPLST	Determine if the current userid must be authorized to entity $$REPLST.qname.recname to access reports of recipient recname. Valid values are: ▪ YES – Current userid must be authorized to entity $$REPLST.qname.recname. If the value NO is specified for the RACULIST, TSSULIST and SAFULIST parameters, it is not necessary to maintain AUTHORIZE fields in the Recipient Tree. This enables report decollation without using the Recipient Tree, but will degrade performance when accessing User Report List files that have a large number of recipients. For information how to avoid performance degradation, see "Decollation Without the Recipient Tree" in the implementation hints chapter in the *Control-D and Control-V User Guide*. ▪ NO – Current userid need not be authorized to entity $$REPLST.qname.recname.

Table 47 Mode Definition Parameter

Mode	Description
Mode Definition	Specify one of the following values to determine the Definition mode for the Control-D security modules: ▪ COND – Conditional Definition mode. Default. ▪ BASIC – Basic Definition mode. ▪ EXTEND – Extended Definition mode.
DFMD01	Definition mode for the CTDSE01 Control-D security module.
DFMD04	Definition mode for the CTDSE04 Control-D security module.
DFMD08	Definition mode for the CTDSE08 Control-D security module.
DFMD19	Definition mode for the CTDSE19 Control-D security module.
DFMD23a	Definition mode for the CTDSE23 Control-D security module.
DFMD24	Definition mode for the CTDSE24 Control-D security module.
DFMD26	Definition mode for the CTDSE26 Control-D security module.
DFMD27a	Definition mode for the CTDSE27 Control-D security module.

Step 1.3. Save Security Parameters into Product

This step saves all the security parameters specified for Control-D. When this step completes, the Status column is automatically updated to COMPLETE.

Step 2. RACF Security Definitions (Optional)

Note: To activate the IOA to XBM interface, XBM must be active and at least one of the following parameters: ZIIPXBMO, ZIIPXBMP, or ZIIPXBMA must be set to Y.

A RACF call is made by XBM on the initial request to determine if a user is authorized to perform the requested function. The following RACF profile is used:

```
BMCXBM.<XBM_SSID>.ZIIP
```

If this profile is not defined, permission will be granted. More detailed information can be found in the XBM documentation.

Step 2.1 Control-D Security Definitions

Select this step to edit member CTDSRAC2 in the IOA INSTWORK library

Perform the following steps to define the required permissions:

1. Associating users with Extended Definition mode.
   a. Add the following commands to define the $$CTDEDM entity and authorize users to use this entity.
   b. Define the entity $$CTDEDM.qname as follows:

      RDEFINE FACILITY $$CTDEDM.qname UACC(NONE)

   c. Authorize USERA to Extended Definition mode as follows:

      PERMIT $$CTDEDM.qname ID(USERA) CLASS(FACILITY) ACCESS(READ)

2. Submit the job for execution.

   This job must be run under a user who has authorization to enter these commands.

   Scan the output of the job for information and error messages. All job steps must end with a condition code of 0.

Step 2.2 Function Security Definitions

Select this step to edit the CTDSRAC3 member in the IOA INSTWORK library. This job contains various definitions for Control-D. Review the definitions and modify according to your site's requirements.

Step 2.3 Control Program Access to Datasets

Select this step to edit the CTDSRAC4 member in the IOA INSTWORK library. This member contains a sample of the definitions required to define Program Pathing access authorizations to Control-D datasets.

Review the definitions and modify according to the requirements of your site.

BMC Software recommends that the security administrator first read Limiting Access to Specific Programs (on page 200) and the *IBM Resource Access Control Facility Security Administrator’s Guide* before submitting this job.

Step 3. TopSecret Security Definitions (Optional)

Step 3.1 Control-D Security Definitions

Select this step to edit member CTDSTSS2 in the IOA INSTWORK library

1. Define Control-D to the TopSecret Facility Matrix.

   CTDSTSS2 contains the necessary command to dynamically define Control-D in TopSecret Facility Matrix.

   a. Modify USER3 in the Facility definition command to a free entry in the Facility Matrix, as follows:

      ```
      TSS MODIFY FAC(USER3=NAME=CTD)
      ```

      This command defines Control-D in the Facility Matrix until the next IPL.

   b. To permanently define the facility, update the TopSecret parameter member. This member is usually called TSSPARM0.

   c. Copy the Control-D facility definition from member CTDSTSS5 in the IOA INSTWORK library to member TSSPARM0.

   d. Update the Facility Matrix entry name to the same name that is specified in the TSS MODIFY command above.

2. Define Control-D ACID to TopSecret.

   Change the value of parameter DEPT from sec-administrator-dept to the appropriate ACID:

   ```
   TSS CRE (CTD) NAME (...) DEPT(sec-administrator-dept)
   ```

3. Define Control-D started tasks to TopSecret.

   Change the ACID definition in the following commands to the appropriate ACID:

   ```
   TSS ADD(STC) PROC(CONTROLD) ACID(CTD)
   TSS ADD(STC) PROC(CTDPRINT) ACID(CTD)
   TSS ADD(STC) PROC(CTDNDAY) ACID(CTD)
   ```

4. Allow Control-D ACID to Control-D datasets.

   Authorizations to access Control-D datasets are defined during the Control-D installation process. This step must be completed before proceeding with security implementation. For information about how to grant users access to Control-D datasets, see the Control-D chapter in the *INCONTROL for z/OS Installation Guide: Installing*.

   Connect the appropriate profile to the Control-D ACID in the following command:

   ```
   TSS ADD (CTD) PROF (profile-name)
   ```

5. Define Control-D entities and user authorizations to TopSecret.

   For more information about how to define Control-D entities and user authorizations to TopSecret, see Control-D and Control-V Basic Definition Security Calls (on page 88), and Control-D and Control-V Extended Definition Security Calls (on page 93).

   Modify the following command to establish ownership of the resources in TopSecret to the appropriate owner:

   ```
   TSS ADD(sec-administrator-dept) IBMFAC($$CTD)
   ```

For samples of user authorizations, review member CTDSTSS3 in the IOA INSTWORK library.

All entity names for each Control-D protected element appear in Control-D and Control-V Basic Definition Security Calls (on page 88) for Basic Definition mode and Control-D and Control-V Extended Definition Security Calls (on page 93) for Extended Definition mode.

6. Associate users with Extended Definition modes.

Customize the following TopSecret command to establish Extended Definition mode for the Control-D installer.

```
TSS PERMIT (USERA) IBMFAC($$CTDEDM.qname) ACC(NONE)
```

Modify USERA to the UID of Control-D installer.

Do not define the $$CTDEDM entity to operate in warning mode since this causes all users to operate in Extended Definition mode.

7. Authorize the Control-D installer to use Control-D facilities

Customize the following command to authorize USERA access Control-D as follows:

```
TSS ADD(USERA) IBMFAC($$CTD)
```

Modify USERA to the user ID of the Control-D installer.

Customize the following command to authorize the Control-D installer to use Control-D facilities:

```
TSS PERMIT(USERA) IBMFAC($$CTD) ACC(READ)
```

8. Submit the job.

This job must be run under the ACID of the general security administrator (SCA) who has authorization to enter these TopSecret commands.

All job steps must end with a condition code of 0.

Step 3.2 Function Security Definitions

Select this step to edit the CTDSTSS3 member in the IOA INSTWORK library. This job contains various definitions for Control-D. Review the definitions and modify according to your site's requirements.

Step 3.3 Control Program Access to Datasets

Select this step to edit the CTDSTSS4 member in the IOA INSTWORK library. This member contains a sample of the definitions required to define Program Pathing access authorizations to Control-D datasets.

Review the definitions and modify according to your site’s requirements.

BMC Software recommends that the security administrator first read Limiting Access to Specific Programs (on page 200) and the *TopSecret Implementation Guide* before submitting this job.

Step 3.4 Define CTD to TopSecret Facility Matrix (Optional)

Select this step to edit the CTDSTSS5 member in the IOA INSTWORK library. Perform the following steps to define Control-D in the TopSecret Facility Matrix:

9. Modify USER3 in the Facility definition command to a free entry in the Facility Matrix, with the following command:

```
TSS MODIFY FAC(USER3=NAME=CTD)
```

10. Copy modified member CTDSTSS5 into TSSPARM0.

Step 4. ACF2 Security Definitions (Optional)

Step 4.1 Control-D Security Definitions

Select this step to edit member CTDSSAF2 in the IOA INSTWORK library

1. Define Control-D started tasks under ACF2.
   - a. Define the Control-D started tasks as a valid started task under ACF2 (CONTROLD, CTDPRINT, CTDNDAY).
   - b. Add the multi-user address space (MUSSAS) parameter to the logon ID record that is created for the Control-D started task.

If the site uses more than one Control-D monitor, parameter MUSSAS must be added to all the logon ID records previously created.

2. Associating users with extended definition mode.

   Define and authorize the entity $$CTDEDM.qname to ACF2 using the following command:

   ```
   SET RESOURCE(CMF)
   COMP
   $KEY($$CTDEDM.qname) TYPE(CMF)
    UID(USERA) ALLOW
   ```

3. Define entities and user authorizations to CA-ACF2/SAF.

   Example

   To authorize USERA (the user ID of the Control-D installer) access to a given Control-D entity, use the following command:

   ```
   SET RESOURCE(CMF)
   COMP
   $KEY($$CTDnnn.qname) TYPE(CMF)
    UID(USERA) ALLOW
   ```

   where qname is the name used to assign different authorizations to different Control-D environments (such as Test and Production). This parameter is specified during IOA installation.

   Change USERA to the UID string of the Control-D installer.

   All entity names for each Control-D protected element appear in Control-D and Control-V Basic Definition Security Calls (on page 88) for Basic Definition mode and Control-D and Control-V Extended Definition Security Calls (on page 93) for Extended Definition mode.

For samples of user authorizations, review member CTDSSAF3 in the IOA INSTWORK library.

4. Submit the job.

   This job must be run under a user of an ACF2 administrator who has authorization to enter these ACF2 commands.

   Scan the output of the job for information and error messages produced by ACF2. All job steps must end with a condition code of 0.

Step 4.2 Function Security Definitions

Select this step to edit the CTDSSAF3 member in the IOA INSTWORK library. This job contains various definitions for Control-D. Review the definitions and modify according to your site's requirements.

Step 4.3 Control Program Access to Datasets

Select this step to edit the CTDSSAF4 member in the IOA INSTWORK library. This member contains a sample of the definitions required to define Program Pathing access authorizations to Control-D datasets.

Review the definitions and modify according to your site's requirements.

BMC Software recommends that the security administrator first read Limiting Access to Specific Programs (on page 200) and the *CA-ACF2 Administrator's Guide* before submitting this job.

Control-D Security Interface Modules

This section describes the Control-D security interface modules.

Module CTDSE01

The CTDSE01 module is the security module of Control-D Exit 1 (CTDX001). This module verifies that the user is authorized to order a mission. A check is performed to verify if the current user is allowed to order missions on behalf of the user ID specified in the owner field of the mission definition.

If the user ID is the same as the owner ID, no security check is performed.

Basic Definition Mode

Use of Recipient Tree Definition

The user's authority to use a certain dataset specified under the DATREE DD is checked with the following entity:

```
$$TREE.dsn.member
```

where

- dsn is the dataset name of a library referenced by DD statement DATREE.
- member is the PDS member referenced by the DD statement.

For each dataset concatenated in DD statement DATREE, the security module is called once and checks each statement with the above entity.

The dataset name is truncated to 23 characters.

To permit USERA to use a DSN set to library-name(member) in the DATREE DD, use the following commands:

For RACF:

```
RDEFINE FACILITY $$TREE.dsn.member UACC(NONE)
```

```
PERMIT $$TREE.dsn.member ACCESS(READ) ID(USERA) CLASS(FACILITY)
```

For TopSecret:

```
TSS ADD(system-dept) IBMFAC($$TREE)
```

```
TSS PERMIT(USERA) IBMFAC($$TREE.library.member-name) ACC(READ)
```

For ACF2 / SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($$TREE.dsn.member-name) TYPE(CMF)
 UID(USERA) ALLOW
```

Access a Report Under Screen U

When the user attempts to specify an action on a certain report (for example, view, ruler change, print), the entity checked is $$CTDACT.qname.userid where userid is the user name related to the report being accessed. There is no distinction between the different actions that can be specified. The user is either allowed to do anything with the report, or completely denied access to the report.

To permit USERA to perform actions to the reports of USERB, use the following command:

For RACF:

```
RDEFINE FACILITY $$CTDACT.qname.USERB UACC(NONE)
PERMIT $$CTDACT.qname.USERB ACCESS(READ) ID(USERA) CLASS(FACILITY)
```

For TopSecret:

```
TSS ADD(system-dept) IBMFAC($$CTDACT)
TSS PERMIT(USERA) IBMFAC($$CTDACT.qname.USERB) ACC(READ)
```

For ACF2 / SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($CTDACT.qname.USERB) TYPE(CMF)
UID(USERA) ALLOW
```

Limit Immediate Print of Reports

When the user requests immediate print for a report, and the number of pages for the report is more than DPAGMIN, an additional entity is checked. The entity checked is:

Table 49 Report Limits

Entity	Description
$$PAGIII	If the number of pages is greater than DPAGMIN but less than or equal to parameter DPAGMID.
$$PAGII	If the number of pages is greater than DPAGMID but less than or equal to parameter DPAGMAX.
$$PAGI	If the number of pages is greater than DPAGMAX.

Extended Definition Mode

When the order is for a printing mission, the entity checked is $$PRTORD.qname.owner where owner is the owner ID specified in the printing mission definition.

RACF Security

The commands to permit USERA to order a printing mission with an owner of USERB are:

```
RDEFINE FACILITY $$PRTORD.qname.USERB UACC(NONE)

PERMIT $$PRTORD.qname.USERB CLASS(FACILITY) ID(USERA) ACC(READ)
```

TopSecret Security

When the order is for a printing mission, the entity checked is $$PRTORD.qname.owner, where owner is the owner ID specified in the printing mission definition.

The commands to permit USERA to order a printing mission with an owner of USERB are:

```
TSS PERMIT(USERA) IBMFAC($$PRTORD.qname.USERB) ACC(READ)
```

ACF2/SAF Security

When the order is for a printing mission, the entity checked is $$PRTORD.qname.owner, where owner is the owner ID specified in the print mission definition.

The commands to permit USERA to order a printing mission with an owner of USERB are:

```
SET RESOURCE(CMF)

COMP

$KEY($PRTORD.qname.USERB)

 UID(USERA) ALLOW
```

The following entities are checked for orders in all security products:

Table 48 CTDSE01 Entity Checking

Order	Entity Checked
Restore mission	$$RSTORD.qname.owner
Backup mission	$$BKPORD.qname.owner
Decollating mission	$$REPORD.qname.owner $$CTDJOB.qname.jobname

Module CTDSE04

The CTDSE04 module is the security module of the CTDX004 Control-D Exit. This module builds a filtered list of reports displayed on the user’s screen and verifies the user’s authority to perform actions in the Control-D User Report List screen (option U in the IOA Primary Option menu).

This module verifies that:

- The user is authorized to use the Recipient Trees defined under ddname DATREE.
- The filtered list of reports applies only to those reports for which the user has authorization.
- The user is authorized to perform a specific action (print, delete, and so on) on a certain report.
- Users are not authorized to print very long reports using immediate print requests.

Filtering of the report list is built by scanning the Recipient Tree without any interaction with the security product. For more details, see "Filtering the List of Reports in the User Screen" in Control-D and Control-V Security (on page 86). The CLASS checked is FACILITY (unless otherwise specified). The entity used to check authorization depends on if Basic Definition mode or Extended Definition mode is used.

Basic Definition Mode

Use of Recipient Tree Definition

The user's authority to use a certain dataset specified under the DATREE DD is checked with the following entity:

```
$$TREE.dsn.member
```

where

- dsn is the dataset name of a library referenced by DD statement DATREE.
- member is the PDS member referenced by the DD statement.

For each dataset concatenated in DD statement DATREE, the security module is called once and checks each statement with the above entity.

The dataset name is truncated to 23 characters.

To permit USERA to use a DSN set to library-name(member) in the DATREE DD, use the following commands:

For RACF:

```
RDEFINE FACILITY $$TREE.dsn.member UACC(NONE)
PERMIT $$TREE.dsn.member ACCESS(READ) ID(USERA) CLASS(FACILITY)
```

For TopSecret:

```
TSS ADD(system-dept) IBMFAC($$TREE)
TSS PERMIT(USERA) IBMFAC($$TREE.library.member-name) ACC(READ)
```

For ACF2 / SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($$TREE.dsn.member-name) TYPE(CMF)
 UID(USERA) ALLOW
```

Access a Report Under Screen U

When the user attempts to specify an action on a certain report (for example, view, ruler change, print), the entity checked is $$CTDACT.qname.userid where userid is the user name related to the report being accessed. There is no distinction between the different actions that can be specified. The user is either allowed to do anything with the report, or completely denied access to the report.

To permit USERA to perform actions to the reports of USERB, use the following command:

For RACF:

```
RDEFINE FACILITY $$CTDACT.qname.USERB UACC(NONE)
```

```
PERMIT $$CTDACT.qname.USERB ACCESS(READ) ID(USERA) CLASS(FACILITY)
```

For TopSecret:

```
TSS ADD(system-dept) IBMFAC($$CTDACT)
```

```
TSS PERMIT(USERA) IBMFAC($$CTDACT.qname.USERB) ACC(READ)
```

For ACF2 / SAF:

```
SET RESOURCE(CMF)
```

```
COMP
```

```
$KEY($CTDACT.qname.USERB) TYPE(CMF)
```

```
UID(USERA) ALLOW
```

Limit Immediate Print of Reports

When the user requests immediate print for a report, and the number of pages for the report is more than DPAGMIN, an additional entity is checked. The entity checked is:

Table 49 Report Limits

Entity	Description
$$PAGIII	If the number of pages is greater than DPAGMIN but less than or equal to parameter DPAGMID.
$$PAGII	If the number of pages is greater than DPAGMID but less than or equal to parameter DPAGMAX.
$$PAGI	If the number of pages is greater than DPAGMAX.

Examples

For RACF:

To allow USERA to immediately print a report of any size, use the following commands:

```
RDEFINE FACILITY $$PAGI* UACC(NONE)
PERMIT $$PAGI* CLASS(FACILITY) ID(USERA) ACCESS(READ)
```

To permit USERA to print reports that do not exceed the DPAGMAX number of pages, use the following commands:

```
RDEFINE FACILITY $$PAGII* UACC(NONE)
PERMIT $$PAGII* ID(USERA) CLASS(FACILITY) ACCESS(READ)
```

For TopSecret:

To allow USERA to immediately print a report of any size, use the following commands:

```
TSS ADD(system-dept) IBMFAC($$PAGI)
TSS PERMIT(USERA) IBMFAC($$PAGI) ACC(READ)
```

For ACF2 / SAF:

To allow USERA to immediately print a report of any size, use the following commands:

```
SET RESOURCE(CMF)
COMP
$KEY($$DPAGI**) TYPE(CMF)
 UID(USERA) ALLOW
```

To permit USERA to print reports that do not exceed the DPAGMAX number of pages, use the following commands:

```
SET RESOURCE(CMF)
COMP
$KEY($$DPAGII*) TYPE(CMF)
 UID(USERA) ALLOW
```

Extended Definition Mode

Use of Recipient Tree Definition

The entity $$TREE.dsn.member is used to verify that the user is authorized to use a dataset referenced by DD statement DATREE.

where

- dsn is the dataset name of a library specified in DD statement DATREE.
- member is the PDS member referenced in the DD statement.

The security module is called once for each dataset concatenated in DD statement DATREE and checks each one with the above entity.

If the library name is longer than 23 characters, it is truncated to 23 characters. To permit USERA to use a member DSN set to library-name(member) referenced in DD statement DATREE, use the following commands:

For RACF:

```
RDEFINE FACILITY $$TREE.library.member UACC(NONE)
PERMIT $$TREE.library.member ACCESS(READ) ID(USERA) CLASS(FACILITY)
```

For TopSecret:

```
TSS PERMIT(USERA) IBMFAC($$TREE.library.member-name) ACC(READ)
```

For ACF2/SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($$TREE.dsn.member-name) TYPE(CMF)
 UID(USERA) ALLOW
```

Access a Report Under Screen U

The user's authority to issue an action (update, delete, and so on) on a certain report is checked with the following entity:

Table 50 Report Access

Action	Entity
Update a record	$$RECUPD.qname.userida
Insert a record	$$RECINS.qname.userida
Delete a record	$$RECDEL.qname.userida
Reprint a report	$$RECRPR.qname.userida
Restore a record	$$RECRPR.qname.userida
Use GIVETO option	$$GIVETO.qname.userid
Define a ruler	$$EXTENT.qname.userid
Suppress or activate a ruler	$$RULONF.qname.userid
Save a ruler definition	$$RULSAV.qname.userid
Use Global ruler	$$RULONF.qname.global-ruler
Immediate print for a report	$$RECIPR.qname.userid
View (browse) a report	$$VIEWCO.qname.userid
Permit report access without Recipient Tree	$$REPLST.qname.userid
Browse NOTES of a report	$$VIEWNO.qname.userid
Add/Update NOTES of a report	$$EDITNO.qname.userid

Action	Entity
Add NOTES to a report	$$ADDNOT.qname.userid
Update NOTES to a report	$$UPDNOT.qname.userid
Delete NOTES	$$DELNOT.qname.userid
Update Report View Indicator	$$VEWUPD.qname.userid
Cancel Restore for History Report	$$UNRSTR.qname.userid
Perform a recall of a migrated CDAM file	$$CHKRCL.qname.userid
Submit a job to perform recall of a migrated CDAM file	$$RECALL.qname.userid
View the report in hexadecimal format	$$RECHEX.qname.userid
Use parameter DREPLST, set to YES	$$REPLST.qname.recipient-name

Control-V:

Table 51 Control-V Features

Action	Entity
Use Control-V Quick Access features	$$CTVQAC.qname.userid
Use Control-V Indexing features	$$CTVINX.qname.userid

In the above entities, userid is the user ID to whom the report belongs.

To permit USERA to view (browse) a report that belongs to USERB, use the following commands:

For RACF:

```
RDEFINE FACILITY $$VIEWCO.qname.USERB UACC(NONE)
PERMIT $$VIEWCO.qname.USERB ACCESS(READ) ID(USERA) CLASS(FACILITY)
```

For TopSecret:

```
TSS ADD(system-dept) IBMFAC($$VIEWCO.qname.USERB)
TSS PERMIT(USERA) IBMFAC($$VIEWCO.qname.USERB) ACC(READ)
```

For ACF2/SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($$VIEWCO.qname.USERB) TYPE(CMF)
 UID(USERA) ALLOW
```

Limit Immediate Print of Reports

When user requests an immediate print for a report, and the number of pages for the report is more than DPAGMIN, an additional entity is checked. The entity structure is as follows:

Table 52 Report Limits

Entity	Description
$$PAGIII	If the number of pages is greater than DPAGMIN but less than or equal to parameter DPAGMID.
$$PAGII	If the number of pages is greater than DPAGMID but less than or equal to parameter DPAGMAX.
$$PAGI	If the number of pages is greater than DPAGMAX.

For RACF:

To allow USERA to immediately print a report of any size, use the following commands:

```
RDEFINE FACILITY $$PAGI* UACC(NONE)
PERMIT $$PAGI* CLASS(FACILITY) ID(USERA) ACCESS(READ)
```

To permit USERA to print reports that do not exceed the DPAGMAX number of pages, use the following commands:

```
RDEFINE FACILITY $$PAGII UACC(NONE)
PERMIT $$PAGII ID(USERA) CLASS(FACILITY) ACCESS(READ)
```

For TopSecret:

To allow USERA to immediately print a report of any size, use the following commands:

```
TSS ADD(system-dept) IBMFAC($$PAGI)
TSS PERMIT(USERA) IBMFAC($$PAGI) ACCESS(READ)
```

To permit USERA to print reports that do not exceed the DPAGMAX number of pages, use the following command:

```
TSS PERMIT(USERA) IBMFAC($$PAGI) ACCESS(READ)
```

For ACF2/SAF:

To allow USERA to immediately print a report of any size, use the following commands:

```
SET RESOURCE(CMF)
COMP
$KEY($$PAGI**) TYPE(CMF)
 UID(USERA) ALLOW
```

To permit USERA to print reports that do not exceed the DPAGMAX number of pages, use the following command:

```
SET RESOURCE(CMF)
COMP
$KEY($$PAGII*) TYPE(CMF)
 UID(USERA) ALLOW
```

Module CTDSE08

The CTDSE08 Control-D security module verifies the user’s authority to access the Active Missions screen (screen A) and perform actions (rerun, hold, delete, and so on) on missions displayed in this screen.

The CTDSE08 security module functions are performed under two modes of operation:

- Online: Active Missions screen (screen A).
- Batch: For example, a restore mission resets the status of the restore mission from "restore in process" to "ended."

IOA checks authorization:

- The class checked is FACILITY.
- The entity checked depends under what mode (Online Basic Definition, Online Extended Definition, or Batch) the module is invoked.

Online   Basic Definition Mode

Initial Entry to Screen A

IOA checks authorization:

- The class checked is FACILITY.
- The entity checked is $$CTDPNLA.qname

No distinction is made between the authority to perform actions on missions that are present in the Active Missions file, and the authority to submit a job.

This is equivalent to asking if the current user has the authority to submit jobs with USER parameter equal to that of the mission’s owner. If a user is authorized to submit a job on behalf of other users, then the user is also authorized to perform the specific action (hold, free, delete, and so on) on missions belonging to other users. If the mission’s owner is the current user, the security check is bypassed.

For RACF:

When an action is performed on missions that are present in the Active Missions file, the entity checked is owner.SUBMIT using the SURROGAT class.

For TopSecret:

When an action is performed on missions that are present in the Active Missions file, the TopSecret Application Interface module (TSSAI) is called with the following parameters:

Resource Class: ACIDCHK

Resource Name: ownerid

For ACF2/SAF:

When an action is performed on missions that are present in the Active Missions file, the entity checked is $SUBMIT.owner using the FACILITY class.

Extended Definition Mode

Initial Entry to Screen A

IOA checks authorization:

- The class checked is FACILITY.
- The entity checked is $$CTDPNLA.qname

Subsequent Operations in Screen A

The actions (hold, delete, rerun, and so on) are separated into different categories of access authority to the Active Missions screen (Screen A). The entity checked is $$MISxrrr.qname

where

- owner is the owner ID that is specified in the Mission Definition screen.
- x is a one digit action identifier.
- rrr is a three character identifier for each action (see the following table).

Table 53 CTDSE08 Action Identifiers

Action Identifier	Action Code	Description
1	ZOO LOG	Zoom Log
2	HLD RRN FRE	Hold Rerun Free
3	CHA DEL PPL UPD	Change Delete Print Update

To permit USERA to hold missions with owner of USERB, use the following command:

For RACF:

```
PERMIT $$MIS2HLD.qname.USERB ACCESS(READ) ID(USERA) CLASS(FACILITY)
```

For TopSecret:

```
TSS PERMIT(USERA) IBMFAC($$MIS2HLD.qname.USERB) ACC(READ)
```

For ACF2/SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($MIS2HLD.qname.USERB) TYPE(CMF)
 UID(USERA) ALLOW
```

Batch

The entity checked is $$CTDRRST.qname.owner

This entity is checked for batch jobs (Control-D and Control-V backup and restore jobs and Control-V migration jobs). In most cases, the batch jobs runs under the user ID of the Control-D and Control-V started tasks.

The user ID of the Control-D started tasks as specified in the Control-D installation procedure.

Use the following command to allow Control-D to access this entity:

For RACF:

```
PERMIT $$CTDRRST.qname.* ACCESS(READ) ID(controld-stc’s-userid)
CLASS(FACILITY)
```

For TopSecret:

```
TSS PERMIT(controld-acid) IBMFAC($$CTDRRST.qname) ACC(READ)
```

For ACF2/SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($$CTDRRST.qname) TYPE(CMF)
 UID(userid) ALLOW
```

where userid is one or more Control-D started tasks that are specified during Control-D installation.

Module CTDSE19

The CTDSE19 Control-D security module verifies the user’s authorization to perform actions on packets displayed under the File Transfer Control screen (screen F). In addition, this module verifies the user’s authorization to use the Control-D File Transfer facility to transfer reports from the mainframe to the PC.

Basic Definition Mode

Initial Entry to Screen F

IOA checks authorization:

- The class checked is FACILITY.
- The entity checked is $$CTDPNLF.qname.

Subsequent Operations in Screen F

For all actions (hold, delete, and so on) that are performed in this screen, IOA performs authorization. No distinction is made between the authority to perform actions on packets that are present in the Active Transfer file, and the authority to submit a job.

The check verifies that the current user who has the authority to submit jobs with a USER parameter is equal to that of the specific job being accessed. A user who is authorized to submit a job on behalf of others is also authorized to perform the specific action (hold, retransmit, print, delete, and so on) in screen F.

For RACF:

The CLASS checked is SURROGAT, and the entity checked is owner.SUBMIT

For TopSecret:

The TopSecret Application Interface module (TSSAI) is called with the following parameters:

Resource Class: ACIDCHK

Resource Name: ownerid

To permit USERA to perform an action on a packet with ownerid USERB, use command:

```
TSS PERMIT(USERA) ACID(USERB)
```

For ACF2/SAF:

To permit USERA to perform an action on a packet with ownerid USERB, specify the following commands:

```
SET RESOURCE(CMF)
COMP
$KEY(USERA.$SUBMIT) TYPE(CMF)
 UID(USERB) ALLOW
```

Extended Definition Mode

Initial Entry to Screen F

IOA checks authorization:

- The class checked is FACILITY.
- The entity checked is $$CTDPNLF.qname

Subsequent Operations in Screen F

For all actions (hold, delete, and so on) that are performed in this screen, IOA performs an authorization check. No distinction is made between the authority to perform actions on packets that are present in the Active Transfer file and the authority to submit a job.

The check verifies that the current user who has the authority to submit jobs with a USER parameter is equal to that of the specific job being accessed. A user who is authorized to submit a job on behalf of others is also authorized to perform the specific action (hold, retransmit, print, delete, and so on) in screen F.

The entity checked is $$DPCxrrr.qname.owner.

where

- owner is the owner ID that is specified in the Mission Definition screen.
- x is a one digit action identifier.
- rrr is a three character identifier for each action (see table below).

Table 54 Action Identifiers

Action Identifiers	Action Code	Description
1	VIE VIE	Getnext Getdir
2	HLD FRE	Hold Free
3	PRN DEL	Print Delete
4	TRN TRN TRN	Transfer Retransmit Confirm

To permit USERA to hold packets with owner ID of USERB, use following command:

For RACF:

```
PERMIT $$DPC2HLD.qname.USERB ACCESS(READ) ID(USERA) CLASS(FACILITY)
```

For TopSecret:

```
TSS PERMIT(USERA) IBMFAC($$DPC2HLD.qname.USERB) ACC(READ)
```

For ACF2/SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($$DPC2HLD.qname.USERB) TYPE(CMF)
 UID(USERA) ALLOW
```

Module CTDSE24

The CTDSE24 module is the security module of Control-D Exit CTDX024. This module is called when a request is made by Control-D/Page On Demand to access the Control-D Active User Report file or the Control-V Migrated User Report file. This module checks the user's authorization according to the mainframe logon user ID specified in the Control-D/WebAccess Server Communication Setup menu.

This module builds a filtered list of reports displayed on the user's screen, and verifies the user's authority to perform actions from Control-D/Page On Demand.

IOA verifies authorization for every action that is performed on a specific report. The CLASS checked is FACILITY (unless otherwise specified) and the entity used to check authorization depends on if Basic or Extended Definition mode is used.

When an attempt is made to access the Control-D and Control-V Active or Migrated User Report file, the CTDSE24 security module is called to check if the access is allowed. In this case, this security module does not perform any security checks. For performance reasons, the check on each screen line is not performed.

Basic Definition Mode

The CTDSE24 security module retrieves security definitions from the Recipient Tree. The administrator can authorize Control-D/Page On Demand users to view mainframe reports by adding the appropriate mainframe logon ID to the AUTHORIZE field in the recipient definitions in the Recipient Tree. These authorizations enable Control-D/Page On Demand users to see the reports of these recipients using Control-D/Page On Demand. For more information, see the Recipient Definition screen in the online facilities chapter of the *Control-D and Control V User Guide*.

When a mainframe logon ID is entered in the AUTHORIZE field of a recipient definition, the authorized Control-D/Page On Demand user can view all the reports of that recipient and descendants in the Recipient Tree. The same mainframe logon ID can be entered in the AUTHORIZE field of more than one recipient in the Recipient Tree.

The following rules apply to mainframe logon IDs entered in the AUTHORIZE field in a recipient definition:

- The specified logon ID is treated as a prefix if optional Wish WD2564 is set to YES in member IOADFLTC in the IOA MAC library.
- The specified mainframe logon ID can contain a number of "?" characters. This wildcard character indicates any single character.

Access a Report from Control-D/Page On Demand

When the user requests an action (view, print) on a certain report, the entity checked is $$CTDASR.qname.userid, where userid is the user name related to the report being accessed.

There is no distinction between the different actions that can be specified. The user is either allowed to perform any valid action with the report or completely denied access to the report.

To permit USERA (the mainframe logon ID) to perform actions to the reports of USERB (the Control-D recipient name), use the following command:

For RACF:

```
RDEFINE FACILITY $$CTDASR.qname.USERB UACC(NONE)
PERMIT $$CTDASR.qname.USERB ACCESS(READ) ID(USERA) CLASS(FACILITY)
```

For TopSecret:

```
TSS PERMIT(USERA) IBMFAC($$CTDASR.qname.USERB) ACC(READ)
```

For ACF2/SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($$CTDASR.qname.USERB) TYPE(CMF)
 UID(USERA) ALLOW
```

Limit Immediate Print of Reports

When the user requests immediate print, and the report contains more than the minimum number of pages specified in parameter DPAGMIN, the following entity is checked to verify that the user is authorized to send the number of pages contained in the report to the printer:

Table 55 Print Limits

Entity	Description
$$PGASRIII	Checked when the number of pages is greater than parameter DPAGMIN and less than or equal to parameter DPAGMID.
$$PGASRII	Checked when the number of pages is greater than parameter DPAGMID and less than or equal to parameter DPAGMAX.
$$PGASRI	Checked when the number of pages is greater than parameter DPAGMAX.

For RACF:

To allow USERA to immediately print a report of any size, use the following commands:

```
RDEFINE FACILITY $$PGASRI UACC(NONE)
PERMIT $$PGASRI CLASS(FACILITY) ID(USERA) ACCESS(READ)
```

To permit USERA to print reports that do not exceed the number of pages specified in parameter DPAGMAX, use the following commands:

```
RDEFINE FACILITY $$PGASRII UACC(NONE)
PERMIT $$PGASRII ID(USERA) CLASS(FACILITY) ACCESS(READ)
```

For TopSecret:

To allow USERA to immediately print a report of any size, use the following commands:

```
TSS PERMIT(USERA) IBMFAC($$PGASRI) ACC(READ)
```

For ACF2/SAF:

To allow USERA to immediately print a report of any size, use the following commands:

```
SET RESOURCE(CMF)
COMP
$KEY($$PGASRI) TYPE(CMF)
 UID(USERA) ALLOW
```

Extended Definition Mode

The CTDSE24 security module retrieves security definitions from the Recipient Tree. The administrator can authorize Control-D/Page On Demand users to view mainframe reports by adding the appropriate mainframe logon ID to the AUTHORIZE field in the recipient definitions in the Recipient Tree. For information about how this is done, see Basic Definition Mode (on page 107).

Access a Report From Control-D/Page On Demand

The user’s authority to issue an action (update, delete, and so on) on a certain report is checked with the following entities:

Table 56 Report Access

Action	Entity
Update report view indicator	$$UPDASR.qname.userid
View a report in browse mode	$$VIEASR.qname.userid
Immediate printing of a report	$$IPRASR.qname.userid
Show notes of a report	$$SHNASR.qname.userid
Add a note	$$ADNASR.qname.userid
Delete a note	$$DLNASR.qname.userid
Update a note	$$UPNASR.qname.userid
View a note	$$VWNASR.qname.userid
Edit a note	$$EDNASR.qname.userid
Restore a report or record	$$RSTASR.qname.userid
Delete a record	$$RDLASR.qname.userid
Update a record	$$UPRASR.qname.userid
Use parameter DREPLST set to YES	$$REPLST.qname.recipient-name

In the above entities, userid is the user ID to whom the report belongs.

To permit USERA (meaning, the mainframe logon ID) to view (browse) a report that belongs to USERB (meaning, the Control-D recipient name), use the following command:

For RACF:

```
RDEFINE FACILITY $$VIEASR.qname.USERB UACC(NONE)
PERMIT $$VIEASR.qname.USERB ACCESS(READ) ID(USERA) CLASS(FACILITY)
```

For TopSecret:

To allow USERA to immediately print a report of any size, use the following commands:

```
TSS PERMIT(USERA) IBMFAC($$VIEASR.qname.USERB) ACC(READ)
```

For ACF2/SAF:

To allow USERA to immediately print a report of any size, use the following commands:

```
SET RESOURCE(CMF)
COMP
$KEY($$VIEASR.qname.USERB) TYPE(CMF)
 UID(USERA) ALLOW
```

Limit Immediate Print of Reports

When the user requests immediate print, and the report contains more than the minimum number of pages specified in parameter DPAGMIN, the following entity is checked to verify that the user is authorized to send to the printer the number of pages contained in the report:

Table 57 Report Limits

Entity	Description
$$PGASRIII	Checked when the number of pages is higher than DPAGMIN and lower than parameter DPAGMID.
$$PGASRII	Checked when the number of pages is higher than DPAGMID and lower than parameter DPAGMAX.
$$PGASRI	Checked when the number of pages is higher than DPAGMAX.

For RACF:

To allow USERA to immediately print a report of any size, use the following commands:

```
RDEFINE FACILITY $$PGASRI UACC(NONE)
PERMIT $$PGASRI CLASS(FACILITY) ID(USERA) ACCESS(READ)
```

To permit USERA to print reports that do not exceed the DPAGMAX number of pages, use the following commands:

```
RDEFINE FACILITY $$PGASRII UACC(NONE)
PERMIT $$PGASRII ID(USERA) CLASS(FACILITY) ACCESS(READ)
```

To allow USERA to immediately print a report of any size, use the following commands:

```
TSS PERMIT(USERA) IBMFAC($$PGASRI) ACC(READ)
```

For ACF2/SAF:

To allow USERA to immediately print a report of any size, use the following commands:

```
SET RESOURCE(CMF)
COMP
$KEY($$PGASRI) TYPE(CMF)
 UID(USERA) ALLOW
```

Module CTDSE28

The CTDSE28 module is the security module of Control-D Exit CTDX028. This module is called when a user attempts to enter to any option in screen DO (Control-D Objects) in the IOA primary menu, in addition this module checks the user’s authorization to create, save or delete a report clique or a resource set.

IOA verifies authorization for every action that is performed on a specific report clique or resource set. The CLASS checked is FACILITY (unless otherwise specified) and the entity used to check authorization depends on action the user attempts to do.

Access to option 1 of the DO screen (Report Clique)

When the user attempts to enter to this option the entity checked is $$CTDOBJ.qname.ENTRY.REPCLQ. To permit USERA (the mainframe logon ID) to enter to this option, use the following command:

For RACF:

```
RDEFINE FACILITY $$CTDOBJ.qname.ENTRY.REPCLQ UACC(NONE)

PERMIT $$CTDOBJ.qname.ENTRY.REPCLQ ACCESS(READ) ID(USERA)

CLASS(FACILITY)
```

For TopSecret:

```
TSS PERMIT(USERA) IBMFAC($$CTDOBJ.qname.ENTRY.
 REPCLQ)  ACC(READ)
```

For ACF2/SAF:

```
SET RESOURCE(CMF)

COMP

$KEY($$CTDOBJ.qname.ENTRY.REPCLQ) TYPE(CMF)

 UID(USERA) ALLOW
```

Access to option 2 of the DO screen (Resource Set)

When the user attempts to enter to this option, the entity checked is $$CTDOBJ.qname.ENTRY.RESSET. To permit USERA (the mainframe logon ID) to enter to this option use the following command:

For RACF:

```
RDEFINE FACILITY $$CTDOBJ.qname.ENTRY.RESSET UACC(NONE)

PERMIT $$CTDOBJ.qname.ENTRY.RESSET ACCESS(READ) ID(USERA)
```

CLASS(FACILITY)

For TopSecret:

```
TSS PERMIT(USERA) IBMFAC($$CTDOBJ.qname.ENTRY. RESSET)
ACC(READ)
```

For ACF2/SAF:

```
SET RESOURCE(CMF)

COMP

$KEY($$CTDOBJ.qname.ENTRY.RESSET) TYPE(CMF)

 UID(USERA) ALLOW
```

➢ To save a new report clique or a modified report clique

When the user attempts to create or to save a report clique the entity checked is $$CTDOBJ.qname.SAVE.report-clique-name.

To permit USERA (the mainframe logon ID) to perform this option use the following command:

For RACF:

```
RDEFINE FACILITY $$CTDOBJ.qname.SAVE.report-clique-name UACC(NONE)

PERMIT $$CTDOBJ.qname.SAVE.report-clique-name ACCESS(READ) ID(USERA)
```

CLASS(FACILITY)

For TopSecret:

```
TSS PERMIT(USERA) IBMFAC($$CTDOBJ.qname.SAVE.report-clique-name) ACC(READ)
```

For ACF2/SAF:

```
SET RESOURCE(CMF)

COMP

$KEY($$CTDOBJ.qname.SAVE.report-clique-name) TYPE(CMF)

 UID(USERA) ALLOW
```

➢ To delete a report clique

When the user attempts to delete a report clique the entity checked is $$CTDOBJ.qname.DELETE.report-clique-name. To permit USERA (the mainframe logon ID) to perform this option use the following command:

For RACF:

```
RDEFINE FACILITY $$CTDOBJ.qname.DELETE.report-clique-name UACC(NONE)

PERMIT $$CTDOBJ.qname.DELETE.report-clique-name ACCESS(READ) ID(USERA)

CLASS(FACILITY)
```

For TopSecret:

TSS PERMIT(USERA) IBMFAC($$CTDOBJ.qname.DELETE.report-clique-name) ACC(READ)

For ACF2/SAF:

```
SET RESOURCE(CMF)

COMP

$KEY($$CTDOBJ.qname.DELETE.report-clique-name) TYPE(CMF)

 UID(USERA) ALLOW
```

➢ To delete a resource set

When the user attempts to delete a resource set the entity checked is $$CTDOBJ.qname.DELETE.resource-set-name. To permit USERA (the mainframe logon ID) to perform this option use the following command:

For RACF:

```
RDEFINE FACILITY $$CTDOBJ.qname.DELETE.resource-set-name UACC(NONE)

PERMIT $$CTDOBJ.qname.DELETE.resource-set-name ACCESS(READ) ID(USERA)

CLASS(FACILITY)
```

For TopSecret:

```
TSS PERMIT(USERA) IBMFAC($$CTDOBJ.qname.DELETE.resource-set-name) ACC(READ)
```

For ACF2 / SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($$CTDOBJ.qname.DELETE.resource-set-name) TYPE(CMF)
 UID(USERA) ALLOW
```

4

Control-O Security

This section describes the procedure used to implement the Control-O security interface. It is recommended that you first review the following information about the elements that are protected in Control-O and then proceed to the step-by-step instructions.

Protecting Control-O Elements:

The security interface protects the following Control-O elements:

- Ordering rules.
- Triggering rules.
- Use of DO statements and ON statements. Controls the authority of the owner of a rule to specify DO statements or ON statements (for example, access or modify prerequisite conditions, issue certain operator commands, issue restricted TSO commands).
- Use of the Rule Status screen (screen OS) and authority to access rules within this screen.
- Execution of DO statements according to the authorization of the user ID associated with the rule. The user ID can be the owner of the rule or the user ID that issued the message or command that triggered the rule.
- Execution of TSO commands and KOA scripts by applying the security specification associated with the rule for the command execution.
- Use of the Automation Options screen (screen OA) and authority to perform actions on Automation Options entities.
- Use of the XAM Interface. Verifies the authority of the user to execute a function using the XAM Interface.

Rule Ordering:

Each Control-O rule is defined with an OWNER parameter. The OWNER is the user ID to which the rule belongs. To order a rule, the user must have authorization to access the owner of the rule. The CTOSE01 Control-O security module verifies that the current user has the authorization to order the rule, using the OWNER field of the rule.

The CMEM default rules in the IOACMEMR table are provided with the OWNER user ID of IOADMIN.

You must grant the user who orders these rules (either CTMCMEM or CONTROLO) permission to load the rules on behalf of the IOADMIN user ID, and grant the IOADMIN user permissions to the perform ON and DO statements in these rules.

Rule Triggering:

Events such as messages, commands, and the other event types defined by the ON statement can cause rules to be triggered if a rule with a matching ON statement is active.

Before triggering the rule, the Control-O CTOSE10 security module validates that the user associated with the event is authorized to trigger the rule. This protection is selective and it is performed if the feature is enabled and only for those rules that request it explicitly by specifying the following anywhere inside the rule:

DO SET = %%PROTRULE=Y

There is no security exit CTOX010 and there is no distinction between basic and extended security mode in security module CTOSE10.

Use of Control-O Functions (DO Statements and ON Statements): Rule execution consists of reacting to the events defined in the rule’s ON statement and performing actions defined in the rule’s DO statement. The security interface verifies if the owner of the rule is allowed to perform these actions. The CTOSE02 Control-O security module performs an authority check for each rule statement before the rule is loaded. If one of the authority checks fails, the entire rule load is canceled.

Access to and Use of the Rule Status Screen:

The Rule Status screen lists the active rules currently handled by Control-O and their status. The user can issue inquiries about a rule in the list or change its status. The CTOSE08 Control-O security module verifies the user’s authorization to access the Rule Status screen and perform actions (hold, delete, and so on) on the rules displayed.

Use of Control-O Commands (DO Statements):

Rule execution consists of performing various actions defined in DO statements. The security interface verifies the authority of the user ID associated with the rule to execute each of the DO statements in the rule. The user ID associated with the rule can be the owner of the rule or the user ID of the user who issued the message or command that triggered the rule, depending on the value of the rule’s RUNTSEC parameter.

The IOASE07 security module checks for authorization to update conditions. The IOASE12 security module checks for authorization to execute operator commands. The CTOSE03 security module checks for authorization to execute DO KOA and DO TSO statements.

Use of TSO Commands and KOA Scripts:

A rule can request execution of a TSO command (or REXX/CLIST) or activation of a KOA script. The command or script can access datasets and various resources in the system. To protect these resources, the command’s execution environment inherits the security environment associated with the rule. This means that the command’s successful execution is dependent on if the user ID associated with the rule has the authority to execute the command. This user ID is either the owner of the rule or the user ID of the user who issued the message or command that triggered the rule, depending on the value of the rule’s RUNTSEC parameter.

Access to the Automation Options Screen:

The Automation Options screen (screen OA) handles various aspects of the Automation environment (for example, issuing operator commands, listing and controlling Control-O servers, checking resource queuing information, viewing the operator console display). The CTOSE04 Control-O security module verifies user authorization to access Automation Options and protects actions performed on entities handled in these screens.

Use of the XAM Interface:

A TSO user, REXX, CLIST or user–written program can request services from the XAM interface. The security interface verifies the authority of the user ID associated with the XAM request to execute the requested function.

When XAM functions are requested under a Control-O server, the user ID is either the OWNER of the rule or the requester (TRIGGER) of the XAM function, depending on the value of rule’s RUNTSEC parameter.

Implementing Control-O Security

This chapter details the steps required to implement the Control-O security interface.

The Control-O security interface can be installed either as part of the customized installation path, or during the Customization process after installation. Both options use the INCONTROL Installation and Customization Engine (ICE) application. If you are not familiar with the ICE interface, see the *INCONTROL for z/OS Installation Guide: Installing*.

The Control-O security interface cannot be implemented until IOA security is implemented. Verify that IOA security is installed before implementing Control-O security.

For Control-M Event Manager (CMEM) users:
If CMEM security is already implemented, it is not necessary to implement Control-O security. Part of the Control-O security implementation is already handled using CMEM security. However, it is necessary to review all the required security definitions described below to protect the additional elements in Control-O.

➢ To install the Control-O security interface

- **a.** Enter the main ICE screen.
- **b.** Select Customization.
- **c.** Enter CTO in the Product field.
- **d.** Select Security Customization.
- **e.** Perform all major and minor steps required to install the security product.

Step 1 Implement Control-O Security

Follow the steps below to implement Control-O security.

Step 1.1 Grant Access Permissions

1. Collect the data you need to define the INCONTROL entities and user authorizations to the security product.
2. In ICE, run the steps "Control-O Security Definitions (Sample)" and "Functions Security Definitions (Sample)" to create two sample jobs.
3. Submit the jobs to define security to IOA and Control-O.

Step 1.2 Customize Security Parameters

Table 58 Security Parameters

Parameter	Description
DEFMCHKO	When choosing a definition mode as COND to any of the Control-O security modules, use qname together with the value given to this parameter as the high level qualifier, to determine the real definition mode to be used.
SECTOLO	Determine which action to perform if your security product is inactive or a specific resource is not defined in the security product. Valid values are: ▪ YES — Perform the action. ▪ NO — Do not perform the action.
Mode Definition	Specify one of the following values to determine the definition mode for Control-O security modules: ▪ COND — Conditional Definition mode. Default. ▪ BASIC — Basic Definition mode. ▪ EXTEND — Extended Definition mode.
DFMO01	Definition mode for the CTOSE01 Control-O security module.
DFMO02	Definition mode for the CTOSE02 Control-O security module.
DFMO03	Definition mode for the CTOSE03 Control-O security module.
DFMO04	Definition mode for the CTOSE04 Control-O security module.
DFMO08	Definition mode for the CTOSE08 Control-O security module.
DFMO10	Definition mode for the CTOSE10 Control-O security module.
DFMO15	Definition mode for the CTOSE15 Control-O security module.

Step 1.3 Save Security Parameters into Product

This step saves all the security parameters specified for Control-O.

When this step completes, the Status column is automatically updated to COMPLETE.

Step 2. RACF Security Definition Samples

Step 2.1 Control-O Security Definitions

Step 2.2 Function Security Definitions

Step 2.3 Control Program Access to Datasets

Select these steps to edit members CTOSRAC2, CTOSRAC3, and CTOSRAC4.

Perform the following steps to define the required permissions.

1. Associate Users with Extended Definition Mode
   a. To define the entity $$CTOEDM.qname, use the following command:

      RDEFINE FACILITY $$CTOEDM.qname UACC(NONE)

   b. To authorize USERA to Extended Definition mode, use the following command:

      PERMIT $$CTOEDM.qname ID(USERA) CLASS(FACILITY) ACCESS(READ)

   c. Submit the CTOSRAC2 job.

      This job must be run under the user ID of an administrator who has authorization to enter these commands.

   d. Scan the output of the job for information and error messages. All job steps must end with a condition code of 0.
2. Define entities and user authorizations.

   For more information about entities and user authorizations, see Control-O Basic Definition Security Calls (on page 135), and Control-O Extended Definition Security Calls (on page 139).

To define and authorize the entity in Extended Definition mode to protect ordering of Control-O rules beginning with SYS, specify the following command:

```
RDEFINE FACILITY $$CTOORD.qname.SYS* UACC(NONE)
PERMIT $$CTOORD.qnam.SYS* CLASS(FACILITY) ID(USERA) ACCESS(READ)
```

where qname is the name used to assign different authorizations to different Control-O environments (for example, Test and Production). This parameter is specified during IOA installation.

To authorize USERA access to a given Control-O entity, use the following command:

```
PERMIT $$CTOnnn.qname CLASS(FACILITY) ID(USERA) ACCESS(READ)
```

where CTOnnn is the name of the Control-O entity to be accessed.

All entity names for each Control-O protected element appear in Basic Definition Mode (on page 146) and Extended Definition Mode (on page 148).

For samples of user authorizations, review member CTOSRAC3 in the IOA INSTWORK library.

Step 3. TopSecret Security Definition Samples

Step 3.1 Control-O Security Definitions

Step 3.2 Function Security Definitions

Step 3.3 Control Program Access to Datasets

Step 3.4 Define CTO to TopSecret Facility Matrix

Select these steps to edit members CTOSTSS2, CTOSTSS3, CTOSTSS4, and CTOSTSS5.

Perform the following steps to define the required permissions.

1. Define Control-O to the TopSecret Facility Matrix

   The Control-O monitor must be defined in the TopSecret Facility Matrix. The CTOSTSS2 member in the IOA INSTWORK library contains the necessary command to dynamically define Control-O in the TopSecret Facility Matrix.

   a. Modify USER4 in the Facility definition command to a free entry in the Facility Matrix, as follows:

      TSS MODIFY FAC(USER4=NAME=CTO)

      This command defines Control-O in the Facility Matrix until the next IPL.

   b. To permanently define the facility, update the TopSecret parameter member. This member is usually called TSSPARM0.

   c. Copy the Control-O facility definition from member CTOSTSS5 in the IOA INSTWORK library to member TSSPARM0.

   d. Update the Facility Matrix entry name with the same name that is specified in the TSS MODIFY command above.

2. Define Control-O ACID to TopSecret by changing the value of parameter DEPT from sec-administrator-dept to the appropriate ACID: as follows:

   TSS CRE(CONTROLO) NAME (...) DEPT(sec-administrator-dept)

3. Define Control-O started tasks to TopSecret by changing the ACID definition in the following commands to the appropriate ACID:

   TSS ADD(STC) PROC(CONTROLO) ACID(CONTROLO)

4. Allow Control-O ACID to Control-O datasets.

   Authorizations to access Control-O datasets are defined during the Control-O installation process. This step must be completed before proceeding with security implementation. For information about how to grant users access to Control-O datasets, see the Control-O chapter of the *INCONTROL for z/OS Installation Guide: Installing*.

5. Connect the appropriate profile to the Control-O ACID with the following command:

   TSS ADD(CTO) PROF (profile-name)

6. Define entities and user authorizations in TopSecret

For information about entities and user authorizations, see Control-O Basic Definition Security Calls (on page 135) and Control-O Extended Definition Security Calls (on page 139).

   a. Modify the following command to establish ownership of the resources in TopSecret to the appropriate owner:

      TSS ADD(sec-administrator-dept) IBMFAC($$CTO)

For samples of user authorizations, review member CTOSTSS3 in the IOA INSTWORK library.

   All entity names for each Control-O protected element appear in Control-O Basic Definition Security Calls (on page 135) for Basic Definition mode and in Control-O Extended Definition Security Calls (on page 139) for Extended Definition mode.

7. Associate users with Extended Definition Modes

a. Customize the following TopSecret command to establish Extended Definition mode for the Control-O installer.

```
TSS PERMIT(USERA) IBMFAC($$CTOEDM.qname) ACC(READ)
```

b. Modify USERA to the UID of Control-O installer.

Do not define the $$CTOEDM entity to operate in warning mode because this causes all users to operate in Extended Definition mode.

8. Authorize the Control-O installer to use Control-O facilities.

   a. Customize the following command to authorize USERA access Control-O:

   ```
   TSS ADD(USERA) IBMFAC($$CTO)
   ```

   b. Modify USERA to the user ID of the Control-O installer.

   c. Customize the following command to authorize the Control-O installer to use Control-O facilities:

   ```
   TSS PERMIT(USERA) IBMFAC($$CTO) ACC(READ)
   ```

9. Submit the job.

   This job must be run under the ACID of the general security administrator (SCA) who has authorization to enter these TopSecret commands.

   All job steps must end with a condition code of 0.

Step 4. ACF2 Security Definition Samples

Step 4.1 Control-O Security Definitions

Step 4.2 Function Security Definitions

Step 4.3 Control Program Access to Datasets

Select this step to edit member CTOSSAF2, CTOSSAF3, and CTOSSAF4 in the IOA INSTWORK library.

1. Define Control-O started tasks under ACF2.

   a. Define the Control-O started tasks (CONTROLO and the Control-O servers CTOSRVxx) as valid started tasks under ACF2.

   b. Add the multi-user address space (MUSSAS) parameter to the logon ID record that is created for the Control-O started task.

2. Associating users with Extended Definition Mode.

   a. Edit member CTOSSAF2 in the IOA INSTWORK library, add the following ACF2 commands to define the $$CTOEDM entity to ACF2/SAF, and authorize users to this entity.

   b. Define and authorize the entity $$CTOEDM.qname to ACF2 using the following commands:

   ```
   SET RESOURCE(CMF)
   COMP
   $KEY($$CTOEDM.qname)
    UID(USERA) ALLOW
   ```

3. Define Entities and User Authorizations to CA-ACF2/SAF

For more information about entities and user authorizations, see Control-O Basic Definition Security Calls (on page 135), and Control-O Extended Definition Security Calls (on page 139).

To authorize USERA (the user ID of the Control-O installer) access to a given Control-O entity, use the following command:

```
SET RESOURCE(CMF)
COMP
$KEY($$CTOnnn.qname) TYPE(CMF)
 UID(USERA) ALLOW
```

where qname is the name used to assign different authorizations to different Control-O environments (such as Test and Production). This parameter is specified during IOA installation.

Change USERA to the UID string of the Control-O installer.

All entity names for each Control-O protected element appear in Control-O Basic Definition Security Calls (on page 135) for Basic Definition mode and Control-O Extended Definition Security Calls (on page 139) for Extended Definition mode.

For samples of user authorizations, review the CTOSSAF3 member in the IOA INSTWORK library.

4. Submit the Job

This job must be run under a user of an ACF2 administrator who has authorization to enter these ACF2 commands.

Scan the output of the job for information and error messages produced by ACF2. All job steps must end with a condition code of 0.

Control-O Security Interface Modules

This section describes the Control-O security interface modules.

Control-O Basic Definition Security Calls

Table 59 Control-O Basic Definition Security Calls

Protected Element	Class Entity Name	Explanation	Security Module
Controlling Rule Ordering			
	SURROGAT owner.SUBMIT ACIDCHK owner FACILITY $SUBMIT.owner	owner is the owner of the rule.	CTOSE01
Controlling Use of Control-O Commands			

Protected Element	Class Entity Name	Explanation	Security Module
	FACILITY ON COMMAND $$IOACMD.qname.cmd-text ON CTOPCMSG $$CTOPCM.qname.msg-txt ON DSNEVENT $$CTODSN.qname.jobname ON EVENT $$CTOENV.qname.event-name ON JOBARRIV $$CTOJAR.qname.jobname ON JOBEND $$CTOJED.qname.jobname ON JOBSYSOUT $$CTOJSO.qname.jobname ON MESSAGE $$CTOMSG.qname.msg-id ON MESSAGE $$CTOONM.qname.msg-string ON OMEGAEXP $$CTOOMG.qname.exception ON RULE $$CTORUL.qname.owner.rule ON STEP $$CTOSTP.qname.jobname	qname is the name used to assign different authorizations to various Control-O environments (for example, Test and Production). cmd-text is the first 21 chars of command text. msg-txt is the first 21 chars of the message text. jobname is the job name in the ON statement. event-name is the "name" of the event. jobname is the job name in the ON statement. jobname is the job name in the ON statement. jobname is the job name in the ON statement. msg-id is the MVS message ID. msg-string is the first 21 chars of MVS message text. exception is the exception code. rule is the rule name in the ON statement. jobname is the job name in the ON statement.	CTOSE02

Protected Element	Class Entity Name	Explanation	Security Module
	FACILITY Runtime security option $$CTORTS.qname.runtime-sec DO ASKOPER $$CTOASK.qname.wtor-text DO COMMAND $$IOACMD.qname.cmd-text DO COND or RESOURCE $$IOARES.qname.res-name DO CTOPCMSG $$CTOPCM.qname.msg-text DO DISPLAY with SUPPRESS set to NO $$CTOMSG.qname.new-msg-txt DO DISPLAY with SUPPRESS set to YES $$CTOMSG.qname.new-msg-txt DO DOM $$CTODOM.qname DO TSO $$CTOTSO.qname.comnd DO FORCEJOB $$CTOCMO.qname.lib-name.tbl DO KSL $$CTOKSL.qname.ksl-name DO RULE $$CTORUL.qname.ownr.rule DO SET $$CTOSET.qname.var-name DO STOPJOB $$CTOJST.qname DO SYSREQ $$CTOSRQ.qname.sysreq-type	runtime-sec is the RUNTSEC value. Valid values: TRIGGER, OWNER, NONE. wtor-text is the first 21 chars of the WTOR name. cmd-text is the first 21 chars of command in the DO statement. res-name is the first 21 chars of the condition or resource name in the DO statement. msg-text is the first 21 chars of command in the DO statement. new-msg-txt is the first 21 chars of message text. new-msg-txt is the first 21 chars of message text. comnd is the first 2 characters of the command. lib-name is the first 21 characters of the Control-M schedule library. tbl is the member name in the Control-M schedule library. The whole entity name is truncated by RACF to 39. This means that tbl will be entirely truncated unless lib-name is less than 21. ksl-name is the first 21 chars of KSL name in the DO statement. ownr is the value of the DO RULE owner parameter. rule is the name of the rule in statement DO RULE. var-name is the first 21 chars of the IOA AutoEdit variable. sysreq-type is the SYSREQ option. Valid value: ENQINFO.	CTOSE02

Protected Element	Class Entity Name	Explanation	Security Module
Controlling Use of TSO Commands and KOA Scripts			
DO TSO	FACILITY $$CTOPRC.qname.envprc $$CTOTSO.qname.envprc		CTOSE03
DO KSL	FACILITY $$CTOPRC.qname.envprc $$CTOKSL.qname.envprc		
Controlling Access to and Use of the Automation Options Screen			
Access to Automation Options screen	FACILITY $$CTOAOP.qname.optnam.ENTRY	optnam is the name of the Automation Option selected under screen OA.	CTOSE04
Use of Automation Options screen	FACILITY $$CTOAOP.qname.optnam.obj	obj is the text (1 – 8 chars) identifying the object on which action was performed. act is the action (option) selected in screen optnam.	CTOSE04
Controlling Access to and Use of the Rule Status Screen			
Initial access to Rule Status screen	FACILITY $$CTOPNLOS.qname		CTOSE08
Controlling actions on rules	SURROGAT owner.SUBMIT ACIDCHK owner FACILITY $SUBMIT.owner	owner is the owner of the rule.	CTOSE08

Protected Element	Class Entity Name	Explanation	Security Module
Controlling Access to Services Provided using the XAM (Extended Automation Mechanism)			
All actions listed below	FACILITY $$CTOXAMF.qname		CTOSE15
INIT action			CTOSE15
TERM action			CTOSE15
RESOLVE action			CTOSE15
SETOLOC action			CTOSE15
SETOGLB action			CTOSE15
DORULE action			CTOSE15

Control-O Extended Definition Security Calls

Table 60 Control-O Extended Definition Security Calls

Protected Element	Class Entity Name	Explanation	Security Module
Controlling Rule Ordering			
	FACILITY $$CTOORD.qname.owner	qname is the name used to assign different authorizations to various Control-O environments (for example, Test and Production). owner is the owner of the rule.	CTOSE01

Protected Element	Class Entity Name	Explanation	Security Module
Controlling Use of Control-O Commands			
	FACILITY ON COMMAND $$CTOONC.qname.cmd-text ON CTOPCMSG $$CTOONP.qname.msg-txt ON DSNEVENT $$CTODSN.qname.jobname ON EVENT $$CTOENV.qname.event-name ON JOBARRIV $$CTOJAR.qname.jobname ON JOBEND $$CTOJED.qname.jobname ON MESSAGE $$CTOONM.qname.msg-id ON MESSAGE $$CTOONM.qname.msg-string ON RULE $$CTOORL.qname.owner.rule ON STEP $$CTOSTP.qname.jobname ON OMEGAEXP $$CTOOMG.qname.exception ON JOBSYSOUT $$CTOJSO.qname.jobname ON CTOPCMSG $$CTOONP.qname.msg-txt	cmd-text is the first 21 chars of command text. msg-txt is the first 21 chars of the message text. jobname is the job name in the ON statement. event-name is the "name" of the event. jobname is the job name in the ON statement. jobname is the job name in the ON statement. msg-id is the MVS message ID. msg-string is the first 21 chars of MVS message text. rule is the rule name in the ON statement. jobname is the job name in the ON statement. exception is the exception code. jobname is the job name in the ON statement. msg-txt is the first 21 chars of the message text.	CTOSE02

Protected Element	Class Entity Name	Explanation	Security Module
	FACILITY Runtime security option $$CTORTS.qname.runtime-sec DO ASKOPER $$CTOASK.qname.wtor-text DO COMMAND $$CTOCMD.qname.cmd-text DO COND or RESOURCE $$CTORES.qname.res-name DO CTOPCMSG $$CTOPCM.qname.msg-text DO DISPLAY with SUPPRESS set to NO $$CTODSP.qname.new-msg-txt DO DISPLAY with SUPPRESS set to NO $$CTODSP.qname.new-msg-txt DO DOM $$CTODOM.qname DO TSO $$CTOTSO.qname.comnd DO FORCEJOB $$CTOCMO.qname.lib-name.tbl DO KSL $$CTOKSL.qname.ksl-name DO RULE $$CTODRL.qname.ownr.rule DO SET $$CTOSET.qname.var-name DO STOPJOB $$CTOJST.qname DO SYSREQ $$CTOSRQ.qname.sysreq-type	runtime-sec is the RUNTSEC value. Valid values: TRIGGER, OWNER, NONE. wtor-text is the first 21 chars of the WTOR name. cmd-text is the first 21 chars of command in the DO statement. res-name is the first 21 chars of the condition or resource name in the DO statement. msg-text is the first 21 chars of command in the DO statement. new-msg-txt is the first 21 chars of message text. new-msg-txt is the first 21 chars of message text. comnd is the first 2 characters of the command. lib-name is the first 21 characters of the Control-M schedule library. tbl is the member name in the Control-M schedule library. The whole entity name is truncated by RACF to 39. This means that tbl will be entirely truncated unless lib-name is less than 21. ksl-name is the first 21 chars of KSL name in the DO statement. ownr is the value of the DO RULE owner parameter. rule is the name of the rule in statement DO RULE. If the OWNER parameter in the DO RULE statement is empty, then the "owner" in the $$CTODRL.qname.ownr.rule entity will be empty and the entity name will only consist of $$CTODRL.qname.rule. var-name is the first 21 chars of the IOA AutoEdit variable. sysreq-type is the SYSREQ option. Valid value: ENQINFO.	CTOSE02

Protected Element	Class Entity Name	Explanation	Security Module
Controlling Use of TSO Commands and KOA Scripts			
DO TSO	FACILITY $$CTOPTS.qname.envprc.cmd-txt	cmd-txt is the first 14 chars of the command text in the DO statement.	CTOSE03
DO KSL	FACILITY $$CTOPKS.qname.envprc.cmd-txt	cmd-txt is the first 14 chars of the command text in the DO statement.	CTOSE03
Controlling Access to and Use of the Automation Options Screen			
Access to Automation Options screen	FACILITY $$CTOAOP.qname.optnam.ENTRY	optnam is the name of the Automation Option selected under screen OA.	CTOSE04
Use of Automation Options screen	FACILITY $$CTOAOP.qname.optnam.obj.act	obj is the text (1 – 8 chars) identifying the object on which action was performed. act is the action (option) selected in screen optnam.	CTOSE04

Protected Element	Class Entity Name	Explanation	Security Module
Controlling Access to and Use of the Rule Status Screen			
Initial access to Rule Status screen	FACILITY $$CTOPNLOS.qname		CTOSE08
Controlling actions on rules	FACILITY Zoom: $$RUL1ZOO.qname.owner Log: $$RUL1LOG.qname.owner Resume: $$RUL2RES.qname.owner Hold: $$RUL2HLD.qname.owner Free: $$RUL2FRE.qname.owner Mode: $$RUL2MOD.qname.owner Delete: $$RUL3DEL.qname.owner Cancel: $$RUL3CAN.qname.owner	owner is the owner of the rule.	CTOSE08
Controlling Access to Services Provided using the XAM (Extended Automation Mechanism)			
	FACILITY		CTOSE15
INIT action	$$CTOXAM.qname.TYPE1INI		
TERM action	$$CTOXAM.qname.TYPE1TRM		
RESOLVE action	$$CTOXAM.qname.TYPE1RSL		
SETOLOC action	$$CTOXAM.qname.TYPE2LOC		
SETOGLB action	$$CTOXAM.qname.TYPE3GLB		

Protected Element	Class Entity Name	Explanation	Security Module
DORULE action	$$CTOXAM.qname.TYPE3RUL or $$CTOXAM.qname.TYPE3RUL.rule name See Optionally including the target rule name in the checked entity (on page 157).		

Module CTOSE01

The CTOSE01 module is the security module of Control-O Exit CTOX001. It is used to verify that the user is authorized to order the Control-O rule. A security check is issued to verify if the logged on (current) user is allowed to order rules on behalf of the user ID specified in the OWNER field of the rule definition.

In TSO or ROSCOE/ETSO, the CTOSE01 module is executed under the address space of the logged on user. In the IOA Online monitor environments (CICS, IMS, ROSCOE, VTAM, and so on) the CTOSE01 module is executed under the address space and TCB of the Online monitor.

Basic Definition Mode

A security check verifies if the user is authorized to use the user ID (owner) in the rule definition.

RACF Security

For this verification:

Entity Checked: owner.SUBMIT

Class: SURROGAT

where owner is the user ID specified as the owner of the Control-O rule.

A user who is authorized to submit a job on behalf of another user is also authorized to order Control-O rules owned by that user.

TopSecret Security

The TopSecret Application Interface module (TSSAI) is called with the following parameters:

Resource Class: ACIDCHK

Resource Name: owner

where owner is the user ID specified as the owner of the Control-O rule.

A user who is authorized to submit a job on behalf of another user is also authorized to order Control-O rules owned by that user.

The following TopSecret command permits USERA to order a rule with ownerid set to USERB:

TSS PERMIT(USERA) ACID(USERB)

ACF2 / SAF Security

For this verification:

Entity Checked: $SUBMIT.owner

Class: FACILITY

where owner is the user ID specified as the owner of the Control-O rule.

Extended Definition Mode

A security check verifies if the user is authorized to specify the user ID (owner) in the rule definition. The class checked is always FACILITY.

RACF Security

The entity checked for this verification is:

$$CTOORD.qname.owner

where owner is the user ID specified as the owner of the Control-O rule.

TopSecret Security

The entity checked for this verification is:

$$CTOORD.qname.owner

where owner is the user ID specified as the owner of the Control-O rule.

Use the following command to permit USERA to order Control-O rules owned by USERB:

TSS PERMIT(USERA) IBMFAC($$CTOORD.qname.USERB) ACC(READ)

ACF2 / SAF Security

The entity checked for this verification is:

$$CTOORD.qname.owner

where owner is the user ID specified as the owner of the Control-O rule.

Use the following command to permit USERA to order Control-O rules owned by USERB:

```
SET RESOURCE(CMF)
COMP
$KEY($$CTOORD.qname.owner) TYPE(CMF)
 UID(USERA) ALLOW
```

Module CTOSE02

The CTOSE02 module is the security module of Control-O Exit CTOX002. It is used to verify that the owner of a Control-O rule is allowed to specify the DO statements or ON statements specified in the rule definition. The module builds a list of security calls, one call for each DO statement and one call each for certain ON statements. For the rule to be loaded, the owner of the rule must have the authority to request all of the statements specified in the rule definition. If authorization fails for one of the calls, the entire rule load is canceled.

IOA performs a security check in which the CLASS checked is always FACILITY. The entity checked for each DO statement depends on if Basic Definition mode or Extended Definition mode is used.

Basic Definition Mode

The structure of the entity is as follows:

Table 61 CTOSE02 Basic Definition Entity Structure

Statement	Entity
DO COMMAND or ON COMMAND	$$IOACMD.qname.command-text This is the same structure that the IOASE12 IOA security module builds to verify authority for operator commands. If the current user is allowed to issue the operator command under the IOA operator command utility, that user is also allowed to define a rule containing this command.
DO DISPLAY	$$CTOMSG.qname.new-msg-text
ON MESSAGE	$$CTOMSG.qname.msgid
ON EVENT	$$CTOENV.qname.event-text
DO CTOPCMSG or ON CTOPCMSG	$$CTOPCM.qname.msg-text
ON JOBARRIV	$$CTOJAR.qname.jobname
ON JOBEND	$$CTOJED.qname.jobname
ON DSNEVENT	$$CTODSN.qname.jobname
ON RULE	$$CTORUL.qname.owner.rulname (owner is the owner of the rule)
ON STEP	$$CTOSTP.qname.jobname
DO COND or DO RESOURCE	$$IOARES.qname.resource-name This is the same structure that the IOASE07 IOA security module builds to verify the user's authorization to access prerequisite conditions and resources. If a user is allowed to access a condition or resource, that user can also access the condition or resource through a Control-O rule execution.

Statement	Entity
DO FORCEJOB	$$CTOCMO.qname.lib-name.table where - lib-name is the first 21 characters of the Control-M schedule library. - table is the member name in the Control-M schedule library. The whole entity name is truncated by RACF to 39. This means that table will be entirely truncated unless lib-name is less than 21.
DO TSO	$$CTOTSO.qname.command-text
DO SET for an IOA AutoEdit variable	$$CTOSET.qname.variable-name
DO DOM (delete operator message)	$$CTODOM.qname
DO ASKOPER if a WTOR is issued	$$CTOASK.qname.wtor-text
DO KSL	$$CTOKSL.qname.ksl-name
DO RULE	$$CTORUL.qname.owner.rulname where owner is the value of parameter OWNER in statement DO RULE.
DO STOPJOB	$$CTOJST.qname
DO SYSREQ	$$CTOSRQ.qname.sysreq-type where sysreq-type is the SYSREQ option. Valid value: ENQINFO
Runtime security setting	$$CTORTS.qname.runtime-sec. Valid values: - TRIGGER - OWNER - NONE as specified in rule parameter RUNTSEC.

In the above entities, command-text or msg-text represents the first 21 characters of the command text or message text. Note the following points regarding text of commands and messages within these entities:

- Multiple non-alphanumeric characters are replaced by one period.
- The period at the end of the text is dropped.
- All non-alphanumeric characters (blanks, commas, and so on) are replaced by periods.

For more details and examples, see Chapter 1, "IOA Security," including Module IOASE12 (on page 42).

Extended Definition Mode

The structure of the entity is as follows:

Table 62 CTOSE02 Extended Definition Entity Structure

Statement	Entity
ON COMMAND	$$CTOONC.qname.command-text
ON MESSAGE	$$CTOONM.qname.msg-text
ON EVENT	$$CTOENV.qname.event-text
ON CTOPCMSG	$$CTOONP.qname.msg-text
ON JOBARRIV	$$CTOJAR.qname.jobname
ON JOBEND	$$CTOJED.qname.jobname
ON DSNEVENT	$$CTODSN.qname.jobname
ON RULE	$$CTOORL.qname.owner.rulname where owner is the owner of the rule
DO COMMAND	$$CTOCMD.qname.command-text
DO COND or DO RESOURCE	$$CTORES.qname.resource-name

Statement	Entity
DO FORCEJOB	$$CTOCMO.qname.lib-name.table where ▪ lib-name is the first 21 characters of the Control-M schedule library. ▪ table is the member name in the Control-M schedule library The whole entity name is truncated by RACF to 39. This means that table will be entirely truncated unless lib-name is less than 21.
ON STEP	$$CTOSTP.qname.jobname
DO TSO	$$CTOTSO.qname.command-text
DO DISPLAY with SUPPRESS set to NO	$$CTODSP.qname.new-msg-text
DO DISPLAY with SUPPRESS set to YES	$$CTOSUP.qname
DO SET for an IOA AutoEdit variable	$$CTOSET.qname.variable-name
DO DOM (delete operator message)	$$CTODOM.qname
DO ASKOPER before a WTOR is issued	$$CTOASK.qname.wtor-text
DO CTOPCMSG	$$CTOPCM.qname.msg-text
DO KSL	$$CTOKSL.qname.ksl-name
DO RULE	$$CTODRL.qname.ownr.rulname where ownr is the value of the DO RULE owner parameter. Note: If the OWNER parameter in the DO RULE statement is empty, then the "owner" in the $$CTODRL.qname.ownr.rulname entity will be empty and the entity name will only consist of: $$CTODRL.qname.rulname.
DO STOPJOB	$$CTOJST.qname

Statement	Entity
DO SYSREQ	$$CTOSRQ.qname.sysreq-type where sysreq-type is the SYSREQ option. Valid value: ENQINFO
Runtime security setting	$$CTORTS.qname.runtime-sec Valid values: ▪ TRIGGER ▪ OWNER ▪ NONE as specified in rule parameter RUNTSEC.

In the above entities, command-text or msg-text represents the first 21 characters of the command text or message text. Note the following regarding text of commands and messages within these entities:

- Multiple non-alphanumeric characters are replaced by one period.
- The period at the end of the text is dropped.
- All non-alphanumeric characters (blanks, commas, and so on) are replaced by periods.

For more details and examples regarding this issue, see Module IOASE12 (on page 42).

Module CTOSE03

The CTOSE03 module is the security module of Control-O Exit CTOX003. This module verifies that the user ID associated with the rule is authorized to execute a DO TSO or DO KSL statement before the statement is executed. Depending on the value of the rule's RUNTSEC parameter, this user ID is either the owner of the rule, or the user ID of the user who issued the message or issued a command that triggered the rule.

Basic Definition Mode

Two security checks are performed for different entities:

- The first check verifies that the user is authorized to use the specific environment procedure for execution of a DO TSO or DO KSL statement. The environment is determined by subparameter INITPROC of the DO TSO/KSL statement. IOA issues a security check to verify authorization in which the CLASS checked is FACILITY and the entity checked is:

  $$CTOPRC.qname.env-prc

- The second check verifies that the user is authorized to execute the specific DO TSO or DO KSL statement. IOA issues a security check to verify authorization in which the CLASS checked is FACILITY and the entities checked are:

  ```
  DO TSO
  $$CTOTSO.qname.command-text
  ```

```
DO KSL
$$CTOKSL.qname.command-text
```

In the above entities, command-text represents the first 21 characters of the command text. Note the following regarding text of commands within these entities:

- Multiple non-alphanumeric characters are replaced by one period.
- The period at the end of the text is dropped.
- All non-alphanumeric characters (blanks, commas, and so on) are replaced by periods.

For more details and examples, see the description of the IOASE12 IOA security module in Chapter 1, "IOA Security."

Extended Definition Mode

Under this mode of operation, the security module verifies the user's authorization to execute a specific DO TSO or DO KSL statement within a specific environment procedure. The environment is determined by subparameter INITPROC of the DO TSO/KSL statement. IOA issues a security check to verify authorization in which the CLASS checked is FACILITY and the entities checked are:

```
DO TSO: $$CTOPTS.qname.env-prc.command-text
DO KSL: $$CTOPKS.qname.env-prc.command-text
```

In the above entities, command-text represents the first 14 characters of the command text. Note the following regarding text of commands within these entities:

- Multiple non-alphanumeric characters are replaced by one period.
- The period at the end of the text is dropped.
- All non-alphanumeric characters (blanks, commas, and so on) are replaced by periods.

For more details and examples, see the description of the IOASE12 IOA security module in 1 IOA Security.

Module CTOSE04

The CTOSE04 Control-O security module is used to verify the user's authority to access Automation Options screens and perform actions on specific Automation Option entities and on specific Control-O/COSMOS Online entities.

The Automation Options screen handles various aspects of the automation environment. Valid Automation Options: COMMAND, CONSOLE, ENQINFO, GLOBALS, OPERATOR, SAMPLES, SERVERS and SUBSYS. Additional Automation Options can be defined at your site (by the system administrator).

For a detailed description of each Automation Option, see the Online facilities chapter of the *Control-O User Guide*.

Basic Definition Mode

Initial Access to the Automation Options Screen (OA)

IOA security performs a security check to verify authorization for the option. The CLASS checked is FACILITY. The entity checked is:

```
$$CTOAOP.qname.option-name.ENTRY
```

Subsequent Operations in Screen OA

For every action that is performed on this screen, security is checked to verify authorization for the action. The CLASS checked is FACILITY. The entity checked is:

```
$$CTOAOP.qname.option.object
```

where

- option is the name of the Automation Option selected under screen OA.
- object is the text (1– 8 characters) identifying the object on which the action is performed.

Extended Definition Mode

Initial Access to the Entry Panel for the Automation Options Screen

A security check verifies authorization for the option in which the CLASS checked is FACILITY and the entity checked is:

```
$$CTOAOP.qname.option name.ENTRY
```

Subsequent Operations in Screen OA

For every action that is performed on this screen, security verifies authorization for the action in which the CLASS checked is FACILITY and the entity checked is:

```
$$CTOAOP.qname.option.object.action
```

where

- option is the name of the Automation Option selected under Screen OA.
- object is the text (1 through 8 characters) identifying the object on which the action was performed. When the CTOSE04 module cannot determine the object on which the action was performed, the word ACTION is used for the object.
- action is the action (one character) entered in the Automation Options screen.

For a description of the actions supported for each option, see the online facilities chapter in the *Control-O User Guide*.

Module CTOSE08

The CTOSE08 Control-O security module verifies the user’s authority to perform actions (hold, delete, and so on) on rules displayed in the Rule Status screen (Screen OS).

Functions performed by this security module depend on if Basic Definition mode or Extended Definition mode is used.

Basic Definition Mode

Initial Entry to the Rule Status Screen (Screen OS)

For every action that is performed on this screen, security verifies authorization for the action in which the CLASS checked is FACILITY and the entity checked is:

```
$$CTOPNLOS.qname
```

Subsequent Operations in the Rule Status Screen (Screen OS)

For all actions (hold, free, delete, and so on) that are performed on this screen, security verifies authorization.

The check verifies that a user who submits jobs with parameter USER has authority equal to that of the rule's owner. A user who is authorized to submit a job on behalf of others is also authorized to perform specific actions (hold, delete, and so on) on rules belonging to other users. If the user is the job's owner, the security check is bypassed.

RACF Security

The CLASS checked is SURROGAT. The entity checked is:

```
owner.SUBMIT
```

where owner is the user ID assigned to the accessed rule.

TopSecret Security

TopSecret Application Interface module (TSSAI) is called with the following parameters:

Class Name: ACIDCHK

Resource Name: ownerid

where ownerid is the user ID assigned to the accessed rule.

ACF2/SAF Security

For all actions (hold, free, delete, and so on) that are performed on this screen, IOA security verifies authorization. The CLASS checked is FACILITY. The entity checked is $SUBMIT.owner

where owner is the user ID assigned to the accessed rule.

Extended Definition Mode

Initial Access to the Rule Status Screen (Screen OS)

For every action that is performed on this screen, security verifies authorization in which the CLASS checked is FACILITY and the entity checked is:

```
$$CTOPNLOS.qname
```

Subsequent Operations to the Rule Status Screen (Screen OS)

The actions (hold, free, delete, and so on) are separated into different categories of access authority. The CLASS checked is FACILITY, and the entity checked is:

```
$$RULxrrr.qname.owner
```

where

- owner is the owner specified in the rule definition.
- x is the one digit action identifier.
- rrr is the three character identifier for each action.

Valid actions and action identifiers are listed in the table below.

Table 63 Action Identifiers

Action Identifier	Action	Description
2	HLD FRE MOD RES	Hold Free Mode Resume
3	DEL CAN	Delete Cancel

The CTOSE08 module can be used to check for authorization to display individual lines on the Rule Status screen. Since a line-by-line authorization check affects performance, Control-O invokes the CTOSE08 module when a user enters the Rule Status screen, but does not perform security checks. Users who want to limit the lines displayed on the Rule Status screen can use the Control-O call to the CTOSE08 module to apply security checks at this stage.

To permit USERA to hold rules owned by USERB, use the following command:

For RACF:

```
PERMIT $$RUL2HLD.qname.USERB ACCESS(READ) ID(USERA) CLASS(FACILITY)
```

For TopSecret:

```
TSS PERMIT(USERA) IBMFAC($$RUL2HLD.qname.USERB) ACC(READ)
```

For ACF2/SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($$RUL2HLD.qname.USERB) TYPE(CMF)
 UID(USERA) ALLOW
```

Module CTOSE10

The CTOSE10 Control-O security module supports protection of Control-O rule triggering. CTOSE10 is invoked before triggering a rule to check if the user who issued the triggering event (message, command, and so on) is authorized to trigger the rule.

There is no security exit CTOX010 and there is no distinction between basic and extended security mode.

CTOSE10 will be invoked if the feature is enabled (variable DFMO10 set to PROD or TEST, as described in Class name customization (on page 155)) and the rule includes the following statement anywhere in the rule:

```
DO SET =  %%PROTRULE=Y
```

The entity name that represents the rule and which is protected by SAF (for example, RACF) is constructed as follows:

```
$$CTOSRL.ioaqname.rule-name.type.table.dsn
```

where

- ioaqname is the IOA QNAME of the installation
- rule-name is the name of the rule which is taken from the first ON statement in the rule
- type is the rule type and can be of the following:

  `COMMAND`

  `DSNEVENT`

  `MESSAGE`

  `CTOPCMSG`

  `JOBARRIV`

  `STRING`

  `JOBEND`

  `STEP`

  `OMEGAEXP`

  `SYSOUT`

  `RULE`

  `MVALARM`

  `SMS`
- table is the rules table (member) name
- dsn is the rules library name

The constructed entity name is not necessarily unique, since it is possible to define multiple rules in the same table with the same first ON statement

Class name customization

The entity name may exceed 39 characters, and therefore a class that supports a higher limit should be used instead of the FACLITY class. The XFACILIT class is used as the default class. The class name can be customized by setting parameter IOAXCLASS as follows:

a. Enter ICE and select Customization.

b. In the Customization window, set the Product to IOA and select Security Customization.

c. Press Enter to display the Major Steps Selection screen.

d. Select major step 1, "Implement IOA Security."

e. Select minor step 2, "Customize Security Parameters."

➢ To activate rule protection

a. If not yet applied, apply IBM APAR OA10774, which supports the XFACILIT class on z/OS systems earlier than V1R7.

b. Add the following to each rule that should be protected:

```
DO SET= %%PROTRULE=YGLOBAL  N
```

c. Grant READ authorization to the appropriate entities who represent the protected rules, to the users that are allowed to trigger them.

d. Customize the DFMO10 variable as follows:

Enter ICE and select Customization.

In the Customization window, set the Product to CTO and select Security Customization.

Press Enter to display the Major Steps Selection screen.

Select major step 1, "Implement Control-O Security."

Select minor step 2, "Customize Security Parameters."

The following values can be specified for DFMO10:

- NO – disables the feature (default)
- TEST – in TEST mode, the authorization is checked and RACF error messages are issued if the user is not authorized, but the rule is still invoked
- PROD – enables the feature

e. Stop and then restart the Control-O or CTMCMEM monitor, or start a new monitor and issue the following command:

```
F <monitor>,RELOAD=CTOWTO
```

Module CTOSE15

The CTOSE15 Control-O security module verifies the user’s authority to request services (INIT, RESOLVE, and so on) from the XAM (Extended Automation Mechanism) and CTOSCMD interface. For information about the XAM interface, see the Extended Automation Mechanism chapter in the *Control-O User Guide*.

The function performed by this security module depends on if Basic Definition mode or Extended Definition mode is used.

Basic Definition Mode

For all actions (INIT, RESOLVE, and so on) performed by the XAM interface, IOA performs a security check for authorization. The class checked is FACILITY. The entity checked is:

```
$$CTOXAMF.qname
```

Extended Definition Mode

Actions (INIT, RESOLVE, and so on) are separated into different categories of access authority. The CLASS checked is FACILITY. The entity checked is:

$$CTOXAM.qname.TYPExrrr

where

- x is the one-digit action identifier.
- rrr is the three-character identifier for each action.

Valid actions and action identifiers are listed in the table below.

Table 64 Action Identifiers

Action Identifier	Action	Description
1	INI TRM RSL	INIT TERM RESOLVE
2	LOC	SETOLOC
3	GLB RUL	SETOGLB DORULE

To protect the Control-O CTOSCMD function, the CTOSCMD rule should be protected, in addition to INIT and TERM.

Optionally including the target rule name in the checked entity

By default the security validation of the Control-O XAM DORULE action does not restrict users from invoking only specific target rules. If such restriction is required, use the option to include the target rule name in the checked entity (in extended definition mode). This option is controlled by parameter ADDRULNM in the SECPARM parameter member.

When ADDRULNM=N (default) the checked entity is:

$$CTOXAM.qname.TYPE3RUL (default).

When ADDRULNM=Y the checked entity is:

$$CTOXAM.qname.TYPE3RUL.rulename (where rulename is the target rule name of the DORULE action).

To set the option for including the target rule name in the checked entity

1. Invoke ICE.
2. On the ICE Main screen, select "Customization".
3. Select Product "CTO", "Security Customization".
4. Select step 1.2 "Customize Security Parameters".
5. Set parameter ADDRULNM to either Y or N.
6. Recreate SECPARM with a new parameter by selecting the step 1.3 "Save Security Parameters into Product".

APAR BO10168 is needed for this option.

To permit USERA to set a local variable using the XAM interface, use the appropriate command.

For RACF:

```
PERMIT $$CTOXAM.qname.TYPE2* ACCESS(READ) ID(USERA) CLASS(FACILITY)
```

For TopSecret:

```
TSS PERMIT(USERA) IBMFAC($$CTOXAM.qname.TYPE2) ACCESS(READ)
```

For ACF2/SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($$CTOXAM.qname.TYPE2*************) TYPE(CMF)
 UID(USERA) ALLOW
```

Control-M/Analyzer Security

This chapter describes the procedure used to implement the Control-M/Analyzer security interface. It is recommended that you first review the explanations below on the elements that are protected in Control-M/Analyzer and then proceed to the step-by-step instructions.

Protecting Control-M/Analyzer Elements

The Control-M/Analyzer security interface protects the following Control-M/Analyzer elements:

- Ordering balancing missions.
- Access to the Control-M/Analyzer files.
- Access to the Rule Activity screen.
- Access to the Control-M/Analyzer Active Balancing Environment screen and the invocation of line commands under the Active Balancing Environment screen (hold, free, delete, and so on).

Balancing Missions

Each Control-M/Analyzer balancing mission contains an OWNER parameter. This parameter is the user ID to which this mission belongs. If a user orders a balancing mission, the user must be authorized to access the owner of the rule. The CTBSE01 Control-M/Analyzer security module verifies that the logged on user is authorized to order a balancing mission that belongs to the owner of the rule.

Access to IOA Files

IOA files contain Control-M/Analyzer information for groups, variables, and variable generations. These files are accessed through the online screens, utilities and runtime environment.

The CTBSE03 Control-M/Analyzer security module verifies the user authority each time IOA Access Method files are accessed.

Access to the Rule Activity Screen

The Rule Activity screen in Control-M/Analyzer lists Control-M/Analyzer rule invocations and their status. The user can view information about a rule in the list, print a rule, or perform a rollback of the rule invocation, and so on.

The CTBSE04 Control-M/Analyzer security module verifies the user’s authority to perform various actions on the rules displayed in the Rule Activity screen.

Access to the Active Balancing Environment Screen

The Control-M/Analyzer Active Balancing Environment screen lists the balancing missions currently being handled by Control-M/Analyzer and their status. The user can issue inquiries about a rule within the list, or change its status.

The CTBSE08 Control-M/Analyzer security module verifies the user's authorization to perform actions (hold, delete, and so on) on balancing missions displayed in the Active Balancing Environment screen.

Control-M/Analyzer Basic Definition Security Calls

Table 65 Control-M/Analyzer Basic Definition Security Calls

Protected Element	Type	Class Entity Name	Explanation	Security Module
Controlling Balancing Mission Ordering		SURROGAT owner.SUBMIT ACIDCHK owner FACILITY $SUBMIT.owner	owner is the name of the user specified in the balancing mission definition.	CTBSE01
Controlling Access to the Control-M/Analyzer Database		FACILITY $$CTBDBA.qname.groupname	groupname contains the first characters of the requested group name (maximum: 12 characters).	CTBSE03
Controlling Access to IOA Access Method Files		FACILITY $$CTBDBA.qname.groupname.varname	groupname contains the first characters of the requested group name (maximum: 12 characters). varname contains the first characters of the requested Database variable name (maximum: 12 characters). varname is optional and can be used only when the object of the request is a Database variable or Database variable generation. If varname is omitted, the full name of the requested group can be used in the corresponding entity.	CTBSE03

Protected Element	Type	Class Entity Name	Explanation	Security Module
Control-M/Analyzer Rule Activity screen		FACILITY $$CTBACT.qname.groupname.jobname	jobname contains the requested job name.	
Controlling Access to the Active Balancing Environment Screen				
Authority to enter Active Balancing Environment screen		FACILITY $$CTBPNLB.qname		CTBSE08
Use of Active Balancing Environment screen		SURROGAT owner.SUBMIT ACIDCHK ownerid FACILITY $SUBMIT.owner	owner is the name if the user specified in the balancing mission definition.	CTBSE08

Control-M/Analyzer Extended Definition Security Calls

Table 66 Control-M/Analyzer Extended Definition Security Calls

Protected Element	Type	Class Entity Name	Explanation	Security Module
Controlling Balancing Mission Ordering		FACILITY $$BALORD.qname.owner	owner is the name of the user specified in the balancing mission order definition.	CTBSE01
Controlling Access to the Control-M/Analyzer Database		FACILITY CREATE a group: $$GRPCRE.qname.groupname UPDATE a group: $$GRPUPD.qname.groupname DELETE a group: $$GRPDEL.qname.groupname VIEW a group: $$GRPVEW.qname.groupname VIEW Database variable in the group: $$GRPVWV.qname.groupname USE a group: $$GRPUSE.qname.groupname	groupname contains the first characters of the requested group name (maximum: 12 characters).	CTBSE03

Protected Element	Type	Class Entity Name	Explanation	Security Module
Controlling Access to IOA Access Method Files		FACILITY CREATE a group's Database variable: $$VRBCRE.qname.groupname. varname UPDATE a group's Database variable: $$VRBUPD.qname.groupname. varname DELETE a group's Database variable: $$VRBDEL.qname.groupname. varname VIEW a group's Database variable: $$VRBVEW.qname.groupname. varname VIEW a Database variable's generation: $$VRBVWG.qname.groupname.v arname CREATE a Database variable's generation: $$VRGCRE.qname.groupname. varname GET a Database variable's generation: $$VRGVEW.qname.groupname. varname UPDATE a Database variable's generation: $$VRGUPD.qname.groupname. varname DELETE a Database variable's generation: $$VRGDEL.qname.groupname. varname ROLL a Database variable's generation: $$VRGROL.qname.groupname. varname	groupname contains the first characters of the requested group name (a maximum of 12 characters). varname contains the first characters of the requested Database variable name (a maximum of 12 characters). varname is optional and can be used only when the object of the request is a Database variable or Database variable generation. If varname is omitted, the full name of the requested group can be used in the corresponding entity.	CTBSE03

Protected Element	Type	Class Entity Name	Explanation	Security Module
Control-M/Analyzer Rule Activity screen		FACILITY VIEW of Invocation Database variables: $$VIEW.qname.groupname.jobname VIEW of Invocation Log: $$LOG.qname.groupname.jobname VIEW of Invocation Report: $$REPORT.qname.groupname.jobname PRINT of Invocation Report: $$PRINT.qname.groupname.jobname ROLL–BACK of Invocation: $$ROLL.qname.groupname.jobname	jobname contains the requested job name	
Controlling Access to the Active Balancing Environment Screen				
Authority to enter the Active Balancing Environment screen		FACILITY $$CTBPNLB.qname		CTBSE08
Use of Active Balancing Environment screen		FACILITY Hold: $$BRULHLD.qname.owner Free: $$BRULFRE.qname.owner Delete: $$BRULDEL.qname.owner Log: $$BRULLOG.qname.owner Why: $$BRULWHY.qname.owner Zoom: $$BRULZOO.qname.owner Save: $$BRULSAV.qname.owner	owner is the name if the user specified in the balancing mission definition.	CTBSE08

Implementing Control-M/Analyzer Security

This section details the steps required to implement the Control-M/Analyzer security interface.

The Control-M/Analyzer security interface can be installed either as part of the customized installation path, or during the Customization process after installation. Both options use the INCONTROL Installation and Customization Engine (ICE) application. If you are not familiar with the ICE interface, see the *INCONTROL for z/OS Installation Guide: Installing*.

The Control-M/Analyzer security interface cannot be implemented until IOA security is installed. Verify that IOA security is installed before proceeding with Control-M/Analyzer security implementation.

➢ To install the Control-M/Analyzer security interface

a. Enter the main ICE screen.

b. Select Customization.

c. Enter CTB in the Product field.

d. Select Security Customization.

e. Perform all major and minor steps required to install the security product.

Step 1 Implement Control-M/Analyzer Security

Follow the steps below to implement Control-M/Analyzer security.

Step 1.1 Grant Access Permissions

Collect the data you need to define the INCONTROL entities and user authorizations to the security product.

In ICE, run the steps "Control-M/Analyzer Security Definitions (Sample)" and "Functions Security Definitions (Sample)" to create two sample jobs.

Step 1.2 Customize Security Parameters

Table 67 Control-M/Analyzer Modes

Mode	Definition
DEFMCHKB	When choosing a definition mode as COND to any of the Control-M/Analyzer security modules, use qname together with the value given to this parameter as the high level qualifier, to determine the real definition mode to be used.
SECTOLB	Determine the action to perform if your security product is inactive or a specific resource is not defined to the security product.Valid values are: ▪ YES — Perform the action. ▪ NO — Do not perform the action.
Mode Definition	Specify one of the following values to determine the Definition mode for the Control-M/Analyzer security modules: ▪ COND-Conditional Definition mode. Default. ▪ BASIC-Basic Definition mode. ▪ EXTEND-Extended Definition mode.
DFMB01	Definition mode for the CTBSE01 Control-M/Analyzer security module.
DFMB03	Definition mode for the CTBSE03 Control-M/Analyzer security module.
DFMB04	Definition mode for the CTBSE04 Control-M/Analyzer security module.
DFMB08	Definition mode for the CTBSE08 Control-M/Analyzer security module.

Step 1.3 Save Security Parameters into Product

This step saves all the security parameters specified for Control-M/Analyzer.

Step 2 RACF Security Definition Samples

Step 2.1 Control-M/Analyzer Security Definitions (Optional)

Step 2.2 Functions Security Definitions (Optional)

Step 2.3 Control Program Access to Datasets (Optional)

Select these steps to edit members CTBSRAC2, CTBSRAC3, or CTBSRAC4 in the IOA INSTWORK library.

Perform the following steps to define the required permissions.

1. Associate users with Extended Definition Mode.

   a. To define the entity $$CTBEDM.qname to RACF, use the following command:

      RDEFINE FACILITY $$CTBEDM.qname UACC(NONE)

   b. To authorize USERA to Extended Definition mode, use the following command:

      PERMIT $$CTBEDM.qname ID(USERA) CLASS(FACILITY) ACCESS(READ)

   c. Submit the job for execution.

      This job must run under a user or administrator who has authorization to enter these commands.

      Scan the output of the job for information and error messages. All job steps must end with a condition code of 0.

2. Define entities and user authorizations.

   For information about entities and user authorizations, see Control-M/Analyzer Basic Definition Security Calls (on page 160), and Control-M/Analyzer Extended Definition Security Calls (on page 162).

To authorize USERA access to a given Control-M/Analyzer entity, use the following command:

```
PERMIT $$CTBnnn.qname CLASS(FACILITY) ID(USERA) ACCESS(READ)
```

where CTBnnn is the name of the Control-M/Analyzer entity to be accessed.

All entity names for each Control-M/Analyzer protected element appear in Control-M/Analyzer Basic Definition Security Calls (on page 160)for Basic Definition mode and Control-M/Analyzer Extended Definition Security Calls (on page 162) for Extended definition mode.

Step 3 TopSecret Security Definition Samples

Step 3.1 ControlM/Analyzer Security Definitions (Optional)

Step 3.2 Functions Security Definitions (Optional)

Step 3.3 Control Program Access to Datasets (Optional)

Select these steps to edit members CTBSTSS2, CTBSTSS3, or CTBSTSS4 in the IOA INSTWORK library.

1. Define Entities and User Authorizations to TopSecret.

   For information about how to define Control-M/Analyzer entities and user authorizations to TopSecret, see Control-M/Analyzer Basic Definition Security Calls (on page 160), and Control-M/Analyzer Extended Definition Security Calls (on page 162).

   a. Add the following command to add the resources to TopSecret:

      TSS ADD(sec-administrator-dept) IBMFAC($$CTB)

      Set the sec-administrator-dept parameter to the appropriate value.

      All entity names for each Control-M/Analyzer protected element appear in Control-M/Analyzer Basic Definition Security Calls (on page 160) for Basic Definition mode and Control-M/Analyzer Extended Definition Security Calls (on page 162)for Extended Definition mode.

2. Associate users with Extended Definition Modes.

   Authorizations to access Control-M/Analyzer datasets are defined during the Control-M/Analyzer installation process. This step must be completed before proceeding with security implementation. For information about how to grant users access to Control-M/Analyzer datasets, see the Control-M/Analyzer chapter in the *INCONTROL for z/OS Installation Guide: Installing*.

   a. Add the following TopSecret command to define the $$CTBEDM.qname entity to TopSecret and authorize users to this entity:

      ```
      TSS PERMIT(USERA) IBMFAC($$CTBEDM.qname) ACC(READ)
      ```

Do not define the $$CTBEDM.qname entity to operate in warning mode because this causes all users to operate in Extended Definition mode.

3. Authorize Control-M/Analyzer installer to all Control-M/Analyzer facilities.

   a. Customize the following command to authorize USERA to Control-M/Analyzer facilities:

      ```
      TSS PERMIT(USERA) IBMFAC($$CTB) ACC(READ)
      ```

      Set the USERA parameter to the user ID of the Control-M/Analyzer installer.

   b. Submit Job CTBSTSS2

      This job must be run under the ACID of the general security administrator (SCA) who is authorized to enter these TopSecret commands.

      All job steps must end with a condition code of 0.

Step 4 ACF2 Security Definition Samples

Step 4.1 ControlM/Analyzer Security Definitions (Optional)

Step 4.2 Functions Security Definitions (Optional)

Step 4.3 Control Program Access to Datasets (Optional)

Select these steps to edit members CTBSSAF2, CTBSSAF3, or CTBSSAF4 in the IOA INSTWORK library.

Perform the following steps to define the required permissions.

1. Associating users With Extended Definition Mode.

   a. Add the following ACF2 commands to define the $$CTBEDM.qname entity to ACF2, and authorize users to this entity.

   b. Define and authorize the entity: $$CTBEDM.qname to ACF2 using the following command:

      ```
      SET RESOURCE(CMF)
      COMP
      $KEY($$CTBEDM.qname) TYPE(CMF)
       UID(USERA) ALLOW
      ```

2. Define entities and user authorizations to CA-ACF2/SAF.

   For information about entities and user authorizations, see Control-M/Analyzer Basic Definition Security Calls (on page 160), and Control-M/Analyzer Extended Definition Security Calls (on page 162).

Example

To authorize USERA (the user ID of the Control-M/Analyzer installer) access to a given Control-M/Analyzer entity, use the following command:

```
SET RESOURCE(CMF)
COMP
$KEY($$CTBnnn.qname) TYPE(CMF)
 UID(USERA) ALLOW
```

where qname is the name used to assign different authorizations to different Control-M/Analyzer environments (such as Test and Production). This parameter is specified during IOA installation.

Set the USERA parameter to the UID string of the Control-M/Analyzer installer.

All entity names for each Control-M/Analyzer protected element appear in Control-M/Analyzer Basic Definition Security Calls (on page 160)for Basic Definition mode and Control-M/Analyzer Extended Definition Security Calls (on page 162)for Extended Definition mode.

For samples of user authorizations, review member CTBSSAF3 in the IOA INSTWORK library.

3. Submit the Job

This job must be run under a user of a ACF2/SAF administrator who has authorization to enter these ACF2 commands.

Scan the output of the job for information and error messages produced by ACF2/SAF. All job steps must end with a condition code of 0.

Control-M/Analyzer Security Interface Modules

This section describes the Control-M/Analyzer Security Interface Modules.

Module CTBSE01

The CTBSE01 module is the security module of Control-M/Analyzer Exit CTBX001. It is used to verify that the user is authorized to order balancing missions. A security check is issued to verify that the logged on user is allowed to order balancing missions on behalf of the user ID as specified in the OWNER field of the mission definition. The CTBSE01 module executes under the address space of the logged on TSO/ROSCOE user or under the TCB related to the logged on user when working in cross memory mode under the Online monitor. The class checked is FACILITY unless otherwise specified.

Basic Definition Mode

IOA verifies if the user is authorized to use the user ID (owner) in the balancing mission definition.

RACF Security

For this verification:

Entity Checked: owner.SUBMIT

Class: SURROGAT

where owner is the user ID specified as the owner of the Control-M/Analyzer balancing mission.

If the logged on user is allowed to submit jobs on behalf of another user, the user is also allowed to order Control-M/Analyzer balancing missions owned by that user.

The commands to permit USERA to order a balancing mission with an owner of USERB are:

```
RDEFINE SURROGAT USERB.SUBMIT UACC(NONE)
PERMIT USERB.SUBMIT ACCESS(READ) ID(USERA) CLASS(SURROGAT)
```

TopSecret Security

The TopSecret Application Interface module (TSSAI) is called with the following parameters:

Resource Class: ACIDCHK

Resource Name: userid (as specified in the OWNER field)

where userid is the user ID specified as the owner of the Control-M/Analyzer balancing mission.

If the logged on user is allowed to submit jobs on behalf of another user, it is assumed that the user is also allowed to order Control-M/Analyzer balancing missions owned by that user.

The command to permit USERA to order a balancing mission with an ownerid of USERB is:

TSS PERMIT(USERA) ACID(USERB)

ACF2 / SAF Security

For this verification:

Entity Checked: $SUBMIT.owner

Class: FACILITY

where owner is the user ID specified as the owner of the Control-M/Analyzer balancing mission.

The ACF2 commands to permit USERA to order a balancing mission with an owner of USERB are:

```
SET RESOURCE(CMF)
COMP
$KEY($SUBMIT.USERB) TYPE(CMF)
 UID (USERA) ALLOW
```

Extended Definition Mode

IOA verifies if the user is authorized to specify the user ID (owner) in the rule definition.

RACF Security

The entity checked for this verification is:

$$BALORD.qname.owner

where owner is the user ID specified as the owner of the Control-M/Analyzer rule or balancing mission. To permit USERA to order Control-M/Analyzer missions owned by USERB, use the following commands:

```
RDEFINE FACILITY $$BALORD.qname.USERB UACC(NONE)
PERMIT $$BALORD.qname.USERB ACCESS(READ) ID(USERA) CLASS(FACILITY)
```

TopSecret Security

The entity checked for this verification is:

$$BALORD.qname.owner

where owner is the user ID specified as the owner of the Control-M/Analyzer rule or balancing mission. To permit USERA to order Control-M/Analyzer missions owned by USERB, use the following commands:

```
TSS ADD(sec-administrator-dept) IBMFAC($$BALORD)
TSS PERMIT(USERA) IBMFAC($$BALORD.qname.USERB) ACC(READ)
```

ACF2 / SAF Security

The entity checked for this verification is:

$$BALORD.qname.owner

where owner is the user ID specified as the owner of the Control-M/Analyzer rule or balancing mission. To permit USERA to order Control-M/Analyzer missions owned by USERB, use the following ACF2 commands:

```
SET RESOURCE(CMF)
COMP
$KEY($$BALORD.qname.USERB)
 UID (USERA) ALLOW
```

Module CTBSE03

The CTBSE03 module is the security module of Control-M/Analyzer Exit CTBX003. This module verifies that the user is authorized to access groups, Database variables, and Database variable generations from the Control-M/Analyzer Database Facility, utilities and runtime environment.

IOA verifies authorization in which the CLASS checked is FACILITY and the entity checked depends on the definition mode.

Basic Definition Mode

The entity used to check authorization is:

$$CTBDBA.qname.groupname.varname

To permit USERA to use all Database variables in all groups beginning with SYS, use the following commands:

For RACF:

```
RDEFINE FACILITY $$CTBDBA.qname.SYS* UACC(NONE)
PERMIT $$CTBDBA.qname.SYS* CLASS(FACILITY) ID(USERA) ACCESS(READ)
```

For TopSecret:

```
TSS PERMIT(USERA) IBMFAC($$CTBDBA.qname.SYS) ACC(READ)
```

For ACF2 / SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($$CTBDBA.qname.SYS)
 UID (USERA) ALLOW
```

Extended Definition Mode

The entity used to check authorization depends on the user request:

$$xxxyyy.qname.groupname.varname

To permit USERA to operate with a group and a Database variable, use the following commands:

For RACF:

```
RDEFINE FACILITY $$xxxyyy.qname.groupname.varname UACC(NONE)
PERMIT $$xxxyyy.qname.groupname.varname CLASS(FACILITY) ID(USERA)
ACCESS(READ)
```

For TopSecret:

```
TSS PERMIT(USERA) IBMFAC($$xxxyyy.qname.groupname.varname) ACC(READ)
```

For ACF2/SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($$xxxyyy.qname.groupname.varname)
 UID (USERA) ALLOW
```

where

xxx is a three character string that defines the object of the request as follows:

Table 68 Object String Definition

String	Description
GRP	Group
VRB	Database variable
VRG	Database variable generation

yyy is a three character string that defines the request as follows:

Table 69 Request String Definition

String	Description
CRE	Create
CNF	Confirm
UPD	Update
DEL	Delete
ROL	Database variable rollback (with VRG only)
VEW	View
VWV	View Database variable in the group
VWG	View Database variable generation
USE	Use (with GRP only)

Table 70 CTBSE03 Parameters

Parameter	Description
groupname	First characters of the requested group name (a maximum of 12 characters).
varname	First characters of the requested Database variable name (a maximum of 12 characters).
	Used only when the object of the request is a Database variable or Database variable generation. If varname is omitted, the full name of the requested group can be used in the corresponding entity. Optional.

When an attempt is made to execute any of these commands, security module CTBSE03 is called to check if the command must be executed. In this case, this security module does not perform security checks for each line of the screen. For performance reasons, the check on each screen line is not performed.

Module CTBSE04

The CTBSE04 module is the security module of Control-M/Analyzer Exit CTBX004. This module verifies that the user is authorized to access groups, jobs and invocations from the Control-M/Analyzer Job Activity screen.

The CLASS checked is FACILITY. The entity used to check authorization depends on if Basic Definition mode or Extended Definition mode is used.

Basic Definition Mode

The entity used to check authorization is $$CTBACT.qname.groupname.jobname

For example, to permit USERA to use all jobs invocations in all groups beginning with SYS, use the following commands:

For RACF:

```
RDEFINE FACILITY $$CTBACT.qname.SYS* UACC(NONE)
PERMIT $$CTBACT.qname.SYS* CLASS(FACILITY) ID(USERA) ACCESS(READ)
```

For TopSecret:

```
TSS PERMIT(USERA) IBMFAC($$CTBACT.qname.SYS) ACC(READ)
```

For ACF2 / SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($CTBACT.qname.SYS***********************)
 UID (USERA) ALLOW
```

Extended Definition Mode

The entity used to check authorization depends on the user request:

$$xxxxxx.qname.groupname.jobname

where

xxxxxx contains a maximum of six letters that define the request:

Table 71 CTBSE04 Request Parameters

Request	Description
FRMCNF	Confirm use of display type
RECCNF	Confirm display of each invocation
VIEW	View Invocation Database variables
LOG	View Invocation log
REPORT	View Invocation report
PRINT	Print Invocation report
ROLL	Rollback of invocation

groupname contains the first letters (maximum: 16) of the requested group name.

jobname contains the requested job name.

For example, to permit USERA to view the log of the invocation for job M999XPRD in group PRODGROUP, use the following commands:

For RACF:

```
RDEFINE FACILITY $$LOG.qname.PRODGROUP.M999XPRD UACC(NONE)
PERMIT $$OLOG.qname.PRODGROUP.M999XPRD CLASS(FACILITY) ID(USERA)
ACCESS(READ)
```

For TopSecret:

```
TSS PERMIT(USERA) IBMFAC($$LOG.qname.PRODGROUP.M999XPRD) ACC(READ)
```

For ACF2/SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($$LOG.qname.PRODGROUP.M999XPRD)
 UID(USERA) ALLOW
```

When an attempt is made to execute any of these commands, the CTBSE04 security module is called to check if the command must be executed. In this case, this security module does not perform security checks for each line of the screen. For performance reasons, the check on each screen line is not performed.

Module CTBSE08

The CTBSE08 module is the security module of Control-M/Analyzer Exit CTBX008. This module verifies that the user is authorized to perform actions (hold, delete, and so on) on balancing missions displayed in the Active Balancing Environment screen.

Basic Definition Mode

Initial Access to the Active Balancing Environment Screen

IOA verifies authorization for the option in which the CLASS checked is FACILITY unless otherwise specified. The entity checked is $$CTBPNLB.qname

Subsequent Operations in the Active Balancing Environment Screen

For every action that is performed on this screen, IOA verifies authorization for the action.

RACF Security

For this verification:

Entity Checked: owner.SUBMIT

Class: SURROGAT

where owner is the user ID specified as the owner of the Control-M/Analyzer balancing mission.

If the logged on user (current user) is allowed to submit jobs on behalf of another user, the current user is also allowed to order Control-M/Analyzer balancing missions owned by the other user.

The commands to permit USERA to order a balancing mission with an owner of USERB are:

```
RDEFINE SURROGAT USERB.SUBMIT UACC(NONE)
PERMIT USERB.SUBMIT ACCESS(READ) ID(USERA) CLASS(SURROGAT)
```

TopSecret Security

The TopSecret Application Interface module (TSSAI) is called with the following parameters:

Resource Class: ACIDCHK

Resource Name: userid (as specified in the OWNER field)

where userid is the user ID specified as the owner of the Control-M/Analyzer balancing mission.

If the logged on user is allowed to submit jobs on behalf of another user, it is assumed that the logged on user is also allowed to order Control-M/Analyzer balancing missions owned by that user.

The command to permit USERA to order a balancing mission with an ownerid of USERB is:

TSS PERMIT(USERA) ACID(USERB)

ACF2/SAF Security

For this verification:

Entity Checked: $SUBMIT.owner

Class: FACILITY

where owner is the user ID specified as the owner of the Control-M/Analyzer balancing mission.

The ACF2 commands to permit USERA to order a balancing mission with an owner of USERB are:

```
SET RESOURCE(CMF)
COMP
$KEY($SUBMIT.USERB) TYPE(CMF)
 UID (USERA) ALLOW
```

Extended Definition Mode

Initial Access to the Active Balancing Environment Screen

IOA verifies authorization for the option in which the CLASS checked is FACILITY and the entity checked is $$CTBPNLB.qname

Subsequent Operations in the Active Balancing Environment Screen

For every action that is performed on this screen, IOA verifies authorization for the action. The entity checked is:

$$BRULxxx.qname.owner

where owner is the user ID assigned as the owner of the balancing mission, and xxx contains 3 letters that define the request:

Table 72 CTBSE08 Request Parameters

Action	Description
ZOO	Zoom
HLD	Hold
FRE	Free
DEL	Delete
LOG	Log
SAV	Save
WHY	Why

To permit USERA to hold balancing missions that are owned by USERB, use the following command:

For RACF:

```
PERMIT $$BRULHLD.qname.USERB ACCESS(READ) ID(USERA) CLASS(FACILITY)
```

For TopSecret:

```
TSS PERMIT(USERA) IBMFAC($$BRULHLD.qname.USERB) ACC(READ)
```

For ACF2/SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($$BRULHLD.qname.USERB)
 UID(USERA) ALLOW
```

When Control-M/Analyzer attempts to display a line on the Active Balancing Environment screen, it calls the CTBSE08 security module. This module can be modified to check if the line must be displayed or not. For performance reasons, the CTBSE08 sample security module performs no security checks.

Control-M/Tape Security

This chapter describes the procedure used to implement the Control-M/Tape security interface. It is recommended that you first review the explanations below on the elements that are protected in Control-M/Tape and then proceed to the step-by-step instructions.

Protecting Control-M/Tape Elements

The Control-M/Tape security interface protects the following Control-M/Tape elements:

- Invocation of the Control-M/Tape initialization.
- Use of JCL parameter BLP and setting of JCL parameter EXPDT to 98000.
- Update of Media Database from the online environment, the real-time environment, or Control-M/Tape utilities.
- Authority to dynamically define tape volumes and datasets in the real-time environment.
- Authority to create and print tape labels in batch and online environments.

Invocation of the Control-M/Tape Initialization Process

Whenever a user attempts to invoke the Control-M/Tape initialization process, the Control-M/Tape initialization procedure invokes the CTTSE01 Control-M/Tape security module to determine if the current user is allowed to activate Control-M/Tape initialization.

This module is also activated for each rule that is loaded to the real-time environment.

Use of JCL Parameter BLP and Setting of JCL Parameter EXPDT to 98000

Whenever a user attempts to use JCL parameter BLP or to set JCL parameter EXPDT to 98000, the Control-M/Tape SVC invokes the CTTSE03 Control-M/Tape security module to determine if the current user is allowed to use/set these parameters in this way.

Media Database Updates

Whenever a user attempts to update the Media Database from the online environment (Inquire, Update, Check in), from the real-time environment (using Control-M/Tape SVC), or from Control-M/Tape utilities (for example, CTTVTM, CTTRTM), the CTTSE06 Control-M/Tape security module is invoked to determine if the current user is allowed to perform the action.

Authority to Dynamically Define Tape Volumes and Datasets

Whenever a batch job requests that a tape volume or dataset be dynamically defined, Control-M/Tape invokes the CTTSE04 Control-M/Tape security module to determine if the user is allowed to perform the action requested. (For more information, see parameters DYNVOL and DYNBS in member CTTPARM.)

Authority to Create and Print Tape Labels

Whenever a user or batch job requests that a tape label be created and or printed, Control-M/Tape invokes the CTTSE09 Control-M/Tape security module to determine if the user is allowed to perform the requested action.

Unauthorized access to the CTTTPI utility

The CTTTPI utility provides extensive facilities for protecting against the destruction of active volumes and for recovering information from files that have been partially overwritten.

The utility, which can be accessed only by authorized users, provides the following major functions:

- Nondestructive Initialization (INITT)
- Nondestructive Erasure (TAPERAS)
- Media Information Mapping (TAPEMAP)

The utility can be used by authorized users only.

You can restrict usage of the CTTTPI utility to a specific started task or user ID. You can also restrict usage of specific CTTTPI utility functions to a specific user ID, using the following command:

```
$IOAUTL.qname.CTTTPI.function resource
```

In the preceding example, function can be INITT, TAPEMAP, or TAPERAS.

Control-M/Tape Security Calls

Control-M/Tape Basic Definition Security Calls (on page 180)and Control-M/Tape Extended Definition Security Calls (on page 183) define the security calls of the Control-M/Tape definition modes.

Control-M/Tape Basic Definition Security Calls

Table 73 Control-M/Tape Basic Definition Security Calls

Protected Element	Class Entity Name	Explanation	Security Module
Controlling Media Database Updates from the Real-time Environment			
Controlling Control-M/Tape Initialization	FACILITY $$CTTINI.qname	qname is the name used to assign different authorizations to various Control-M/Tape environments (such as Test and Production).	CTTSE01
BLP parameter is specified	FACILITY $$CTTBLP.qname.volser This entity is not checked for Basic Definition mode unless TBLPCHK is set to YES.	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE03
EXPDT parameter is set to 98000	FACILITY $$CTTBYPASS.qname.volser	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE03
Controlling access to dynamically define a dataset on a tape volume	FACILITY $$CTTMDBDEF.qname		CTTSE04
Controlling Media Database Updates from the Online Environment, the Real-time Environment, or Control-M/Tape Utilities			

Protected Element	Class Entity Name	Explanation	Security Module
Requesting initialization in batch	FACILITY $$CTTVOL.qname		CTTSE06
Requesting bypass security	FACILITY $$CTTBYSEC.qname		CTTSE06
Performing volume checkout	FACILITY $$CTTVOL.qname.volser	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06
Returning a volume that was checked out	FACILITY $$CTTVOL.qname.volser	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06
Deleting a volume	FACILITY $$CTTVOL.qname.volser	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06
Unscratching a volume	FACILITY $$CTTVOL.qname.volser	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	
Selecting a volume	FACILITY $$CTTVOL.qname.extension	extension is the volume serial number or the dataset name, depending on the current request.	
Cleaning a volume	FACILITY $$CTTVOL.qname.volser	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06

Protected Element	Class Entity Name	Explanation	Security Module
Inserting a dataset record Inserting a volume record	DATASET dsname FACILITY $$CTTVOL.qname.volser	dsname is the requested dataset name on the tape volume. volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06
Updating a dataset record Updating a volume record	DATASET dsname FACILITY $$CTTVOL.qname.volser	dsname is the requested dataset name on the tape volume. volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06
Deleting a dataset record Deleting a volume record	DATASET dsname FACILITY $$CTTVOL.qname.volser	dsname is the requested dataset name on the tape volume. volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06
Selecting a dataset Selecting a volume dataset	DATASET dsname FACILITY $$CTTVOL.qname.volser	dsname is the requested dataset name on the tape volume. volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06
Causing a dataset to expire	DATASET dsname	dsname is the requested dataset name on the tape volume.	CTTSE06
Extending the expiration date of a dataset	DATASET dsname	dsname is the requested dataset name on the tape volume.	CTTSE06
Sending a volume to a specified vault	FACILITY $$CTTVOL.qname.volser	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06

Protected Element	Class Entity Name	Explanation	Security Module
Recalling a volume	FACILITY $$CTTVOL.qname.volser	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06
Displaying related volumes in the multivolume set to which this volume belongs	FACILITY $$CTTVOL.qname.volser	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06
Displaying Additional Dataset Information	DATASET dsname	dsname is the requested dataset name on the tape volume.	CTTSE06
Displaying Additional Volume Information	FACILITY $$CTTADDINF.qname.extension	extension is the volume serial number or the dataset name, depending on the current request.	CTTSE06
Unscratching a Volume	FACILITY $$CTTUNSCR.qname.volser	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06
Create and print tape label	FACILITY $$CTTPRLAB.qname		CTTSE09

Control-M/Tape Extended Definition Security Calls

Table 74 Control-M/Tape Extended Definition Security Calls

Protected Element	Class Entity Name	Explanation	Security Module
Controlling Media Database Updates from the Real-time Environment			
BLP parameter is specified	FACILITY $$CTTBLP.qname.volser This entity can only be checked for Extended Definition mode unless TBLPCHK is set to YES.	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE03

Protected Element	Class Entity Name	Explanation	Security Module
Controlling Control-M/Tape Initialization	FACILITY $$CTTINI.qname	qname is the name used to assign different authorizations to various Control-M/Tape environments (such as Test and Production).	CTTSE01
EXPDT parameter is set to 98000	FACILITY $$CTTBYPASS.qname.volser	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE03
Controlling access to dynamically define a tape volume	FACILITY $$CTTVOLDEF.qname.volser	volser is the volume serial number of the requested tape volume.	CTTSE04
Controlling access to dynamically define a tape volume	FACILITY $$CTTVOLDEF.qname.volser.dsn	volser is the volume serial number of the requested tape volume, and dsn is the requested dataset name of the tape volume.	CTTSE04
Controlling Media Database Updates from the Online Environment, the Real-time Environment, or Control-M/Tape Utilities			
Requesting initialization in batch	FACILITY $$CTTINIT.qname		CTTSE06
Requesting bypass security	FACILITY $$CTTBYSEC.qname		CTTSE06
Performing volume checkout	FACILITY $$CTTCHKOUT.qname.volser	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06
Returning a volume that was checked out	FACILITY $$CTTBACKLB.qname.volser	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06

Protected Element	Class Entity Name	Explanation	Security Module
Deleting a volume	FACILITY $$CTTDELVOL.qname.volser	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06
Unscratching a volume	FACILITY $$CTTUNSCR.qname.volser	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06
Selecting a volume	FACILITY $$CTTSELECT.qname.extension	extension is the volume serial number or the dataset name, depending on the current request.	CTTSE06
Selecting a dataset	FACILITY $$CTTSELECT.qname.dsname		
Cleaning a volume	FACILITY $$CTTCLNVOL.qname.volser	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06
Inserting a dataset record Inserting a volume record	FACILITY $$CTTRECINS.qname.dsname FACILITY $$CTTRECINS.qname.volser	dsname is the requested dataset name on the tape volume. volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06
Updating a dataset record Updating a volume record	FACILITY $$CTTRECUPD.qname.dsname FACILITY $$CTTRECUPD.qname.volser	dsname is the requested dataset name on the tape volume. volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06

Protected Element	Class Entity Name	Explanation	Security Module
Deleting a dataset record	FACILITY $$CTTRECDEL.qname.dsname	dsname is the requested dataset name on the tape volume.	CTTSE06
Deleting a volume record	FACILITY $$CTTRECDEL.qname.volser	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	
Selecting a dataset	FACILITY $$CTTSELECT.qname.dsname	dsname is the requested dataset name on the tape volume.	CTTSE06
Selecting a volume	FACILITY $$CTTSELECT.qname.volser	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	
Causing a dataset to expire	FACILITY $$CTTEXPIRE.qname.dsname	dsname is the requested dataset name on the volume.	CTTSE06
Causing a volume to expire	FACILITY $$CTTEXPIRE.qname.volser	volser is the requested volume serial number of the volume.	CTTSE06
Extending expiration date of a dataset	FACILITY $$CTTEXTEND.qname.dsname	dsname is the requested dataset name on the volume.	CTTSE06
Extending expiration date of a volume	FACILITY $$CTTEXTEND.qname.volser	volser is the requested volume serial number of the volume.	CTTSE06
Sending a volume to a specified vault	FACILITY $$CTTVAULT.qname.volser	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06
Recalling a volume	FACILITY $$CTTRECALL.qname.volser	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06
Displaying related volumes in the multivolume set to which this volume belongs	FACILITY $$CTTGROUP.qname.volser	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06

Protected Element	Class Entity Name	Explanation	Security Module
Displaying Additional Dataset Information	FACILITY $$CTTADDINF.qname.dsname	dsname is the requested dataset name on the tape volume.	CTTSE06
Displaying Additional Volume Information	FACILITY $$CTTADDINF.qname.extension	extension is the volume serial number or the dataset name, depending on the current request.	CTTSE06
Unscratching a volume	FACILITY $$CTTUNSCR.qname.volser	volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.	CTTSE06
Create and print tape label	FACILITY $$CTTPRLAB.qname		CTTSE09

Implementing Control-M/Tape Security

This section details the steps required to implement the Control-M/Tape security interface.

The Control-M/Tape security interface can be installed either as part of the customized installation path, or during the Customization process after installation. Both options use the INCONTROL Installation and Customization Engine (ICE) application. If you are not familiar with the ICE interface, see the *INCONTROL for z/OS Installation Guide: Installing*.

Do not proceed with Control-M/Tape security implementation until IOA security is implemented.

➢ To install the Control-M/Tape security interface

a. Enter the main ICE screen.

b. Select Customization.

c. Enter CTT in the Product field.

d. Select Security Customization.

e. Perform all major and minor steps required to install the security product.

Step 1. Implement Control-M/Tape Security

Follow the steps below to implement Control-M/Tape security.

Step 1.1 Grant Access Permissions

Collect the data you need to define the INCONTROL entities and user authorizations to the security product.

In ICE, run the steps "Control-M/Tape Security Definitions (Sample)" and "Functions Security Definitions (Sample)" to create two sample jobs.

Step 1.2 Customize Security Parameters

Table 75 Control-M/Tape Security Parameters

Parameter	Description
DEFMCHKT	When choosing a definition mode as COND to any of the Control-M/Tape security modules, use qname together with the value given to this parameter as the high level qualifier, to determine the real definition mode to be used.
DSNCSE03	This parameter determines whether to perform a dataset authorization check in the CTTSE03 security module. Valid values are: ▪ YES — Perform dataset authorization check. Default ▪ NO — Do not perform dataset authorization check
DSNCSE06	This parameter determines whether to perform a dataset authorization check in the CTTSE06 security module. Valid values are: ▪ YES — Perform dataset authorization check. Default ▪ NO — Do not perform dataset authorization check
SECTOLT	This parameter determines the action to perform if your security product is inactive or a specific resource is not defined to the security product. Valid values are: ▪ YES —- Perform the action ▪ NO — Do not perform the action
TBLPCHK	This parameter determines whether the BLP parameter is checked when using Basic Definition mode. Valid values are: ▪ YES — Check parameter BLP in Basic Definition mode ▪ NO — Do not check parameter BLP in Basic Definition mode. Default

Parameter	Description
TRULCHK	During rule loading by CTTINIT, this parameter determines whether the CTTSE01 module checks if the dataset names or masks specified in the ON DATASET statement can be used by the owner of the rule. Valid values are: ▪ YES — Perform an ON DATASET authority check during rule loading. ▪ NO — Do not perform an ON DATASET authority check during rule loading. Default.
	Note: This check cannot be performed for rules with ON DATASET set to *, because it is impossible to verify if the owner of the rule is allowed to use any dataset in the site. Instead, the GRANTTB table in the CTTSE01 source member lists all the rules that can use this dataset masking.
TCHKINVL	During CHECK-IN process check VOLSER or SLNAME.

Table 76 Mode Parameters

Parameter	Description
Mode Definition	Specify one of the following values to determine the definition mode for the Control-M/Tape security modules: ▪ COND—Conditional Definition mode. Default. ▪ BASIC—Basic Definition mode. ▪ EXTEND—Extended Definition mode.
DFMT01	Definition mode for Control-M/Tape security module CTTSE01.
DFMT03	Definition mode for Control-M/Tape security module CTTSE03.
DFMT04	Definition mode for Control-M/Tape security module CTTSE04.
DFMT06	Definition mode for Control-M/Tape security module CTTSE06.
DFMT09	Definition mode for Control-M/Tape security module CTTSE09.

Step 1.3 Save Security Parameters into Product

This step saves all the security parameters specified for Control-M/Tape. When this step completes, the Status column is automatically updated to COMPLETE.

Step 2. RACF Security Definition Samples

Step 2.1 ControlM/Tape Security Definitions (optional)

Select this step to edit the CTTSRAC2 member in the IOA INSTWORK library.

Perform the following steps to define the required permissions.

1. Associate users with Extended Definition Mode.

   a. To define the entity $$CTTEDM.qname, use the following command:

   ```
   RDEFINE FACILITY $$CTTEDM.qname UACC(NONE)
   ```

   b. To authorize USERA to Extended Definition mode, use the following command:

   ```
   PERMIT $$CTTEDM.qname ID(USERA) CLASS(FACILITY) ACCESS(READ)
   ```

   c. Submit the job for execution.

   This job must be run under an administrator user ID who has authorization to enter these commands.

   d. Scan the output of the job for information and error messages produced. All job steps must end with a condition code of 0.

2. Define entities and user authorizations.

   For information about entities and user authorizations, see Control-M/Tape Basic Definition Security Calls (on page 180)and Control-M/Tape Extended Definition Security Calls (on page 183).

   Example

   To authorize USERA access to a given Control-M/Tape entity, use the following command:

   ```
   PERMIT $$CTTnnn.qname CLASS(FACILITY) ID(USERA) ACCESS(READ)
   ```

   where CTTnnn is the name of the Control-M/Tape entity to be accessed.

   All entity names for each Control-M/Tape protected element appear in Control-M/Tape Basic Definition Security Calls (on page 180) for Basic Definition mode and Control-M/Tape Extended Definition Security Calls (on page 183) for Extended Definition mode.

Step 2.2 Functions Security Definitions (Optional)

Select this step to edit the CTTSRAC3 member in the IOA INSTWORK library. This member contains definition samples for the various Control-M/Tape entities. Modify the definitions according to the requirements of the site and submit the job.

Step 2.3 Control Program Access to Datasets (Optional)

Select this step to edit the CTTSRAC4 member in the IOA INSTWORK library. This member contains a sample of the definitions required to define Program Pathing access authorizations to Control-M/Tape datasets. Review the definitions and modify to meet the requirements of your site.

BMC Software recommends that the security administrator first read Limiting Access to Specific Programs (on page 200).

Step 3. TopSecret Security Definition Samples

Step 3.1 ControlM/Tape Security Definitions (Optional)

Select this step to edit the CTTSTSS2 member in the IOA INSTWORK library.

1. Define entities and user authorizations to TopSecret.

   For information about how to define Control-M/Tape entities and user authorizations to TopSecret, see Control-M/Tape Basic Definition Security Calls (on page 180) and Control-M/Tape Extended Definition Security Calls (on page 183).

   a. Add the following command to add the resources to TopSecret:

      ```
      TSS ADD(sec-administrator-dept) IBMFAC($$CTT)
      ```

      Set the sec-administrator-dept parameter to the appropriate value.

      All entity names for each Control-M/Tape protected element appear in Control-M/Tape Basic Definition Security Calls (on page 180) for Basic Definition mode and Control-M/Tape Extended Definition Security Calls (on page 183) for Extended Definition mode.

2. Associate users with Extended Definition Modes.

   Authorizations to access Control-M/Tape datasets are defined during the Control-M/Tape installation process. This step must be completed before proceeding with security implementation. For details on how to grant users access to Control-M/Tape datasets, see the *INCONTROL for z/OS Installation Guide: Installing*.

   a. Add the following TopSecret commands to define the $$CTTEDM entity to TopSecret, and authorize users to this entity:

      TSS PERMIT(USERA) IBMFAC($$CTTEDM.qname) ACC(READ)

      Set the USERA parameter to the user ID of the Control-M/Tape installer.

Do not define the $$CTTEDM entity to operate in warning mode since this causes all users to operate in Extended Definition mode.

3. Authorize the Control-M/Tape installer to use Control-M/Tape facilities.

   a. Customize the following command to authorize USERA to Control-M/Tape facilities:

      ```
      TSS PERMIT(USERA) IBMFAC($$CTT) ACC(READ)
      ```

      Set the USERA parameter to the user ID of the Control-M/Tape installer.

4. Submit the job.

   This job must be run under the ACID of the general security administrator (SCA) who is authorized to enter these TopSecret commands.

   All job steps must end with a condition code of 0.

Step 3.2 Functions Security Definitions (Optional)

Select this step to edit the CTTSTSS3 member in the IOA INSTWORK library. This member contains definition samples for the various Control-M/Tape entities. Modify the definitions according to the requirements of the site and submit the job.

Step 3.3 Control Program Access to Datasets (Optional)

Select this step to edit the CTTSTSS4 member in the IOA INSTWORK library. This member contains a sample of the definitions required to define Program Pathing access authorizations to Control-M/Tape datasets. Review the definitions and modify to meet the requirements of your site.

BMC Software recommends that the security administrator first read Limiting Access to Specific Programs (on page 200).

Step 4. ACF2 Security Definition Samples

Step 4.1 ControlM/Tape Security Definitions (Optional)

Select this step to edit the CTTSSAF2 member in the IOA INSTWORK library.

Perform the following steps to define the required permissions.

1. Associate users with Extended Definition Mode.
   1. Edit the CTTSSAF2 member in the IOA INSTWORK library.
   2. Define and authorize entity $$CTDEDM.qname to ACF2/SAF and authorize users to use this entity using the following commands:

      ```
      SET RESOURCE(CMF)
      COMP
      $KEY($$CTTEDM.qname) TYPE(CMF)
      UID(USERA) ALLOW
      ```

2. Define entities and user authorizations to CA-ACF2/SAF.

   For information about entities and user authorizations, see Control-M/Tape Basic Definition Security Calls (on page 180), and Control-M/Tape Extended Definition Security Calls (on page 183).

   Example

   To authorize USERA (the user ID of the Control-M/Tape installer) to access a given Control-M/Tape entity, use the following command:

   ```
   SET RESOURCE(CMF)
   COMP
   $KEY($$CTTnnn.qname) TYPE(CMF)
   UID(USERA) ALLOW
   ```

   where qname is the name used to assign different authorizations to various Control-M/Tape environments (such as Test and Production). This parameter is specified during IOA installation.

   Set the USERA parameter to the UID string of the Control-M/Tape installer.

   All entity names for each Control-M/Tape protected element appear in Control-M/Tape Basic Definition Security Calls (on page 180) for Basic Definition mode and in Control-M/Tape Extended Definition Security Calls (on page 183) for Extended Definition mode.

3. Submit Job for Execution

   This job must be run under the user ID of an ACF2 administrator who has authorization to enter these ACF2 commands.

   Scan the output of the job for information and error messages produced by ACF2/SAF. All job steps must end with a condition code of 0.

Step 4.2 Functions Security Definitions (Optional)

Select this step to edit the CTTSSAF3 member in the IOA INSTWORK library. This member contains definition samples for the various Control-M/Tape entities. Modify the definitions according to the requirements of the site and submit the job.

Step 4.3 Control Program Access to Datasets (Optional)

Select this step to edit the CTTSSAF4 member in the IOA INSTWORK library. This member contains a sample of the definitions required to define Program Pathing access authorizations to Control-M/Tape datasets. Review the definitions and modify to meet the requirements of your site.

BMC Software recommends that the security administrator first read Limiting Access to Specific Programs (on page 200).

Control-M/Tape Security Interface Modules

This section describes the Control-M/Tape security interface modules.

Module CTTSE01

The CTTSE01 module is the security module of Control-M/Tape Exit CTTX001. This module verifies that the user is authorized to activate the Control-M/Tape initialization process.

The CLASS checked is FACILITY unless otherwise specified.

RACF Security

The entity used to check authorization is $$CTTINI.qname. The access level used to check this authorization is READ.

The following commands authorize USERA to activate the Control-M/Tape initialization process:

```
RDEFINE FACILITY $$CTTINI.qname UACC(NONE)
PERMIT $$CTTINI.qname ID(USERA) CLASS(FACILITY) ACCESS(READ)
```

TopSecret Security

The entity used to check authorization is: $$CTTINI.qname.

The access level used to check this authorization is READ.

The following sample commands authorize USERA to activate the Control-M/Tape initialization process:

TSS ADD(sec-administrator-dept) IBMFAC($$CTTINI.qname)
TSS PERMIT(USERA) IBMFAC($$CTTINI.qname) ACC(READ)

ACF2/SAF Security

The entity used to check authorization is $$CTTINI.qname. The access level used to check this authorization is READ.

The following sample ACF2 commands authorize USERA to activate the Control-M/Tape initialization process:

```
SET RESOURCE(CMF)
COMP
$KEY($$CTTINI.qname) TYPE(CMF)
 UID(USERA) ALLOW
```

Module CTTSE03

The CTTSE03 module is the security module of Control-M/Tape Exit CTTX003. This module verifies that the user is authorized to use JCL parameter BLP and to set JCL parameter EXPDT to 98000.

Basic Definition Mode

The entity used to check authorization depends on the value of parameter TBLPCHK in the security interface program. The following flag values are used:

- NO—Do not check parameter BLP in Basic Definition mode. Default.
- YES—Check parameter BLP in Basic Definition mode.

The CLASS checked is FACILITY. The entity used to check authorization is:

Table 77 CTTSE03 Basic Definition Authorization Entities

Entity	Use
$$CTTBLP.qname.volser	For parameter BLP (when TBLPCHK is set YES).
$$CTTBYPASS.qname.volser	For parameter EXPDT when set to 98000.

where volser is the volume serial number of the requested tape volume of a single volume file or the first volume of a multivolume file.

MVS/SAF always checks the usage of the BLP parameter using a class of FACILITY and an entity of ICHBLP.

Extended Definition Mode

The CLASS checked is FACILITY. The entity used to check authorization is:

Table 78 CTTSE03 Extended Definition Authorization Entities

Entity	Use
$$CTTBLP.qname.volser	For parameter BLP.
$$CTTBYPASS.qname.volser	For parameter EXPDT when set to 98000.

where:

volser: Volume serial number of the requested tape volume of a single volume file or the first volume of a multivolume file.

The commands listed below permit USERA to use JCL parameter BLP and to set JCL parameter EXPDT to 98000 for any tape volume.

For RACF:

```
RDEFINE FACILITY $$CTTBLP.qname.* UACC(NONE)
RDEFINE FACILITY $$CTTBYPASS.qname.* UACC(NONE)
PERMIT $$CTTBLP.qname.* ID(USERA) ACCESS(READ)
PERMIT $$CTTBYPASS.qname.* ID(USERA) ACCESS(READ)
```

For TopSecret:

```
TSS ADD(sec-administrator-dept) IBMFAC($$CTTBLP.qname)
TSS ADD(sec-administrator-dept) IBMFAC($$CTTBBYPASS.qname)
TSS PERMIT(USERA) IBMFAC($$CTTBLP.qname) ACC(READ)
TSS PERMIT(USERA) IBMFAC($$CTTBYPASS.qname) ACC(READ)
```

For ACF2 / SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($$CTTBLP.qname.************************) TYPE(CMF)
 UID(USERA) ALLOW
$KEY($$CTTBYPASS.qname.*********************) TYPE(CMF)
 UID(USERA) ALLOW
```

Module CTTSE04

The CTTSE04 module is the security module of Control-M/Tape Exit CTTX004. This module verifies that the user is authorized to dynamically define tape volumes and datasets in the Control-M/Tape Media database.

Basic Definition Mode

The CLASS checked is FACILITY. The entity is $$CTTMDBDEF.qname

The following commands authorize USERA to dynamically define tape volumes and datasets in the Control-M/Tape Media database:

For RACF:

```
RDEFINE FACILITY $$CTTMDBDEF.qname UACC(NONE)
PERMIT $$CTTMDBDEF.qname ID(USERA) CLASS(FACILITY) ACCESS(READ)
```

For TopSecret:

```
TSS ADD(sec-administrator-dept) IBMFAC($$CTTMDBDEF.qname)
TSS PERMIT(USERA) IBMFAC($$CTTMDBDEF.qname) ACC(READ)
```

For ACF2/SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($$CTTMDBDF.qname) TYPE(CMF)
 UID(USERA) ALLOW
```

Extended Definition Mode

The CLASS checked is FACILITY. The entity used to check authorization depends on the user's request.

For volume definition: $$CTTVOLDEF.qname.volser

For dataset definition: $$CTTVOLDEF.qname.volser.dataset-name

where volser is the volume serial number of the requested tape volume, and dataset is the requested dataset name on the tape volume.

The following sample commands authorize USERA to dynamically define datasets on the TAPE01 tape volume in the Control-M/Tape database:

For RACF:

```
RDEFINE FACILITY $$CTTVOLDEF.qname.TAPE01.* UACC(NONE)
PERMIT $$CTTINITDEF.qname.TAPE01.* ID(USERA) CLASS(FACILITY) ACCESS(READ)
```

For TopSecret:

```
TSS ADD(sec-administrator-dept) IBMFAC($$CTTVOLDEF.qname.TAPE01)
TSS PERMIT(USERA) IBMFAC($$CTTVOLDEF.qname.TAPE01) ACC(READ)
```

For ACF2/SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($$CTTVOLDEF.qname.TAPE01***************) TYPE(CMF)
 UID(USERA) ALLOW
```

Module CTTSE06

The CTTSE06 module is the security module of Control-M/Tape Exit CTTX006. This module verifies that the user is authorized to update the Media Database from the online environment (Inquire, Update, Check In), from the real-time environment (using Control-M/Tape SVC), or from Control-M/Tape utilities (for example, CTTVTM, CTTRTM).

When this module is invoked, a preliminary check is performed using a function called MDBOPEN. This function checks the user's access to Control-M/Tape Media Database data component, index component, and trace component files.

If the user is authorized to update these datasets, the Media Database components are opened and the user is allowed to perform actions on them. If update authority is denied, the datasets are opened for READ only access by the user.

Basic Definition Mode

The entity used to check authorization depends on the user's request.

For dataset operations:

CLASS checked is DATASET. The entity is dataset name of requested tape file.

For volume operations:

CLASS checked is FACILITY. The entity is $$CTTVOL.qname.volser

where volser is the volume serial number of the requested tape volume of a single volume operation, or the first volume of a multivolume operation.

The following commands permit USERA to perform any volume operation from the Inquire or Update screen in Basic Definition mode:

For RACF:

```
RDEFINE FACILITY $$CTTVOL.qname UACC(NONE)
PERMIT $$CTTVOL.qname ID(USERA) ACCESS(READ)
```

For TopSecret:

```
TSS ADD(sec-administrator-dept) IBMFAC($$CTTVOL.qname)
TSS PERMIT(USERA) IBMFAC($$CTTVOL.qname) ACC(READ)
```

For ACF2/SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($$CTTVOL.qname) TYPE(CMF)
 UID(USERA) ALLOW
```

In Basic Definition mode, security checks are bypassed for Control-M/Tape utilities CTTVTM and CTTRTM.

Extended Definition Mode

Online Environment

The CLASS checked is FACILITY. The entity used to check authorization depends on the type of request:

Table 79 CTTSE06 Extended Definition Authorization Entity

Request	Entity
INIT Control-M/Tape BATCH	$$CTTINIT.qname
CHECKIN	$$CTTRECINS.qname.extension
UPDATE RECORD	$$CTTRECUPD.qname.extension
DELETE RECORD	$$CTTRECDEL.qname.extension
CHECKOUT	$$CTTCHKOUT.qname.volser
BACK-IN-LIBRARY	$$CTTBACKLB.qname.volser
CLEAN	$$CTTCLNVOL.qname.volser

Request	Entity
DELETE	$$CTTDELVOL.qname.volser
EXPIRE	$$CTTEXPIRE.qname.dataset
EXTEND	$$CTTEXTEND.qname.dataset
VAULT	$$CTTVAULT.qname.volser
RECALL	$$CTTRECALL.qname.volser
SELECT	$$CTTSELECT.qname.extension
GROUP	$$CTTGROUP.qname.volser
ADDINFO	$$CTTADDINF.qname.extension
UNSCRATCH	$$CTTUNSCR.qname.volser

where extension is the volume serial number or dataset name, depending on the current request.

Real-time Environment

When a user tries to process a dataset, the Control-M/Tape SVC calls the CTTSE06 security module. The CLASS checked is FACILITY. The entity used to check authorization depends upon the user's request.

Table 80 Real-time Authorization Entities

Entity	Use
$$CTTRECINS.qname.volser	For dynamic definition of a volume
$$CTTRECINS.qname.dataset	For creation of a new dataset
$$CTTRECUPD.qname.volser	For any access to a volume
$$CTTRECUPD.qname.dataset	For any access to a dataset

Security checks are bypassed when called from the Control-M/Tape SVC environment.

Special Utility Processing

When called under the batch environment from a Control-M/Tape utility, the CTTSE06 module performs one additional security check for the INIT request.

The CLASS checked is FACILITY. The entity checked is $$CTTBYSEC.qname.

If the user is granted access to this entity, the user is permitted to work in a special mode and all further security checks are bypassed for improved batch utility performance.

This check is performed only after user authority to make INIT Control-M/Tape BATCH requests has been verified.

Module CTTSE09

The CTTSE09 module is the security module of Control-M/Tape Exit CTTX009. This module verifies that the user is authorized to create and print a tape label.

The CLASS checked is FACILITY. The entity used to check authorization is $$CTTPRLAB.qname.

The access level used to check this authorization is READ.

The following commands enable the user to create and print a tape label:

For RACF:

```
RDEFINE FACILITY $$CTTPRLAB.qname UACC(NONE)
PERMIT $$CTTPRLAB.qname ID(USERA) CLASS(FACILITY) ACCESS(READ)
```

For TopSecret:

```
TSS ADD(sec-administrator-dept) IBMFAC($$CTTPRLAB.qname)
TSS PERMIT(USERA) IBMFAC($$CTTPRLAB.qname) ACC(READ)
```

For ACF2/SAF:

```
SET RESOURCE(CMF)
COMP
$KEY($$CTTPRLAB.qname) TYPE(CMF)
 UID(USERA) ALLOW
```

7

Limiting Access to Specific Programs

- **Protecting Access to Datasets Through Specific Programs**

  A user’s access to a dataset can be protected by a security product. However, in MVS, the security products do not control a user’s access to part of a dataset. When a user is granted access to a dataset, the access applies to the entire dataset.

  When a dataset contains information that pertains to several users, it may be necessary to permit access to part of the data and deny access to the remaining data in the dataset.

  When it is necessary to authorize a user limited access to part of a dataset, the security administrator may need to rely on the application program to verify that the specific authorization applies only to the data that the user requires.

  To effectively use this type of protection, user access to the dataset must be restricted so that the user can access the dataset only when using the trusted application program. The security products used in MVS allow the security administrator to specify that a dataset can be accessed by a user only when using a specific program or programs.

  Whenever a dataset is protected with a conditional access rule such as "User U is allowed to access dataset D only through program P," dataset D is known as a Program-Accessed dataset (PADS) that is protected using a method called Program Pathing.

  Program pathing is a feature that is available in all security products. However, it is implemented differently in each environment. To use program pathing in IOA, the security administrator must be familiar with the program pathing method as it is implemented by the security product used at the site.

- **RACF Security**

  BMC Software recommends that you do not use RACF program pathing when working under ISPF

  A program-accessed dataset is a dataset that is protected by a dataset profile. The profile contains a conditional access list that specifies the user ID or group ID that allows access to a dataset only when a specific program is used.

  A controlled program is a module that is protected by a discrete or generic profile in the PROGRAM class. The controlled program definition may include either the PADCHK or the NOPADCHK attribute.

  When attribute PADCHK is specified during controlled program fetch from the LOADLIB by MVS, the program must appear in the conditional access list of any open program-accessed datasets.

  When attribute NOPADCHK is specified, the security product does not perform the program-accessed data check for the program. Therefore, any controlled program with attribute NOPADCHK that is loaded can access any currently open program-accessed dataset.

  Whenever a dataset is protected by a conditional access list and the user ID accessing the dataset is not authorized to access the dataset through the regular access list, the security product verifies the following:

  The program name exists in the conditional access list of the dataset profile with at least the requested level of authority. In addition, all programs in the active RB-chain exist in the conditional access list.

The user ID or group ID associated with the program name appears in the conditional access list.

The current task must not be previously loaded using a non-controlled program. Any other tasks in the address space that are previously loaded in a non-controlled program must not be dispatchable.

Therefore, after a non-controlled program is loaded by the task, no program- accessed dataset can be accessed by the task.

Generally, LOAD libraries from IBM and third party vendors can be treated as controlled libraries.

To allow program pathing to work correctly within the IOA environment, the IOA LOAD library must be defined as a controlled library using the following command:

RDEFINE PROGRAM ** ADDMEM('ioa.load-lib'/volser/NOPADCHK)

For more information, refer to the *IBM Resource Access Control Facility Security Administrator's Guide*.

- **TopSecret Security**

  A program-accessed dataset is a dataset that is protected by a dataset profile, with a conditional access list, specifying the user ID or group ID allowed to access a dataset only when using a specific program (that is, parameter PRIVPGM of the PERMIT command in TopSecret).

  In TopSecret, the program associated with the Current-TCB-Top-PRB appears in the PRIVPGM parameter (specifically, or using a pattern), and the access level specified in the ACCESS parameter of the PERMIT command is sufficient.

  In addition, the LIBRARY parameter of the PERMIT command allows the security interface to specify a LOAD library from which the program must originate. If the LIB parameter is not specified, the program must originate from the LNKLST libraries concatenation.

  It is highly recommended that the TopSecret PERMIT command be used because this command allows specification of the program prefix in parameter PRIVPGM. For example, to allow a user to access the Control-M Resources file only through INCONTROL products, use the following command:

  TSS PERMIT(USERID) DATASET (IOA.V600.RES) ACC(UPDATE)

  PRIVPGM(IOA(G), CTM(G)) LIBRARY(IOA.V600.LOAD)

- **ACF2/SAF Security**

  A program-accessed dataset is a dataset that is protected by a dataset profile. A conditional access list specifies user IDs or group IDs that are allowed to access a dataset only when using a specific program.

  The program associated with the Current-TCB Top-PRB appears in the PGM parameter (specifically or using a pattern) of the access rule and the requested access level allowed by the access rule.

  To define a program path protected dataset under CA-ACF2, the security administrator must code a condition access rule containing the program name and the library through which the access is granted. The following sample command creates a rule to allow a user to access the IOA Conditions file using INCONTROL products:

  $KEY (IOA)

  V600.RES UID(user) LIBRARY(IOA.V600.LOAD) PGM(IOA-) READ(A) WRITE(A)

8

IOA Security Interface Support for Session Managers

Many MVS sites use product packages that enable multiple VTAM sessions to run simultaneously, thereby enabling users to alternate between different sessions. These packages allow users to perform a single logon to the session manager product, and then log on to other VTAM applications without having to specify the user ID and password. Using this method, users can skip the logon screen whenever they log on to the IOA Online monitor (IOAOMON1). For this purpose, logon data in free-form text is passed on to the VTAM application using the following command:

LOGON APPLID(*vtam_appl_name*) DATA(*tranid*+*userid*+*pwd*)

where

- *vtam_appl_name* is the name of the VTAM application
- *tranid* is a 4-character IOA tran ID
- *userid* is an 8-byte user ID
- *pwd* is an 8-byte password or password phrase from 14 to 100 characters in length

Whenever the logon data field is specified and the user ID and password or password phrase was verified successfully, the IOASE09 security module notifies the caller not to display the logon screen.

Deactivating IOA and INCONTROL Product Security

To deactivate the IOA security or any INCONTROL product security, the security administrator must delete the following entities in the relevant security product:

- For IOA, delete entity $$SECIOA.qname
- For Control-M, delete entity $$SECCTM.qname
- For Control-D, delete entity $$SECCTD.qname
- For Control-O, delete entity $$SECCTO.qname
- For Control-M/Analyzer, delete entity $$SECCTB.qname
- For Control-M/Tape, delete entity $$SECCTT.qname

Where qname is defined during the IOA installation process.

When you deactivate security, you must also recycle the STC that has security activated and all users must exit and reenter the IOA Online facility.

10

Security Module Trace Level Numbers

When working in the IOA Online facility it may be necessary to activate or deactivate the trace function. The following table shows the commands used to activate or deactivate the trace function:

Table 81 Trace Function Activation and Deactivation Commands

Trace Function	Command

ON	SET TRACE=nnn
	or
	SET TRACE=nnn,ON
	where nnn is the trace level.

OFF	SET TRACE=nnn,OFF where nnn is the trace level.

The following table shows the trace levels assigned to specific security modules:

Table 82 Security Module Trace Levels

Security Module	Trace Level
IOASE06	210
IOASE07	211
IOASE09	212
IOASE12	213
IOASE16	214
IOASE32	215
IOASE40	216
IOASE42	217

Security Module	Trace Level
CTMSE01	221
CTMSE02	222
CTMSE08	223
CTDSE01	224
CTDSE04	225
CTDSE08	226
CTDSE19	227
CTDSE23	228
CTDSE24	229
CTDSE26	230
CTDSE27	246
CTDSE28	247
CTOSE01	231
CTOSE02	232
CTOSE03	233
CTOSE04	234
CTOSE08	235
CTOSE15	236

Security Module	Trace Level
CTTSE01	237
CTTSE03	238
CTTSE04	239
CTTSE06	240
CTTSE09	241
CTBSE01	242
CTBSE03	243
CTBSE04	244
CTBSE08	245