CICS® Transaction Server for VSE/ESA™

Security Guide

Release 1

SC33-1942-00

CICS® Transaction Server for VSE/ESA™

Security Guide

Release 1

SC33-1942-00

Note!

Before using this information and the product it supports, be sure to read the general information under “Notices” on page 195.

First Edition (June 1999)

This edition applies to Release 1 of CICS Transaction Server for VSE/ESA, program number 5648-054, and to all subsequent versions, releases, and modifications until otherwise indicated in new editions. Make sure you are using the correct edition for the level of the product.

Order publications through your IBM representative or the IBM branch office serving your locality.

At the back of this publication is a page entitled “Sending your comments to IBM”. If you want to make any comments, please use one of the methods described there.

Contents

Preface . . . vii
Notes on terminology . . . viii

Part 1. Introduction . . . 1

Chapter 1. Security facilities in CICS . . . 3
Why CICS needs security . . . 3
What CICS security protects . . . 4
What CICS security does not protect . . . 4
Terminal user security . . . 4
Preset terminal security . . . 5
Non-terminal security . . . 5
Transaction security . . . 6
CICS resource security . . . 6
CICS command security . . . 6
Surrogate user security . . . 6
QUERY SECURITY command . . . 7
APPC (LU6.2) session security . . . 7
Multiregion operation (MRO) security . . . 8
Report Controller security . . . 8
Front End Programming Interface security . . . 8
Generating and using PassTickets . . . 8

Chapter 2. Facilities provided by an external security manager . . . 9
Overview . . . 9
ESM user definitions . . . 9

Part 2. Implementing protection for a single-region CICS . . . 17

Chapter 3. CICS data set and system security . . . 19
Specifying the CICS region userid . . . 19
Authorizing access to CICS data sets . . . 22
Authorizing access to VSE libraries . . . 23
Authorizing access to the CICS region . . . 24
Authorizing the CICS region userid as surrogate user . . . 24
Defining security-related system initialization parameters . . . 24

Chapter 4. Verifying CICS users . . . 31
Identifying CICS terminal users . . . 31
Sign-on process . . . 31
Sign-off process . . . 33
Auditing sign-on and sign-off activity . . . 34
Controlling access to CICS from specific ports of entry . . . 35
Preset terminal security . . . 35
Using a VSE system console as a CICS terminal . . . 38
Obtaining CICS-related data for a user . . . 38
National language and non-terminal transactions . . . 39

Chapter 5. Transaction security . . . 41

CICS parameters controlling transaction-attach security . . . 41
Defining transaction profiles to the ESM . . . 42
Transactions not associated with a terminal . . . 43

Chapter 6. Resource security . . . 45
General resource security checking . . . 45
Security for general resource types . . . 48
Security checking of transactions running under CEDF . . . 56

Chapter 7. Surrogate user security . . . 59
Where surrogate user checking applies . . . 59
ESM definitions for surrogate user checking . . . 62

Chapter 8. CICS command security . . . 65
CICS resources subject to command security checking . . . 65
Parameters for specifying command security . . . 67
Security checking of transactions running under CEDF . . . 69
CEMT considerations . . . 70
Authorization failures . . . 70

Chapter 9. Security checking using the QUERY SECURITY command . . . 73
How the QUERY SECURITY mechanism works . . . 73
QUERY SECURITY RESTYPE . . . 74
QUERY SECURITY RESCLASS . . . 77
Querying a user's surrogate authority . . . 77
Logging for QUERY SECURITY RESTYPE and RESCLASS . . . 78
Uses for QUERY SECURITY RESTYPE and RESCLASS . . . 78

Chapter 10. Security for CICS-supplied transactions . . . 81
Categories of CICS-supplied transactions . . . 81

Part 3. Intercommunication security . . . 87

Chapter 11. Overview of intercommunication security . . . 89
Introduction . . . 89
Planning for intercommunication security . . . 90
Summary of intercommunication security levels . . . 92
Implementing intercommunication security . . . 92

Chapter 12. Implementing LU6.2 security . . . 95
Bind-time security with LU6.2 . . . 95
Link security with LU6.2 . . . 99
User security with LU6.2 . . . 100
SNA profiles and attach-time security . . . 105
Attach-time security and the USEDFLTUSER option . . . 106
Transaction, resource, and command security with LU6.2 . . . 107
Transaction routing security with LU6.2 . . . 108
Function shipping security with LU6.2 . . . 109
Distributed program link security with LU6.2 . . . 110
Security checking done in AOR with LU6.2 . . . 112
Summary of resource definition options for LU6.2 security . . . 114

Chapter 13. APPC password expiration management . . . 115

Introduction to APPC password expiration management . . . 115
What you require to use APPC PEM . . . 116
Roles of PEM client and CICS PEM server . . . 116
APPC PEM processing . . . 119
Overview of APPC PEM processing . . . 119
Setting up the PEM client . . . 124
PEM client input and output data . . . 127

Chapter 14. Implementing LU6.1 security . . . 137
Link security with LU6.1 . . . 137
Specifying ATTACHSEC with LU6.1 . . . 138
Transaction, resource, and command security with LU6.1 . . . 138
Function shipping security with LU6.1 . . . 139
Security checking done in AOR with LU6.1 . . . 140
Summary of resource definition options for LU6.1 security . . . 141

Chapter 15. Implementing MRO security . . . 143
Bind-time security with MRO . . . 143
Logon security checking with MRO . . . 143
Link security with MRO . . . 145
User security with MRO . . . 146
Transaction, resource, and command security with MRO . . . 149
Transaction routing security with MRO . . . 150
Function shipping security with MRO . . . 152
Distributed program link security with MRO . . . 153
Security checking done in AOR with MRO . . . 154
Summary of resource definition options for MRO security . . . 155

Chapter 16. Security for shared data tables . . . 157
Overview . . . 157
Security checking . . . 157
LOGON security check . . . 158
CONNECT security checks . . . 158

Chapter 17. Security for the Report Controller facility and CICS SPOOL interface . . . 161
ESM requirement . . . 161
Retention of RSL . . . 161
Mapping of RSL values to ESM resource names . . . 161
Report security checking . . . 162
Controlling RCF printers. . . . 162
Report Browse . . . 163

Part 4. Customization . . . 165

Chapter 18. Customizing security processing . . . 167
Installation data parameter list . . . 167
CICS security control points . . . 167
Determining the userid of the CICS region . . . 169
Specifying user-defined resources to the ESM . . . 170

Part 5. Migration and coexistence . . . 173

Chapter 19. Migration considerations 175

Part 6. Problem determination 177

Chapter 20. Problem determination in a security environment 179
Resolving problems when access is denied incorrectly 179
Resolving problems when access is allowed incorrectly 181
CICS initialization failures related to security 182
Password expiry management problem determination 185

Part 7. Appendix 187

Appendix A. Resource and command check cross reference 189

Notices 195
Trademarks and service marks 196

Bibliography 197
Books from VSE/ESA 2.4 base program libraries 198
Books from VSE/ESA 2.4 optional program libraries 200

Index 203

Preface

What this book is about

This book is about the security requirements and facilities available when CICS Transaction Server for VSE/ESA Release 1 is used in conjunction with an external security manager (ESM).

CICS provides an interface to an external security manager (ESM), which may be either user-written, the basic security manager (BSM) supplied by VSE/ESA, or a product supplied by a vendor. CICS security uses, via the RACROUTE macro, the VSE/ESA system authorization facility (SAF) interface to route authorization requests to the ESM. For more information about the CICS ESM interface, including the control points at which CICS issues a RACROUTE macro to route authorization requests, see the *CICS Customization Guide*.

This book is limited to describing the security functions required from within CICS. Each function is dependent on the required support being available in the installed ESM. In each case, you must refer to the appropriate ESM documentation, to determine whether the function is supported and how to implement it. For more information about the Basic Security Manager (BSM), see the *VSE/ESA Administration Manual*.

Who this book is for

This book is intended for security administrators responsible for controlling access to resources used by CICS. These resources are used by CICS terminals, users, or transactions in CICS regions, and by CICS application programs running in those regions. The book will also be of interest for CICS system programmers who may need to communicate their requirements to the security administrator for their installation.

What you need to know to understand this book

It is assumed that you have a good working knowledge of security requirements. It is also assumed that you know something about the types of resource owned and controlled by CICS.

How to use this book

The parts and chapters of this book are self-contained. Use an individual part or chapter where it contains information about the particular task you are engaged in. For example, see Chapter 18, “Customizing security processing” on page 167 if your task is to customize your CICS security processing.

Notes on terminology

The terms listed in Table 1 are commonly used in the CICS Transaction Server for VSE/ESA Release 1 library. See the *CICS Glossary* for a comprehensive definition of terminology.

Table 1 (Page 1 of 2). Commonly used words and abbreviations

Term	Definition (and abbreviation if appropriate)
$(the dollar symbol)	In the character sets and programming examples given in this book, the dollar symbol ($) is used as a national currency symbol and is assumed to be assigned the EBCDIC code point X'5B'. In some countries a different currency symbol, for example the pound symbol (£), or the yen symbol (¥), is assigned the same EBCDIC code point. In these countries, the appropriate currency symbol should be used instead of the dollar symbol.
BSM	BSM is used to indicate the basic security management supplied as part of the VSE/ESA product. It is RACROUTE-compliant, and provides the following functions: • Signon security • Transaction attach security
C	The C programming language
CICSplex	A CICSplex consists of two or more regions that are linked using CICS intercommunication facilities. Typically, a CICSplex has at least one terminal-owning region (TOR), more than one application-owning region (AOR), and may have one or more regions that own the resources accessed by the AORs
CICS Data Management Facility	The new facility to which all statistics and monitoring data is written, generally referred to as “DMF”
CICS/VSE	The CICS product running under the VSE/ESA operating system, frequently referred to as simply “CICS”
COBOL	The COBOL programming language
DB2 for VSE/ESA	Database 2 for VSE/ESA which was previously known as “SQL/DS”.

Table 1 (Page 2 of 2). Commonly used words and abbreviations

Term	Definition (and abbreviation if appropriate)
ESM	ESM is used to indicate a RACROUTE-compliant external security manager that supports some or all of the following functions: • Signon security • Transaction attach security • Resource security • Command security • Non-terminal security • Surrogate user security • MRO/ISC security (MRO, LU6.1 or LU6.2) • FEPI security.
FOR (file-owning region)—also known as a DOR (data-owning region)	A CICS region whose primary purpose is to manage VSAM and DAM files, and VSAM data tables, through function provided by the CICS file control program.
IBM C for VSE/ESA	The Language Environment-conforming version of the C programming language compiler. Generally referred to as "C/VSE".
IBM COBOL for VSE/ESA	The Language Environment-conforming version of the COBOL programming language compiler. Generally referred to as "COBOL/VSE".
IBM PL/I for VSE/ESA	The Language Environment-conforming version of the PL/I programming language compiler. Generally referred to as "PL/I VSE".
IBM Language Environment for VSE/ESA	The common runtime interface for all LE-conforming languages. Generally referred to as "LE/VSE".
PL/I	The PL/I programming language
VSE/POWER	Priority Output Writers Execution processors and input Readers. The VSE/ESA spooling subsystem which is exploited by the report controller.
VSE/ESA System Authorization Facility	The new VSE facility which enables the new security mechanisms in CICS, generally referred to as "SAF"
VSE/ESA Central Functions component	The new name for the VSE Advanced Function (AF) component
VSE/VTAM	"VTAM"

Part 1. Introduction

This part introduces you to the subject of CICS security, using an ESM as the CICS external security manager. It provides an overview of the CICS security requirements, and the facilities an ESM provides to satisfy those requirements. Part 1 contains the following:

- Chapter 1, “Security facilities in CICS” on page 3 introduces you to the various aspects of CICS transaction and resource security.
- Chapter 2, “Facilities provided by an external security manager” on page 9 describes the basic facilities that an ESM provides, and that CICS relies upon for its security administration.

Chapter 1. Security facilities in CICS

This chapter describes, from the CICS viewpoint, the following aspects of CICS transaction and resource security:

- "Why CICS needs security"
- "What CICS security protects" on page 4
- "What CICS security does not protect" on page 4
- "Terminal user security" on page 4
- "Preset terminal security" on page 5
- "Non-terminal security" on page 5
- "Transaction security" on page 6
- "CICS resource security" on page 6
- "CICS command security" on page 6
- "Surrogate user security" on page 6
- "QUERY SECURITY command" on page 7
- "APPC (LU6.2) session security" on page 7
- "Multiregion operation (MRO) security" on page 8
- "Generating and using PassTickets" on page 8

Why CICS needs security

Today, an unprecedented number of computer system users are completely dependent on their systems, and on the data managed by those systems. There are now terminals in many different locations in most organizations, and their use is commonplace. At the same time, easy-to-use, high-level inquiry languages are available, and there is much greater familiarity with data processing methods. This means that more and more people can use computers to retrieve or modify data stored within a computer system.

The speed, flexibility, and size of modern systems make large quantities of data accessible to many terminal users. As the systems become easier to use, there is also more scope for terminal users to gain access to confidential or valuable data.

Without a corresponding growth in awareness of good data security practices, these advances can result in accidental (or deliberate) data exposure. This means that your data can be subject to:

- Unauthorized access
- Disclosure
- Modification
- Destruction

As an online transaction-processing system (often supporting many thousands of terminals), CICS clearly needs the protection of a security system to ensure that the resources to which it manages access are protected, and are secure from unauthorized access.

To provide the necessary security for your CICS regions, CICS uses the VSE/ESA system authorization facility (SAF) to route authorization requests to an external security manager (ESM) at appropriate points within CICS transaction processing.

What CICS security protects

Let us take a brief look at the assets that CICS manages, and potential exposures. The assets are the application programs, the application data, and the application output. To prevent disclosure, destruction, or corruption of these assets, you must first safeguard the CICS system components themselves.

There are two distinct areas from which exposures to the CICS system can arise. The first of these is from sources external to CICS, for example, unauthorized access from batch users.

The other potential area of exposure arises from CICS users. CICS provides a variety of security and control mechanisms. These can limit the activities of CICS terminal users to only those functions that any particular individual user is authorized to use.

What CICS security does not protect

CICS itself does **not** provide facilities to protect its own assets from external access. You should restrict access to the program libraries, to the CICS regions, and to those responsible for incorporating approved application and system changes. Similarly, the data sets and databases used by CICS and by CICS applications must be accessible only by approved batch processing and operations procedures.

CICS does not protect your system from application programs that use undocumented or unsupported interfaces to bypass CICS security. You are responsible for ensuring that such programs are not installed on your system.

CICS does not protect your application source libraries. You should ensure that procedures are established and followed that prevent the introduction of unauthorized or untested application programs into your “production” application base. You should also protect the integrity of your system by exercising control over libraries that are admitted to the system, and changes to those libraries.

Terminal user security

To secure resources from unauthorized access, CICS needs some means of uniquely identifying individual users of the system. For this purpose, first define the users to the ESM by creating an entry in the ESM database, referred to as a **user profile**. To identify themselves to CICS, users sign on by specifying their ESM user identification (userid) and the associated password, or operator identification card (OIDCARD) (if the card reader supports the DFHOPID attention identifier (AID)) in the CICS-supplied sign-on transaction, CESN. Alternatively, they can use an equivalent transaction developed by your own installation by issuing the EXEC CICS SIGNON command provided for this purpose.

When users enter the CESN transaction, CICS verifies userids and passwords by a call to the ESM. If the terminal user signon is valid, the CICS user domain keeps

track of the signed-on user. Thereafter, CICS uses the information about the user when calling the ESM to make authorization checks. See Chapter 4, “Verifying CICS users” on page 31 for information about using terminal user security in CICS.

Preset terminal security

For some selected terminals, and consoles when used as CICS terminals, consider using CICS preset terminal based security as an alternative to terminal user security. A terminal becomes a preset security terminal when you specify the USERID operand on the terminal definition.

CICS preset terminal security allows you to associate a userid permanently with a terminal that is defined to CICS. This means that CICS implicitly “signs on” the terminal when it is being installed, instead of the terminal being signed on subsequently. Preset security is often defined for devices without keyboards, such as printers, at which users cannot sign on.

You can also use this form of security on ordinary display terminals as an alternative to terminal user security. This permits anyone with physical access to a terminal with preset security to enter the transactions that are authorized for that terminal, without the need to sign on to CICS. The terminal remains signed on as long as it is installed, and no explicit sign-off can be performed against it. If the userid associated with a display terminal with preset security authorized to use any sensitive transactions, ensure that the terminal is in a secure location to which access is restricted. For example, terminals physically located within a CICS network control center might be appropriate for preset security.

You can use preset security to assign a userid with **lower** authority than the default, for terminals in unrestricted areas.

For example, to define a terminal with preset security, use the CICS (CEDA) command as follows:

```
CEDA DEFINE TERMINAL(cics_termid) NETNAME(vtam_termid) USERID(userid)
            TYPETERM(cics_typeterm)
```

For further information on preset security terminals in the transaction routing environment refer to “Preset-security terminals and transaction routing” on page 108 (LU6.2 security) and “Preset-security terminals and transaction routing” on page 151 (MRO security).

Non-terminal security

You can also specify security for transactions that are not associated with terminals. These are:

- Started non-terminal transactions
- Transient data trigger-level transactions
- Program List Table (PLT) programs that run during CICS initialization

Security for transactions not associated with terminals may also be affected by surrogate security. For more information about non-terminal security, see "Transactions not associated with a terminal" on page 43.

Transaction security

CICS facilities for transaction security ensures that CICS calls the ESM each time a transaction is initiated, to verify that the userids associated with that transaction are permitted access to it.

See Chapter 5, "Transaction security" on page 41 for information about using transaction security.

CICS resource security

You can control access to CICS resources that a transaction uses. You do this by specifying YES on the resource security parameter, RESSEC, in the CICS TRANSACTION resource definition. These CICS resources can be:

- Application programs
- Files—VSAM and DAM
- Journals
- Temporary storage queues
- Transient data queues
- Transactions initiated by a CICS START command

See Chapter 6, "Resource security" on page 45 for information about using CICS resource security.

CICS command security

You can control security for a system programming subset of the CICS application programming interface (SPI) commands. You do this by specifying YES in the command security parameter, CMDSEC, on the CICS TRANSACTION resource definition. This is known as CICS command security, and operates on all the commands that require the special CICS translator option, SP. (These can be seen in Table 12 on page 66). Command security operates in addition to any transaction or resource security you define for a transaction. For example, if a user is permitted to use a transaction called FILA, which issues an EXEC CICS INQUIRE FILE command that the user is **not** permitted to use, CICS issues a "not authorized" (NOTAUTH) condition in response to the command, and the command fails.

See Chapter 8, "CICS command security" on page 65 for information about using CICS command security.

Surrogate user security

CICS performs surrogate user security checking in a number of instances to ensure that a surrogate user is authorized to act for another user. For more information see Chapter 7, "Surrogate user security" on page 59.

Surrogate user checking can be enforced for:

- CICS default user
- Started transactions
- Preset terminal security
- PLT security
- EXCI calls
- Installation of transient data queues.

QUERY SECURITY command

In addition to using CICS security checking for CICS-controlled resources (or as an alternative to it), you can use the EXEC CICS QUERY SECURITY command to control security access within the CICS application. This method also allows you to define security profiles to the ESM for resources other than CICS resource profiles, and enables a more detailed level of security checking than is available through the standard resource classes.

See Table 4 on page 14 for information about the resource classes that are supported for resource security checking within transactions. For more information about resource security checking, see Chapter 6, “Resource security” on page 45.

APPC (LU6.2) session security

So far, all the discussion has been about the security CICS performs for transactions running within a single CICS region, with its own resources and terminal network. A number of CICS regions can also be connected by means of **intercommunication**; for example, **intersystem communication (ISC)** using an SNA access method, such as VTAM, to provide the necessary communication protocols. This method is normally used for communication between CICS regions residing in different host computers, but it can also connect CICS regions in the same host computer. (See the *CICS Intercommunication Guide* for more information about CICS intercommunication facilities.)

One of the ISC protocols that CICS uses is for advanced program-to-program communication (APPC), which is the CICS implementation of the LU6.2 part of the SNA architecture.

For interconnected systems, the same basic security principles apply, but the resource definition is more complex, and you have additional security requirements. CICS treats APPC sessions, connections, and partners as resources, all of which have security requirements. In addition to the transaction, resource, and command security introduced earlier, CICS provides the following security mechanisms for the APPC environment:

- Bind-time (or session) security, prevents an unauthorized connection between CICS regions.
- Link security defines the authority of the remote system to access transactions or resources to which the connection itself is not authorized.
- User security checks that a user is authorized both to attach a transaction and to access all the resources and SP-type commands that the transaction is programmed to use.

See Chapter 12, “Implementing LU6.2 security” on page 95 for more information.

Multiregion operation (MRO) security

Another means of using intercommunication is **multiregion operation** (MRO). This is available for links between CICS regions in the same VSE image, independent of the systems network architecture (SNA) access method. See Chapter 15, “Implementing MRO security” on page 143 for information about MRO security.

Report Controller security

Report Controller security is discussed in more detail in Chapter 17, “Security for the Report Controller facility and CICS SPOOL interface” on page 161, and the *CICS Report Controller Planning Guide*.

Front End Programming Interface security

The security options provided for the Front End Programming Interface are equivalent to those provided for CICS command security (see page 6). Front End Programming Interface security is not discussed in this book, but in the *CICS Front End Programming Interface User's Guide*.

Generating and using PassTickets

A PassTicket is a program-generated character string that can be used in place of a password, with the following constraints:

- A specific PassTicket may be used for authentication **once**.
- The PassTicket has a limited lifespan, and must be used within the lifespan time of being generated.
- To ease the problem of system time differences, a specific PassTicket can be used up to the limited time, earlier or later in a target system, compared to the generating system.

Front end programming interface (FEPI) security can generate a PassTicket for use on a target system. The PassTicket can be used anywhere a password can be used.

Note: The PassTicket generation and validation algorithm means that the system that creates the PassTicket and the system that validates it must both use the same level of this function. That is, if the creating system has the function applied, and the validating system does not, the PassTicket is invalid.

Chapter 2. Facilities provided by an external security manager

For its security management capability, CICS relies on a number of facilities provided by an ESM. Although an ESM provides the basic security access and authorization facilities, it does not by itself perform any security checking.

This chapter discusses general facilities which may be available in an ESM and which are relevant to CICS security. Not all ESMs support all of these facilities. You should refer to your ESM documentation for details, This chapter covers:

- "Overview"
- "ESM user definitions"

Overview

An external security manager may provide the following facilities:

- The necessary functions to record information identifying individual users of system resources, and information identifying the resources that require protection.
- The facilities to define which users, or groups of users, are either permitted access, or excluded from access, to the resources for which profiles have been defined.
- A method to process requests, issued by subsystems or jobs running in a VSE/ESA system, to authenticate the identity of users defined to the ESM, and to check their access authorization to resources.
- The facilities for logging security-related events, such as users signing on and signing off, the issuing of ESM commands, and attempts to access protected resources.

ESM user definitions

ESMs holds user data in the form of user profiles in a database. These user profiles consist of one or more segments— ESM-specific segments, and others that are optional. For CICS users, the important segments are:

- The ESM-specific segment(s), which holds the basic information for a user.
- The CICS segment, which holds data for each CICS user
- The LANGUAGE segment, which specifies the user's national language preference

These segments are explained briefly in the following sections.

Table 2 on page 10 summarizes where the ESM userids for different types of CICS users are obtained.

Table 2. Types of CICS users and their userids	
User type	**Userid obtained from**
Region user	The userid under which the CICS region executes. It is specified in the USER parameter of the CICS startup ID statement.
CICS default user	The userid specified on the DFLTUSER system initialization parameter. It is used for terminal users who have not signed on. (See "CICS default user" on page 12.)
PLTPI user	The userid for PLTPI programs. It is specified on the PLTPIUSR system initialization parameter. The default is the region ID.
CICS terminal user who signs on	The userid specified by a terminal user during explicit sign-on. (See "Identifying CICS terminal users" on page 31.)
Preset terminal user	The userid specified on the terminal definition. (See "Preset terminal security" on page 5.)
ATI user	The userid operand specified within an intrapartition transient data queue definition, or EXEC CICS SET TDQUEUE ATIUSERID option.
Started transaction user	The userid for a started non-terminal transaction.
Link user	The userid used during MRO or ISC communication. (See "Link security" on page 90.)
Remote user	The userid for a transaction attached by the userid on a remote system. For example, by using transaction routing.
Surrogate user	The userid specified for a user who has the authority to start work on behalf of another user and is authorized to act for that user. (See Chapter 7, "Surrogate user security" on page 59.)

CICS segment

The CICS segment of the ESM user profile contains data for CICS users. For information on the order in which CICS searches for the operator information, see "Obtaining CICS-related data for a user" on page 38.

CICS user data

The information you may be able to specify in the CICS segment is as follows:

OPCLASS

CICS uses the operator classes when routing basic mapping support (BMS) messages initiated within a CICS transaction. The operator classes are numeric values in the range 1–24.

Specify operator classes for users who use CICS transactions that issue EXEC CICS ROUTE commands with the (optional) OPCLASS parameter. For automatic routing to occur, you specify the corresponding value as an operator class in the CICS segment of the user profile.

See the *CICS Application Programming Guide* for information about BMS and the use of the OPCLASS parameter for routing messages.

The default value for OPCLASS is 1. (See “Obtaining CICS-related data for the default user” on page 38.)

OPIDENT
The 1- to 3-character operator identification code that you assign to each operator.

CICS stores the code in the operator’s terminal entry in the CICS terminal control table (TCTTE) when the operator signs on. This operator ID is displayed in certain CICS messages and can also be used in the EXEC CICS ROUTE command for routing BMS messages. (For more information about BMS, see the *CICS Application Programming Guide*). It is also used when using the CEDA LOCK function, as described in the *CICS Resource Definition Guide*.

The default value for OPIDENT is blank. (See “Obtaining CICS-related data for the default user” on page 38.)

OPPRTY
The operator priority value—a decimal number that you want CICS to use when determining the task priority for CICS transactions that the operator invokes at a CICS terminal. The priority value can be in the range 0 through 255, where 255 is the highest priority.

CICS uses the sum of operator priority, terminal priority, and transaction priority to determine the dispatching priority of a transaction.

The default value for OPPRTY is 0. (See “Obtaining CICS-related data for the default user” on page 38.)

TIMEOUT
The time that must elapse since the user last used the terminal before CICS “times-out” the terminal.

The time must be a decimal integer in the range 0 through 9959 (the last two digits represent a number of minutes, and must be 00 through 59. Any digits to the left of these represent hours).

The value of 0 (the default) means that the terminal is **not** timed out (see “Obtaining CICS-related data for the default user” on page 38).

XRFSOFF
The CICS extended recovery facility (XRF) sign-off option. You specify this to indicate whether or not you want CICS to sign off the operator following an XRF takeover.

FORCE
Specify FORCE if you want CICS to sign off the operator automatically in the event of an XRF takeover.

NOFORCE
Specify NOFORCE if you want CICS to leave an operator signed on in the event of an XRF takeover.

The default value for XRFSOFF is NOFORCE. (See “Obtaining CICS-related data for the default user” on page 38.)

Defining XRFSOFF

The XRFSOFF function is also available at the TYPETERM definition level, as described in the *CICS Resource Definition Guide*, and at the CICS system level in the form of a system initialization parameter, as described in the *CICS System Definition Guide*. (As for the CICS segment, the default value for XRFSOFF in the system initialization parameters and in the TYPETERM definition is NOFORCE.)

Note that the FORCE option in the system initialization table or the TYPETERM definition overrides NOFORCE in the CICS segment.

Table 3 shows how specifying FORCE or NOFORCE in the system initialization parameters, on the TYPETERM definition (or the terminal control table (TCT)), and in the CICS segment together determine whether a terminal remains signed on after an XRF takeover.

As Table 3 shows, for a terminal to remain signed-on after an XRF takeover, NOFORCE must be specified in all three locations.

Table 3. Effects of FORCE and NOFORCE options

TYPETERM definition	CICS segment	System initialization parameter	
		FORCE	NOFORCE
FORCE	FORCE	Signed-off	Signed-off
	NOFORCE	Signed-off	Signed-off
NOFORCE	FORCE	Signed-off	Signed-off
	NOFORCE	Signed-off	**Signed-on**

Note: If takeover has exceeded the time specified by the XRFSTME system initialization parameter, users at terminals that have a nonzero TIMEOUT value do not remain signed-on after takeover.

CICS default user

When you are using CICS with external security, CICS assigns the security attributes of the CICS **default user** to all CICS terminal users who do not sign on. CICS also assigns the operator data from the CICS segment of the default user to signed-on users who do not have their own CICS segment data. To enable CICS to assign default security attributes and operator data, you define a CICS default userid to the ESM. You then tell CICS which default user to use by specifying the DFLTUSER system initialization parameter. (See the *CICS System Definition Guide* for information about this parameter.) If you do not specify a default userid on the DFLTUSER parameter, CICS uses the name “CICSUSER.”

Whether you use installation-defined operator data on your DFLTUSER parameter, or use the default, it is essential that the userid is defined to the ESM and that the region userid has installed surrogate security to use the default user (see “Surrogate user security” on page 6).

CICS “signs on” the default user during system initialization. **If you specify SEC=YES as a system initialization parameter, and CICS cannot “sign on” the default userid, CICS initialization fails**.

CICS uses the security attributes of the default userid to perform all the security checks for terminal users who do not explicitly sign on. These security checks

include **resource** and **command** security checking, in addition to **transaction-attach** security checking.

LANGUAGE segment

The language segment holds information about the national language in which the user receives messages. You may be able to specify two languages, but CICS assigns each user only one language. It assigns the primary language if it is specified and CICS supports that language. If the primary language is not specified or is not supported, CICS assigns the secondary language if it is specified and CICS supports it. If no language preference is specified in the LANGUAGE segment, the default national language for the CICS system is used, as specified on the NATLANG system initialization parameter.

Specify the user's preferred national languages in the LANGUAGE segment of the ESM user profile.

The available languages are English, Kanji, German, and traditional Chinese.

IBM-supplied resource class names for CICS

There is an IBM-supplied set of default resource names for use by CICS. You can also use resource classes defined by your installation.

Table 4. IBM-supplied resource class names for CICS

Default class name	Description	Class
TCICSTRN	CICS transactions, normal attach security	Member
GCICSTRN	CICS transaction groups	Group
ACICSPCT	CICS-started transactions and the following EXEC CICS commands: COLLECT STATISTICS, TRANSACTION, CREATE TRANSACTION, DISCARD TRANSACTION, and INQUIRE|SET TRANSACTION	Member
BCICSPCT	Groups for the above	Group
DCICSDCT	CICS transient data queues	Member
ECICSDCT	Groups for the above	Group
FCICSFCT	CICS files	Member
HCICSFCT	CICS file groups	Group
JCICSJCT	CICS journals	Member
KCICSJCT	CICS journal groups	Group
MCICSPPT	CICS programs	Member
NCICSPPT	CICS program groups	Group
SCICSTST	CICS temporary storage queues	Member
UCICSTST	CICS temporary storage queue groups	Group
CCICSCMD	EXEC CICS system commands, EXEC CICS FEPI system commands	Member
VCICSCMD	EXEC CICS system command groups and EXEC CICS FEPI system command groups	Group

Note: Each default class name has been allocated a group or class category according to its initial character.

Other IBM-supplied resource class names affecting CICS

The following other IBM-supplied resource class names affect CICS:

APPCLU The resource class in which you define profiles for verifying the identity of APPC partner logical units (LU6.2) during VTAM session establishment.

APPL The resource class in which you define profiles for controlling terminal users' access to VTAM applications, such as CICS.

FACILITY The resource class that includes profile definitions for controlling:

- MRO bindtime security
- Shared data tables security

SURROGAT The resource class that includes profiles for the following userids:

- preset
- default
- non-terminal
- PLTPI

It is also used for transactions started without a terminal, and for controlling job submission.

TERMINAL The resource class used to define profiles for terminals.

Unlike the IBM-supplied resource classes provided for CICS, you cannot change the class names of these general resource classes. Two of them have CICS system initialization parameters—XAPPC for APPCLU and XUSER for SURROGAT profiles.

Part 2. Implementing protection for a single-region CICS

This part discusses how to implement security on a single-region CICS, regardless of where the task needs to be performed—either in the CICS environment or in the external security environment. Where necessary, it refers you to other manuals in the CICS and for more detailed information about resource and security-related definitions.

- **Chapter 3, "CICS data set and system security" on page 19** deals with protecting the data sets that CICS requires—the VSE/ESA sublibraries and the CICS system data sets (such as the local and global catalogs, journal, auxiliary temporary storage, and transient data intrapartition data sets).
- **Chapter 4, "Verifying CICS users" on page 31** deals with all aspects of sign-on security, including the part played by the CICS segment.
- **Chapter 5, "Transaction security" on page 41** describes the security checks that CICS performs to verify that a user entering a transaction at a CICS terminal is authorized to use the transaction. This is known as **transaction-attach security**. It also explains the part played by the CICS **default userid**.
- **Chapter 6, "Resource security" on page 45** describes the RESSEC and CMDSEC attributes on resource definitions. It explains the purposes of the ESM resource class definitions, and gives examples illustrating how CICS and the ESM together control access to resources.
- **Chapter 7, "Surrogate user security" on page 59** describes the surrogate user checking activity that CICS can perform. It describes the ESM definitions needed, and gives some examples using the ESM surrogate user facility.
- **Chapter 8, "CICS command security" on page 65** describes CICS command security for the system programming commands. You can use these commands either through the CEMT master terminal transaction, or through the CICS API. This chapter also discusses the CMDSEC attribute on resource definitions.
- **Chapter 9, "Security checking using the QUERY SECURITY command" on page 73** describes security checking by the user application using the EXEC CICS QUERY SECURITY command, which enables an application program to request from the ESM the level of access a user has to a particular resource. The application program can determine what action to take based on the CICS-value data area (CVDA) values that CICS returns.
- **Chapter 10, "Security for CICS-supplied transactions" on page 81** describes how to protect the CICS-supplied transactions, both those that are for CICS internal use only (and cannot be invoked directly from a CICS terminal), and those provided explicitly for users at CICS terminals.

Chapter 3. CICS data set and system security

This chapter describes how to protect the data sets that CICS requires. It discusses the following:

- "Specifying the CICS region userid"
- "Authorizing access to CICS data sets" on page 22
- "Authorizing access to the CICS region" on page 24
- "Defining security-related system initialization parameters" on page 24

Specifying the CICS region userid

When you start a CICS region in a VSE/ESA environment that has an ESM installed, the job or task is associated with a userid, referred to as the **CICS region userid**. The authority associated with this userid determines which ESM-protected resources the CICS region can access.

Each CICS region, for either production or test use, should be subject to data set protection based on the region userid under which the CICS region executes. You can specify the region userid under which CICS executes:

- On the USER parameter of the ID statement immediately following the JOB statement when you start CICS as a JOB.
- On the SEC parameter on the JECL JOB statement if VSE/ESA batch security is active.

To ensure the authorizations for different CICS regions, are properly differentiated, run each with a unique region userid. For example, the userid under which you run the production CICS regions to process payroll and personnel applications should be the only CICS userid authorized to access production payroll and personnel data sets.

If you are using intercommunication, it is particularly important to use unique userids, unless you want to bypass link security checking by using equivalent systems. For more information, see "Link security with LU6.2" on page 99, "Link security with LU6.1" on page 137, or "Link security with MRO" on page 145, depending on

Defining user profiles for CICS region userids

Before bringing up a CICS region, ensure that the required userids are defined - the CICS region userid and the CICS default userid. (The default user ID is specified on the system initialization parameter, DFLTUSER).

Coding the USER parameter on the CICS JOB statement

If you start CICS from a job, include the parameters USER= and PWD= on the ID statement immediately after the JOB statement. For example:

```
// JOB jobname
// ID USER=userid,PWD=password
```

To change a password use the NEWPW parameter. For example:

```
// ID USER=userid,PWD=password,NEWPW=new password
```

Coding the SEC parameter on the POWER JECL statement

If you start CICS from a job with POWER JECL, and VSE/ESA batch security is active, (SEC=YES in the IPL procedure), include the SEC= parameter on the POWER JOB statement. For example,

```
* $$ JOB JNM=jobname,CLASS=A,DISP=D,SEC=(userid,password)
```

Authorities required for CICS region userids

The CICS control program runs under the CICS region userid. Therefore, this userid needs access to all the resources that CICS itself needs to use. There are two types of these resources:

1. Resources external to CICS, such as disk files, VSE libraries, the spool system, and the VTAM network.
2. Resources internal to CICS, such as system transactions and auxiliary userids.

Authorizing external resources: Like a batch job, each CICS region must be able to access many external resources. The authority for CICS to access these resources is obtained from the CICS region userid. It doesn't matter which signed-on user requests CICS to perform the actions that access these resources. The external services are aware only that CICS is requesting them, under the region userid's authority.

Give access to these resources:

- External disk data sets used by CICS

  CICS needs authority to open all the disk data sets that it uses. See "Authorizing access to CICS data sets" on page 22.

- External disk data sets used by application programs

  CICS needs authority to open all the disk data sets that your own application programs need. See "Authorizing access to user data sets" on page 23.

- VSE sublibraries used by CICS

  CICS needs authority to access members from VSE sub-libraries. For more information, see "Authorizing access to VSE libraries" on page 23.

Authorizing internal resources: There are several internal functions in which CICS behaves like an application program, but is actually performing housekeeping functions that are not directly for any end user. The associated transactions execute under control of the CICS region userid, and because they access CICS internal resources, you must give the CICS region userid authority to access them. These are:

- CICS system transactions

  CICS needs authority to attach all the internal housekeeping transactions that it uses. See "Category 1 transactions" on page 81.

- Auxiliary userids

  If CICS surrogate user checking is specified with the XUSER system initialization parameter, CICS needs authority to use certain additional userids. These are:

  – The default userid

    See “CICS default user” on page 59.

  – The userid used for post-initialization processing (PLTPIUSR)

    See “Post-initialization processing” on page 59.

  – The userid used for transient data trigger transactions

    See “Transient data trigger-level transactions” on page 61.

  – Resources used by PLTPI programs

    If the PLTPIUSR system initialization parameter is omitted, the CICS region userid is used for all PLTPI programs. In this case, give the CICS region userid access to all the CICS resources that these programs use. See “PLT programs” on page 43.

Defining the default CICS userid to the ESM

For each CICS region for which you specify SEC=YES, define an ESM user profile whose userid matches the value of the system initialization parameter, DFLTUSER. For example, if you specify DFLTUSER=NOTSIGND, define an ESM user profile named NOTSIGND.

If you do not specify a value for the DFLTUSER parameter, the CICS-supplied default userid is CICSUSER—define an ESM user profile named CICSUSER.

Define a different default CICS userid for each CICS region if any of the following considerations applies:

- The default CICS userid requires different security attributes.
- The default CICS userid requires different operator data (CICS segment of the ESM user profile).
- The default CICS userid requires a different default language (LANGUAGE segment of the ESM user profile).

If you have specified the system initialization parameter XUSER=YES, authorize the CICS region userid to be a surrogate user of the default userid.

During startup, CICS “signs on” the default userid. If the default user sign-on fails (because, for example, the userid is not defined to the ESM), CICS issues message DFHXS1104 and terminates CICS initialization.

When CICS successfully signs on a valid ESM userid as the default user, it establishes the terminal user data for the default user from one of the following sources:

- The CICS segment of the default user’s ESM user profile
- Built-in CICS system default values

See “Obtaining CICS-related data for the default user” on page 38 for details of the sign-on process for obtaining CICS terminal operator data.

CICS assigns the security attributes of the default userid to all CICS terminals before any terminal user begins to sign on. The security attributes and terminal user data of the default user also apply to any terminals at which users do not sign on (using either the CICS-supplied CESN transaction or a user-written equivalent based on the EXEC CICS SIGNON command), unless the security has been explicitly preset by specifying a value for the USERID option in the terminal definition.

CICS also assigns the security attributes of the default userid to any "trigger level transactions" that are initiated for transient data queues without a USERID parameter.

Ensure the default userid gives at least the minimum authorities that ought to be granted to any other terminal user. In particular:

- Give the default user access to the region's APPLID. See "Authorizing access to the CICS region" on page 24.
- Give the default user access to the CICS-supplied transactions that are intended to be used by everybody. (In particular, the "good morning" and "good night" transactions, as specified on system initialization parameters GMTRAN and GNTRAN, if these are specified for your installation and the signon transaction CESN).

Authorizing access to CICS data sets

When you have defined a region userid for your CICS job (or started task), permit that userid to access the CICS system data sets with the necessary authorization.

When authorizing access to CICS system data sets, choose appropriately from the UPDATE and CONTROL levels of access. You should make certain that these data sets are protected to ensure that only CICS region userids can access those data sets. For information about the CICS region userid, see "Specifying the CICS region userid" on page 19.

CICS needs CONTROL access for the transient data intrapartition, temporary storage, and CICS availability manager (CAVM) data sets.

The CICS filenames for the data sets in this category are as follows:

DFHNTRA Transient data intrapartition data set
DFHTEMP Temporary storage data set
DFHXCTL XRF control data set
DFHXMSG XRF message data set

CICS needs UPDATE access to the remainder of all the data sets covered by these profiles. The CICS filenames for the data sets in this category are as follows:

DFHGCD Global catalog data set
DFHLCD Local catalog data set
DFHAUXT Auxiliary trace data set, A extent
DFHBUXT Auxiliary trace data set, B extent
DFHDMPA Transaction dump data set, A extent
DFHDMPB Transaction dump data set, B extent

Note: The auxiliary trace data set, and the transaction dump data set may contain sensitive information. Protect them from unauthorized access.

Therefore, for CICS system data sets you need at least two generic profiles to restrict access to the appropriate level. See Table 5 on page 23.

Table 5. Summary of generic data set profiles

Required access level	Type of CICS data sets protected
UPDATE	Auxiliary trace; transaction dump; system definition; global catalog; local catalog; and restart
CONTROL	Temporary storage; intrapartition transient data; XRF message; and XRF control

If you use generic naming of the data set profiles, you can considerably reduce the number of profiles you need for your CICS regions.

By establishing a naming convention for the data sets belonging to each region, and one generic profile for each CICS region, with the CICS VTAM APPLID as one of the data set qualifiers, you can ensure that only one CICS region has access to the data sets.

The CICS system definition data set (CSD) is protected by a discrete profile to which all CICS groups have access. This assumes that all the CICS regions are sharing a common CSD. If your CICS regions do not share a common CSD and each region has its own CSD, or if groups of regions share a CSD, define discrete or generic data set profiles as appropriate.

Authorizing access to user data sets

When you have defined the ESM userids for your CICS regions and given them access to the CICS system data sets, permit the userids to access the CICS **application** data sets with the necessary authority.

Authorizing access to VSE libraries

VSE library security is related to data set security because all libraries are associated with a data set. In addition to the authorization checks performed against the full data set name, the ESM may perform checking against the 7-character library and sub-library names, as well as individual library members.

The CICS region userid should be permitted access to the VSE sub-libraries, referenced within the CICS startup JCL, with the appropriate authorization. For example:

- READ access to all sub-libraries specified in any LIBDEF SEARCH statements.
- UPDATE or higher access to the system dump library as specified on the LIBDEF DUMP,CATALOG= statement.

Authorizing access to the CICS region

You can restrict access by terminal users to specific CICS regions using the ESM APPL resource class. Where the application name is the VTAM applid, as specified in the system initialization parameter APPLID= , or the generic applid in the case of an XRF system.

Notes:

1. CICS always passes the APPLID to the ESM when requesting the ESM to perform user sign-on checks, and there is no mechanism within CICS to prevent this.
2. For MRO only, the APPLID is propagated from the terminal-owning region (TOR) to the other region that the user accesses. Therefore, you can force users to sign on through a TOR, by denying users access to any APPLID except that of the TOR.

Authorizing the CICS region userid as surrogate user

When CICS performs surrogate user checking, the CICS region userid must be authorized as a surrogate. Grant authorization for the CICS region userid acting as a surrogate user for the following:

- The CICS default user
- The userid used for post-initialization processing (PLTPIUSR)
- All userids used for transient data trigger level transactions

For more information about surrogate user checking, see Chapter 7, “Surrogate user security” on page 59.

Defining security-related system initialization parameters

There are several system initialization parameters that CICS provides for specifying your security requirements at the system level. These parameters are coded in the CICS system initialization table (SIT) or as system initialization overrides. For full details of system initialization parameters, see the *CICS System Definition Guide*.

SEC

You use the SEC system initialization parameter to specify the level of resource security management you want for your CICS region. There are two options:

YES
This means that the CICS external security interface will be initialized, and control of CICS security is determined by the other security-related system initialization parameters:

```
SECPRFX              XRFSOFF
DFLTUSER             XRFSTME
ESMEXITS             XCMD
SNSCOPE              XDCT
CMDSEC               XFCT
RESSEC               XJCT
PLTPIUSR             XPCT
PLTPISEC             XPPT
XAPPC                XTRAN
XUSER                XTST
```

NO
This means that there is no security checking whether users are allowed to access CICS (and non-CICS) resources from this region, and sign-on cannot take place.

Note: Even if you have specified SEC=NO, with MRO bind-time security, the CICS region userid is sent to the secondary system, and bind-time checking is carried out in the secondary system. See “Bind-time security with MRO” on page 143 for more information.

SECPRFX

This parameter is effective only if you also specify SEC=YES. You use the SECPRFX system initialization parameter to specify whether you want CICS to prefix the resource names that it passes to the ESM for authorization. The prefix that CICS uses is the ESM userid under which the CICS region is running.

Prefixing is useful mainly when you have more than one CICS region. It enables you to prevent users on one CICS region from accessing the resources of a different CICS region that has a different prefix. For example, you might have one CICS region with the prefix CICSPROD and another with prefix CICSTEST. Users of the CICSTEST system would be able to use profiles that included the CICSTEST prefix, and users of the CICSPROD system would be able to use profiles that included the CICSPROD prefix. Users of both systems would be able to use resources protected by profiles that included CICS.

There are two options on the SECPRFX parameter:

NO
CICS does not prefix the resource names in authorization requests that it passes to the ESM from this CICS region.

YES
CICS prefixes the resource names with its ESM userid when passing authorization requests to the ESM. The prefix corresponds to the CICS region userid.

CMDSEC

Code CMDSEC to specify whether or not you want CICS to honor the CMDSEC option specified on a transaction's resource definition. CMDSEC specified with the option ASIS means that CICS obeys the CMDSEC option. CMDSEC specified with the option ALWAYS means that CICS ignores the CMDSEC option, and always performs the command check.

DFLTUSER

Specify a value for DFLTUSER to identify to CICS the name you have defined to the ESM as the default userid. If you omit this parameter, the name defaults to CICSUSER. See "Defining the default CICS userid to the ESM" on page 21.

ESMEXITS

Use ESMEXITS to specify whether you want CICS to pass installation data for use by the ESM installation exits. For more information on ESMEXITS, see Chapter 18, "Customizing security processing" on page 167.

PLTPISEC

Code PLTPISEC to specify whether or not you want CICS to perform command security or resource security checking for PLT programs that run during CICS initialization.

PLTPIUSR

Code PLTPIUSR to specify the userid that CICS is to use for security checking for PLT programs that run during CICS initialization.

RESSEC

Code this to specify whether or not you want CICS to honor the RESSEC option specified on a transaction's resource definition. RESSEC specified with the option ASIS means that CICS obeys the RESSEC option. RESSEC specified with the option ALWAYS means that CICS ignores the RESSEC option, and always performs the resource check. For more information about these options, see the *CICS System Definition Guide*.

SNSCOPE

SNSCOPE—the sign-on SCOPE—applies to all userids signing on by explicit sign-on request; for example, the EXEC CICS SIGNON command or the CESN transaction. Use it to specify whether or not a userid can have more than one CICS session active at the same time.

CICS resource class system initialization parameters

You specify at the system level (with the SEC=YES parameter) that you want CICS to use the ESM to authorize access to CICS resources. You also specify at the system level which particular CICS resources you want CICS to check by means of the X*name* system initialization parameters. The full list of the CICS resource classes is shown in Table 6 on page 27, each with corresponding X*name* system initialization parameter.

Table 6. System initialization parameters for the CICS resource classes	
System initialization parameter	**Resource**
XAPPC={**NO**|YES}	APPC partner-LU verification
XCMD={**NO**|name|YES}	EXEC CICS system commands EXEC CICS FEPI system commands
XDCT={**NO**|name|YES}	Transient data destinations
XFCT={**NO**|name|YES}	Files
XJCT={**NO**|name|YES}	Journals and logs
XPCT={**NO**|name|YES}	Started transactions and EXEC CICS commands: COLLECT STATISTICS TRANSACTION CREATE TRANSACTION DISCARD TRANSACTION INQUIRE TRANSACTION and SET TRANSACTION
XPPT={**NO**|name|YES}	Programs
XTRAN={**YES**|name|NO}	Attached transactions
XTST={**NO**|name|YES}	Temporary storage entries
XUSER={**NO**|YES}	Surrogate user checking

Notes:

1. The parameters are effective only with SEC=YES.
2. None of the parameters can be entered as a console override.

If you specify YES for any X*name* system initialization parameter, CICS uses the default class name for that parameter. (See “IBM-supplied resource class names for CICS” on page 13.)

As examples, the effect of specifying SEC=YES with three of the resource class parameters specified as X*name*=YES are illustrated in the following Table 7 on page 28, and Table 8 on page 28

Table 7. Specifying external security with default resource classes	
System initialization parameter	**Effect**
SEC=YES	CICS initializes external security interface.
XTRAN=YES	CICS uses the TCICSTRN and GCICSTRN resource class profiles for transaction-attach security checking.
XFCT=YES	CICS uses the FCICSFCT and HCICSFCT resource class profiles for file access security checking.

As a second example, the effect of specifying SEC=YES with the same three associated resource class parameters specified as X*name*=*username* is shown in Table 8.

Table 8. Specifying external security for user-defined resource classes	
System initialization parameter	**Effect**
SEC=YES	CICS initializes external security interface.
XTRAN=$usrtrn	CICS uses the T$usrtrn and G$usrtrn user-defined resource class profiles for transaction-attach security checking.
XFCT=$usrfct	CICS uses the F$usrfct and H$usrfct user-defined resource class profiles for file access security checking.

When CICS is being initialized, it requests the ESM to bring resource profiles into main storage to match all the resource classes that you specify on system initialization parameters. Note that (except for XTRAN) X*name*=NO is the default in the system initialization parameters. You must supply ESM profiles for all those resources for which you specify X*name*=YES. If CICS requests the ESM to load a general resource class that does not exist or is not correctly defined, CICS issues a message indicating that external security initialization has failed, and terminates CICS initialization.

For guidance on the syntax of external security system initialization parameters, see the *CICS System Definition Guide*.

The way you define the individual transaction definitions in the CSD determines whether you want to use external security for the resources and commands used with transactions. See Chapter 4, “Verifying CICS users” on page 31 and Chapter 5, “Transaction security” on page 41 for information about specifying resource and command security for transactions.

XAPPC and XUSER

The syntax of the XAPPC and XUSER system initialization parameters is slightly different from that of the other X*name* parameters. You can only specify YES or NO.

XAPPC=YES indicates that you want session security for APPC sessions. If XAPPC=YES is specified and the APPCLU class is not activated in the ESM, CICS fails to initialize. For more information on what happens in these circumstances, see “CICS initialization failures related to security” on page 182.

XAPPC enables LU6.2 bind-time (also known as APPC) security. For more information, see “Bind-time security with LU6.2” on page 95.

XUSER activates surrogate user security. For more information, see Chapter 7, “Surrogate user security” on page 59. If XUSER=YES is specified and the SURROGAT class is not activated in the ESM, CICS fails to initialize.

Chapter 4. Verifying CICS users

This chapter covers all aspects of CICS sign-on security, including the use of the ESM CICS segment. It discusses the following:

- "Identifying CICS terminal users"
- "Sign-on process"
- "Sign-off process" on page 33
- "Auditing sign-on and sign-off activity" on page 34
- "Controlling access to CICS from specific ports of entry" on page 35
- "Preset terminal security" on page 35
- "Using a VSE system console as a CICS terminal" on page 38
- "Obtaining CICS-related data for the default user" on page 38
- "National language and non-terminal transactions" on page 39

Identifying CICS terminal users

If you are running CICS with external security checking, you control users' access to CICS resources through levels of authorization you define in ESM-managed resource profiles. You define these authorizations for specific users by adding individual ESM userids to the resource access lists; or, for unsigned-on users, by adding the default CICS userid to selected resource access lists.

All CICS terminal-user data is defined in the ESM CICS segment. See "Obtaining CICS-related data for a user" on page 38 for more information about CICS terminal-user data, and how CICS obtains it.

Sign-on process

When users log-on to CICS through VTAM, but do not sign on, they can use only those transactions that the CICS default user is permitted to use. As these are likely to be strictly limited, users must sign on to obtain authorization to run the transactions that they are permitted to use.

Explicit sign-on

Users can explicitly sign on either by using the CICS-supplied transaction, CESN, which can be defined as the "good morning" transaction on the GMTRAN system initialization parameter; or by using an installation-provided sign on transaction which uses the EXEC CICS SIGNON command. OIDCARD users can use CESN to sign on if the card reader supports the DFHOPID identifier (AID). If it does not, use your own installation-provided sign-on transaction. For information about CESN, see the *CICS-Supplied Transactions* manual. For programming information about EXEC CICS SIGNON, see the *CICS Application Programming Reference* manual.

When a user signs on to CICS, the sign-on process involves the following **phases**:

Scoping
After the sign-on panel is completed and sent, CICS verifies that the entered userid does not match a userid already signed on within the scope defined for the CICS system by the SNSCOPE system initialization parameter.

Identification
CICS calls the ESM with the supplied userid to confirm that a profile has been defined for the user.

Verification
CICS passes information to the ESM to verify that the user is genuine. This is either a password or an OIDCARD or both. If the password entered has expired, CICS prompts the user for a new password. When the new password conforms to the ESM password formatting rules for an installation, the new password and the date-of-change are recorded in the ESM user profile.

Immediately following the request to the ESM for userid and password verification, CICS clears the internal password field. This minimizes the possibility of the password being revealed in any dump of the CICS address space that may be taken.

You may also voluntarily change your password by entering a new value.

```
                Sign-on for CICS                    APPLID CICSA100

. . . . . . This is where the good morning message appears. . . . . .
. . . . . . It can be up to four lines in depth . . . . . .
. . . . . . to contain the maximum message length . . . . . .
. . . . . . of 246 characters . . . . . .


Type your userid and password, then press ENTER:

         Userid . . . . ________

         Password . . . ________

         Groupid . . . ________

         Language . . . __

     New Password . . . ________




DFHCE3520 Please type your userid.
F3=Exit
```

Figure 1. The CICS sign-on panel

Authorization
The ESM performs checks on the CICS system name and the port of entry to verify that the user is allowed to use the CICS system. In the CICS system name check, the ESM determines whether the user is authorized to access this particular CICS system. (See “Authorizing access to the CICS region” on page 24).

With the port of entry check, the ESM verifies that the user is authorized to sign on using that port of entry. The use of defined terminals can be restricted to certain times of the day, and to certain days of the week. See “Controlling access to CICS from specific ports of entry” on page 35.

These checks restrict the user to signing on only to those CICS regions for which they are authorized, and only from terminals they are authorized to use.

Explicit sign-on, reached through CESN or EXEC CICS SIGNON, is performed by the user at the port of entry.

Table 9. Explicit and implicit signons

Phase	Explicit	Implicit
Scoping	Yes	No
Identification	Yes	Yes
Verification	Yes	No except with ATTACHSEC(IDENTIFY)
Authorization	Yes	Yes

User attributes
CICS obtains CICS user attributes from the CICS and LANGUAGE segments of the ESM database.

Sign-off process

The sign-off process dissociates a user from a terminal where the user had been previously signed on. The user can explicitly sign off using the CESF transaction or an installation-provided transaction that uses the EXEC CICS SIGNOFF command. If the attributes of the signed-on user include a non-zero value for TIMEOUT, an implicit sign-off occurs if this interval expires after a transaction terminates at this terminal.

When the time-out period expires, if the system initialization parameter GNTRAN=NO is specified (or allowed to default), CICS performs an immediate signoff. If GNTRAN specifies a transaction-id to be scheduled and that transaction performs a signoff, the action CICS takes depends on the SIGNOFF option specified in the terminal’s RDO TYPETERM resource definition.

An exceptional case is that the goodnight transaction is not used for the user of a CRTE session. A surrogate user whose time expires is signed off, losing the security capabilities the terminal previously had. Message DFHSN1200 is sent to the CSCS log, and indicates what has happened.

The possible signoff options in the TYPETERM resource definition and the associated actions are as follows:

SIGNOFF(YES)
CICS signs off the operator from CICS, but the terminal remains connected.

SIGNOFF(LOGOFF)
CICS signs off the operator from CICS **and** logs off the terminal from VTAM.

In addition, if the terminal is autoinstalled, the delay period specified by the AILDELAY system initialization parameter commences, and if the delay period

expires before the terminal attempts to log on again, CICS deletes the terminal entry (TCTTE) from the TCT. For information about CICS autoinstall, see the *CICS Resource Definition Guide*.

SIGNOFF(NO)
CICS leaves the user signed on and the terminal remains logged on, effectively overriding the time-out period.

Explicit sign-off

Explicit sign-off removes the user's scoping. The user must be explicitly signed on before signing off with CESF or EXEC CICS SIGNOFF. The user is returned to the default level of security.

Note: CESN will not sign the user off until a valid attempt has been made to use the panel, even if the sign-on attempt subsequently fails. It is not recommended that CESN be used for the Goodnight transaction.

Implicit sign-on and implicit sign-off

Implicit sign-on means that all other userids added to the system by CICS are considered to be implicitly signed on without a password.

A user is implicitly signed off if the transaction suffers a TERMERR condition while attempting to send data to its principal facility. However, the user is not subject to USRDELAY but is signed off immediately. If SNSCOPE is in use, the scope will be released at the time of sign off. If the transaction handles the ABEND, it continues running as a non-terminal task with the authority of the starting user.

Auditing sign-on and sign-off activity

You may wish to use the auditing/tracking function available from the ESM to record sign-on and sign-off activity. You can only properly interpret the logging of unsuccessful sign-on attempts by also recording successful sign-ons. For example, if a user makes one or two unsuccessful attempts followed immediately by a successful sign-on, the unsuccessful sign-ons can be interpreted as being caused by keying errors at the terminal. However, several unsuccessful attempts for a variety of userids occurring within a short space of time, and without any subsequent successful sign-on activity being recorded, may well be cause for a security concern that warrants investigation.

Recording the successful sign-on and sign-off activities establishes an audit trail of the access to particular systems by the terminal user population. This may also be useful for systems capacity planning.

CICS uses its CSCS transient data destination for security messages. Messages of interest to the security administrator for the CICS region are directed to this destination. In some instances, when security-related messages are directed to terminal users, corresponding messages are written to the CSCS transient data destination. In the case of the DFHCE3544 and DFHCE3545 messages that are sent to terminal users, for example, the corresponding messages DFHSN1118 and DFHSN1119 are sent to CSCS. The DFHSNxxxx messages include reason codes that indicate the precise nature of the invalid sign-on attempt.

Controlling access to CICS from specific ports of entry

During sign-on processing, CICS issues a request to the ESM to verify the user's password, and to check whether the user is allowed to access that terminal. This check is also performed for the userid specified for preset security terminal definitions.

Terminals are protected using the CICS-supplied ESM resource class, TERMINAL.

Preset terminal security

For some selected terminals, and VSE consoles when used as CICS terminals, consider using CICS preset terminal security as an alternative to terminal user security. A terminal becomes a preset security terminal when you specify the userid operand on the terminal definition.

CICS preset terminal security allows you to associate a userid permanently with a terminal that is defined to CICS. This means that CICS implicitly signs on the terminal when it is being installed, instead of a subsequent sign-on of that terminal by a user. Typically, you define preset security for devices without keyboards, such as printers, at which users cannot sign on.

You can also use this form of security on ordinary display terminals as an alternative to terminal user security. This permits anyone with physical access to a terminal with preset security to enter the transactions that are authorized for that terminal. The terminal remains signed on as long as it is installed, and no explicit sign-off can be performed against it. If the userid associated with a display terminal with preset-security has been authorized to use any sensitive transactions, ensure that the terminal is in a secure location to which access is restricted. Preset-security might be appropriate, for example, for the terminals physically located within a CICS network control center.

You can use preset-security to assign a userid with **lower** authority than the default, for terminals in unrestricted areas.

For example, to define a terminal with preset-security, use CICS (CEDA) commands as follows:

```
CEDA DEFINE TERMINAL(cics_termid)
            NETNAME(vtam_termid)
            USERID(userid)
            TYPETERM(cics_typeterm)
```

For further information on preset-security terminals in the transaction routing environment, refer to “Preset-security terminals and transaction routing” on page 108 (LU6.2 security) and “Preset-security terminals and transaction routing” on page 151 (MRO security).

Controlling the use of preset-security

When a preset-security terminal is installed, the specified userid is implicitly signed on at the terminal. Ensure that only a trusted person is allowed to define and install terminals with preset security, because the userid specified on the terminal may have access to CICS resources not available to the installer.

Surrogate user checking ensures that a user is authorized to act for another user. Surrogate user checking can be enforced when a user installs a terminal that is preset for a different userid, and is specified by the ESM SURROGAT resource class. The CICS *userid.*DFHINSTL resource can be defined in the SURROGAT resource class for authorization to install terminals that are preset for that specific userid.

When a terminal is installed with a preset userid, the surrogate user is the userid performing the installation. See Chapter 7, “Surrogate user security” on page 59 for more information.

The CEDA command checks the authority of the user to install preset terminals. Consider, therefore, whether to restrict the following functions with a view to controlling who can define and install terminals with preset security:

- The CEDA transaction
- The SURROGAT resource class
- The XUSER system initialization parameter
- Batch access to the CSD using the DFHCSDUP utility
- The LOCK command for locking CSD definitions

Note: When CICS installs a GRPLIST that contains preset terminal definitions, no checking is done at initialization time. However, you can still ensure that you control who can define and install terminals and sessions with preset security by using the CEDA LOCK command to control the contents of GRPLIST groups.

Restricting use of the CEDA transaction

If the CEDA transaction is enabled on your production CICS regions, restrict its use to authorized users. This gives you control over who can define resources, such as terminals, to CICS. See Chapter 10, “Security for CICS-supplied transactions” on page 81 for information about protecting CICS-supplied transactions.

Using the SURROGAT resource class

Also ensure that you restrict who can install terminals with preset security, so that even when such terminals are defined in the CSD, only authorized users can install them on CICS. (This authority is additional to the authority needed to run CEDA.) The user must already have authority to run the CEDA transaction.

Defining the XUSER system initialization parameter

To ensure that CICS can perform surrogate user security checks on the use of the CEDA INSTALL command for terminals with preset security, define the XUSER system initialization parameter. See “CICS resource class system initialization parameters” on page 26 for information about defining the XUSER parameter.

Restricting batch access to the CSD

You can also use the CSD batch utility program, DFHCSDUP, to define resources in the CSD. So that only authorized users are allowed to update the production CSDs, you should restrict access to the CSD data set profile in the ESM to the CICS region userids and other authorized users only. The INSTALL command is not available in DFHCSDUP.

Using the LOCK command

CICS also installs resource definitions in the CSD during an initial or cold start, from the list of groups defined on the GRPLIST system initialization parameter. To control the addition of resource groups to the CICS startup group list, you should use the CEDA or DFHCSDUP LOCK command to lock the list. This protects the group list from unauthorized additions. Also, lock all the groups that are specified in this list.

Note: The OPIDENT of the signed-on user is used as the key for the LOCK and UNLOCK commands. For information about LOCK and UNLOCK, see the *CICS Resource Definition Guide*.

Other preset security considerations

If you intend to use preset security, consider these additional topics:

- Autoinstall models
- Sessions with preset security
- Terminals defined in the TCT

Autoinstall models

If you are using autoinstall models with preset security, CICS makes the same authorization check as for ordinary terminals when the model is installed. It does not check authorization when the autoinstall model is used to perform the automatic installation. If an autoinstall model with preset security becomes invalid (for example, if the userid is revoked), any attempts to install a terminal with this model fail.

Sessions

A session becomes governed by preset security if you specify the userid operand on the session definition. The same checking is performed if you install preset security sessions.

Terminals defined in the terminal control table

For terminals defined in the terminal control table (TCT) (for example, sequential devices), the userid is also defined in the TCT, and, when CICS initializes, it signs on these terminals. If the sign-on fails (for example, if the userid is revoked), the terminal is put out of service. If the userid later becomes valid (for example, if it is resumed), setting the terminal in service results in a successful sign-on. CICS does not perform a surrogate user check for these terminals.

Using a VSE system console as a CICS terminal

If you intend to use a VSE system console as a CICS terminal, we also recommend that preset security be specified on the console's CICS terminal definition. If it is not, explicitly signon to get more authority than the default user. The password will usually be seen on the console and in the system log.

The format of the CESN command, when entered from a console, is as follows:

```
MSG partition,DATA=CESN[USERID=userid]
       [,PS=password]
       [,NEWPS=newpassword]
       [,GROUPID=groupid]
       [,LANGUAGE=language-code]
```

If any of the data entered on the CESN command is invalid, or if the password is missing or expired, CICS prompts the user to enter the missing or invalid data by issuing a system message that requires a response. When CICS prompts for a password, it uses a security routing code to ensure that the response is not recorded on the console or in the system hardcopy log. To terminate the sign-on process, enter a null reply to the message.

Obtaining CICS-related data for a user

CICS obtains CICS-related data from one of the following sources:

- The CICS and LANGUAGE segments of the ESM profile
- Built-in CICS system default values.

This section explains how the data is obtained, for the default user and terminal users signing on.

Obtaining CICS-related data for the default user

When implicitly signing on the CICS default user during initialization, CICS obtains attributes in the following way:

1. CICS calls the ESM to request user data for the CICS default user from the CICS segment and the LANGUAGE segment. If the CICS segment **or** the LANGUAGE segment data is present for the default userid, the ESM returns this data to CICS. See "CICS segment" on page 10 for details of the information that you can define in the CICS segment. See "LANGUAGE segment" on page 13 for details of the LANGUAGE segment.
2. If the ESM does not return the CICS segment or LANGUAGE segment data for the default userid, CICS assigns the following built-in system default values:

National language	Obtained from the first operand on the NATLANG system initialization parameter. This defaults to US English if not specified.
Operator class	One (OPCLASS=1)
Operator identification	Blank (OPIDENT=' ')
Operator priority	Zero (OPPRTY=0)
Timeout	Zero (TIMEOUT=0)
XRF signoff	Signoff not forced (XRFSOFF=NOFORCE)

Obtaining CICS-related data at signon

When handling an explicit sign-on for a CICS terminal user, CICS obtains the terminal user attributes in the following way:

1. CICS calls the ESM to request data about the CICS terminal user from the CICS segment and the LANGUAGE segment. If the CICS segment **or** the LANGUAGE segment data is present for the terminal user, the ESM returns this data to CICS. See "CICS segment" on page 10 for details of the information that you can define in the CICS segment. See "LANGUAGE segment" on page 13 for details of the LANGUAGE segment.
2. If the ESM does not return the CICS segment or LANGUAGE segment data for the user, CICS uses the user attributes of the CICS default user, defined during system initialization. (See "Obtaining CICS-related data for the default user" on page 38.)

CICS obtains the national language attribute in the following order:

1. The LANGUAGE option on the CICS-supplied CESN transaction, or the LANGUAGECODE or NATLANG option of the EXEC CICS SIGNON command, if supported by CICS. A **supported** national language is a **valid** national language that has been specified in the NATLANG system initialization parameter and has the corresponding message definitions. See the *CICS System Definition Guide* for more information about defining this parameter.
2. The *primary-language* parameter in the LANGUAGE segment of the user's ESM profile, if supported by CICS.
3. The *secondary-language* parameter in the LANGUAGE segment of the user's ESM profile, if supported by CICS.
4. The NATLANG parameter of the RDO TERMINAL resource definition.
5. The language established for the default user as described on page 38.

National language and non-terminal transactions

When a user specifies a national language during sign-on, the sign-on option overrides the language specified in the user's ESM CICS or LANGUAGE segment. The language thus specified is set for the time the user is signed on at the terminal. Any transaction invoked by the signed-on user runs with the national language specified on the sign-on.

However, if a transaction uses the EXEC CICS START command to start another transaction, the national language attribute for the started transaction is derived as follows:

1. If the USERID parameter is specified on the START command, the national language is taken from the ESM CICS or LANGUAGE segment of the specified userid.
2. If the user is signed on at a terminal with a preset national language specified on the terminal definition, this preset national language is assigned to the started transaction.
3. If there is no userid on the START command, and no preset national language on the terminal, the started transaction inherits the national language specified

in the ESM CICS or LANGUAGE segment of the signed-on user (not the national language used in the sign-on).

If the national language of the original terminal is required, the terminal's national language can be inquired about before the EXEC CICS START command is issued. The information can then be passed as data in the START command for the use of the transaction that has been started.

Chapter 5. Transaction security

CICS can apply two levels of security to a transaction. The first is security checking on the transaction itself, sometimes referred to as **attach-time**, or **transaction-attach security**. This chapter discusses transaction-attach security—the security checks that CICS performs to verify that a terminal user is authorized for the transaction to be run at the user's terminal.

Transaction-attach security applies to transactions that a user enters directly at a terminal, and also to transactions started from another CICS transaction.

The other level of security you can use for CICS transactions applies to the resources used by the transactions: files, databases, and CICS commands. For more information, see Chapter 6, "Resource security" on page 45.

This chapter discusses transaction-attach security under the following main headings:

- "CICS parameters controlling transaction-attach security"
- "Defining transaction profiles to the ESM" on page 42
- "Transactions not associated with a terminal" on page 43

CICS parameters controlling transaction-attach security

You control CICS transaction-attach security checking through CICS system initialization parameters. These are:

SEC Specify SEC=YES if you want to use the ESM services to control access to any CICS resources—in particular, CICS transactions. (For more information, see "SEC" on page 24.)

SECPRFX Specify SECPRFX=YES if your transaction profiles are defined to the ESM with a prefix that corresponds to the userid of the CICS region. (For more information, see "SECPRFX" on page 25.)

XTRAN Specify XTRAN=YES or XTRAN=*resource_class_name* if you want CICS to control who can initiate transactions. If you specify YES, CICS uses profiles defined in the ESM default resource classes TCICSTRN and GCICSTRN. (See "IBM-supplied resource class names for CICS" on page 13 for details of these resource classes.)

If you specify XTRAN=NO, CICS does not perform any authorization check on users initiating transactions.

Note that the default is YES. Therefore if you specify SEC=YES and omit the XTRAN parameter, transaction-attach security is in effect, using the default resource class names.

There are no CICS parameters that allow you to control transaction-attach security at the individual transaction level. When you specify SEC=YES and XTRAN=YES (or XTRAN=resource_class_name), CICS issues an authorization request for every transaction. It does this whether the transaction is started from a terminal, by using an EXEC CICS START command, or triggered from the transient data queue, either with or without the termid operand. CICS performs this security check even

if no user has signed on. Users who do not sign on can use only those transactions that are authorized to the default user.

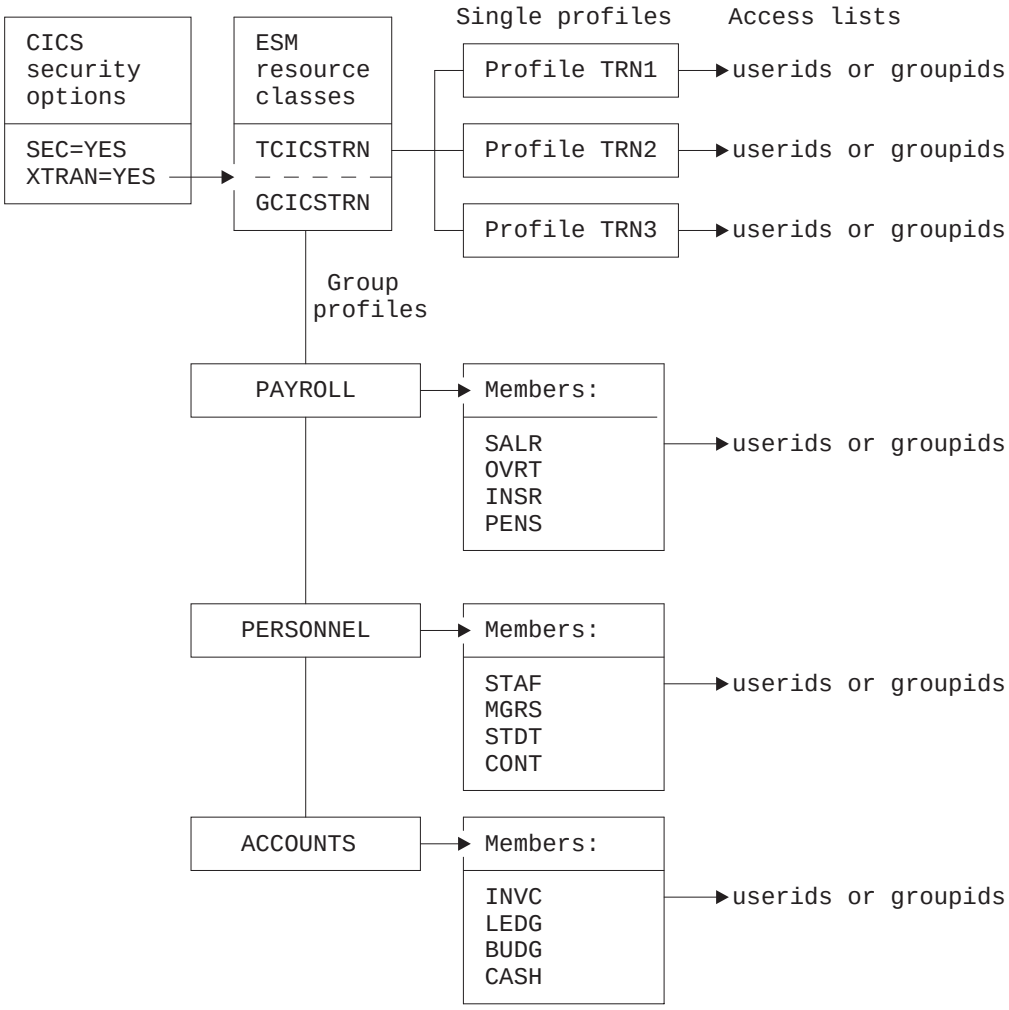


Figure 2. Illustration of the main elements of CICS transaction security

Transaction-attach processing when SEC=YES and XTRAN=YES

Every time a transaction is initiated at a CICS terminal, CICS issues an authorization request to determine whether the user associated with the terminal is authorized for that transaction. CICS and the ESM process the authorization request using the currently active transaction profiles in the ESM resource class identified by the XTRAN system initialization parameter.

Defining transaction profiles to the ESM

For those CICS regions running with transaction security checking, define transaction profiles for all transactions that need to be protected from unauthorized access. You can define these profiles either in the default transaction resource classes, or in installation-defined classes that you have added to the ESM. (See “IBM-supplied resource class names for CICS” on page 13 for information about the transaction resource classes.)

CEBT transaction

The CEBT transaction (the master terminal transaction used to control the alternate CICS system in an XRF environment) is not subject to transaction security checking. This means that any user is authorized to use CEBT. CEBT can only be issued from the operating system console.

Transactions not associated with a terminal

For all resource security checking, CICS needs a userid in order to check the user's authority to access the resource. CICS can protect resources against unauthorized use if those resources are used in transactions that are not associated with a terminal. In addition to transactions started by an EXEC CICS START command without a terminal identifier specified, there are two other types:

- Transactions started without a terminal when the trigger level is reached for an intrapartition transient data queue
- Programs executed from the second phase of the program list table (PLT) during CICS startup

Triggered transactions

The DFHDCT macro, and the ATIUSERID option of the EXEC CICS SET TDQUEUE command establish security for non-terminal transactions started by a transient data trigger level. The user issuing the SET, INSTALL, or CREATE command must have surrogate authority for the userid specified on the ATIUSERID option. The user to be associated with the triggered transaction is specified on the USERID attribute of the transient data queue definition.

PLT programs

If PLT programs are to be executed during CICS startup, CICS performs a surrogate user security check for the region userid. See "Defining user profiles for CICS region userids" on page 19. This check determines whether the CICS job is authorized to be the surrogate of the userid specified on the PLTPIUSR parameter. The PLTPIUSR and PLTPISEC system initialization parameters specify security options for PLT programs that are run from the third stage of CICS startup (which is the second phase of the PLTPI initialization).

When the PLTPISEC=NONE option is defined. No surrogate check is required for this. If your PLT programs issue START commands, the region userid has surrogate authority to start them when no userid is coded. Note that the starter always has surrogate authority to itself. When the started transaction starts up, another check is made to see if the userid has authority to attach the transaction and access this transaction in the TCICSTRN class. Rather than giving the region userid access to additional resources, you can use the PLTPIUSR and PLTPISEC parameters.

During shutdown, CICS runs PLT programs under the authority of the userid for the transaction that requested the shutdown. The values of the RESSEC and CMDSEC options for that transaction are also applied to the PLT programs. If RESSEC(YES) and CMDSEC(YES) are specified on the definition of the transaction issuing the EXEC CICS PERFORM SHUTDOWN command, security checking is done at the first stage of shutdown.

Chapter 6. Resource security

This chapter describes the facilities provided by CICS and an ESM for controlling access to resources protected by ESM general resource security classes. They are discussed in the following sections:

- "General resource security checking"
- "Security for general resource types" on page 48
- "Security checking of transactions running under CEDF" on page 56

Chapter 5, "Transaction security" on page 41 described how to control access to CICS transactions, using CICS transaction-attach security. This chapter describes how you can implement a further level of security, by controlling access to the resources used by the CICS transactions. The implication of this is that a user who is authorized to invoke a particular CICS transaction may not be authorized to access files, or other general resources used within the transaction. Unlike transaction-attach security, which cannot be turned off for individual transactions, you can control resource security checking at the individual transaction level.

Resources defined to CICS to support application programming languages are also subject to security checking if resource or command security checking is specified. For example, if a PL/I program abends, it may attempt to write diagnostic information to the CESE transient data queue. If resource checking is active, and the user is not authorized to write to the CESE transient data queue, the program will terminate with a 4094 abend.

You control who can access the general resources used by CICS transactions, by:

- Specifying SEC=YES as a system initialization parameter
- Specifying RESSEC=ALWAYS as a system initialization parameter
- Specifying RESSEC(YES) in the RDO TRANSACTION resource definition
- Specifying the types of resource you want to protect by defining CICS system initialization parameters for the ESM general resource classes
- Defining the CICS resources to the ESM in resource class profiles, with appropriate access lists

General resource security checking

CICS uses an ESM to protect the general resources that you can access through a CICS application program. Each resource is described briefly in Table 10 on page 46, with the associated CICS system initialization parameter that you use to specify the ESM resource class name.

Note that no authorization processing is done for BMS commands.

Table 10. General resource checking by CICS

CICS parameter	General resource protected	Further information
XAPPC	Partner logical units (LU6.2). This resource is included in this list for completeness, but is not discussed in this chapter.	Chapter 12, "Implementing LU6.2 security" on page 95.
XCMD	The subset of CICS application programming commands that are subject to command security checking. This resource is included in this list for completeness, but is not discussed in this chapter. EXEC CICS FEPI system commands are also controlled by this parameter.	Chapter 8, "CICS command security" on page 65.
XDCT	CICS extrapartition and intrapartition transient data destinations, also known as queues. Define profiles in the destination class to control who is allowed to access CICS transient data queues.	"Transient data" on page 48.
XFCT	CICS file-control-managed VSAM and DAM files. Define profiles in the file class to control who is allowed to access CICS VSAM and DAM files.	"Files" on page 50.
XJCT	CICS system log and user journals. Define profiles in the journal class to control who is allowed to access CICS journals on CICS log streams.	"Journals" on page 51.
XPCT	CICS started transactions and EXEC CICS commands: COLLECT STATISTICS TRANSACTION, CREATE TRANSACTION, DISCARD TRANSACTION, INQUIRE TRANSACTION, INQUIRE REQID, SET TRANSACTION, and CANCEL. Define profiles in the started-transactions class to control who is allowed access to started CICS transactions.	"Started and XPCT-checked transactions" on page 52.
XPPT	CICS application programs. Define profiles in the program class to control who is allowed to access CICS application programs that a CICS application invokes by means of a LINK, XCTL, or LOAD command.	"Application programs" on page 54.
XTRAN	CICS transactions. This resource is included in this list for completeness, but is not discussed in this chapter.	Chapter 5, "Transaction security" on page 41.
XTST	CICS temporary storage destinations. Define profiles in the temporary storage class to control who is allowed to access CICS temporary storage queues.	"Temporary storage" on page 55.
XUSER	Surrogate user security. This resource is included in this list for completeness, but is not discussed in this chapter.	Chapter 7, "Surrogate user security" on page 59.

RESSEC transaction resource security parameter

Specifying RESSEC(YES) in the definition of an RDO TRANSACTION resource definition, together with the appropriate resource classes defined in the system initialization parameters, introduces another layer of security checking in addition to the transaction-attach security described in “Transaction-attach processing when SEC=YES and XTRAN=YES” on page 42.

For most simple (or single-function) transactions, this extra layer of security is not necessary. For example, if the transaction is designed to enable the terminal user to update a personnel file and nothing else, it should be sufficient to authorize access to the transaction without controlling access to the file also. However, if you have a complex transaction that offers users a choice of functions, or you are unsure about all the options available within a transaction, you may want to add the extra layer of security to restrict access to the data as well as to the transaction. Before implementing resource security checking, take into account the extra overhead that resource security checking involves, and only implement it if you believe the extra cost is worthwhile.

If you specify RESSEC(YES) on an RDO TRANSACTION resource definition, CICS calls the ESM for each CICS command that applies to a resource for which you have requested security, using an X*name* resource class parameter. This is shown in Figure 3, in which the execution of transaction TRN1 results in seven calls to the ESM.

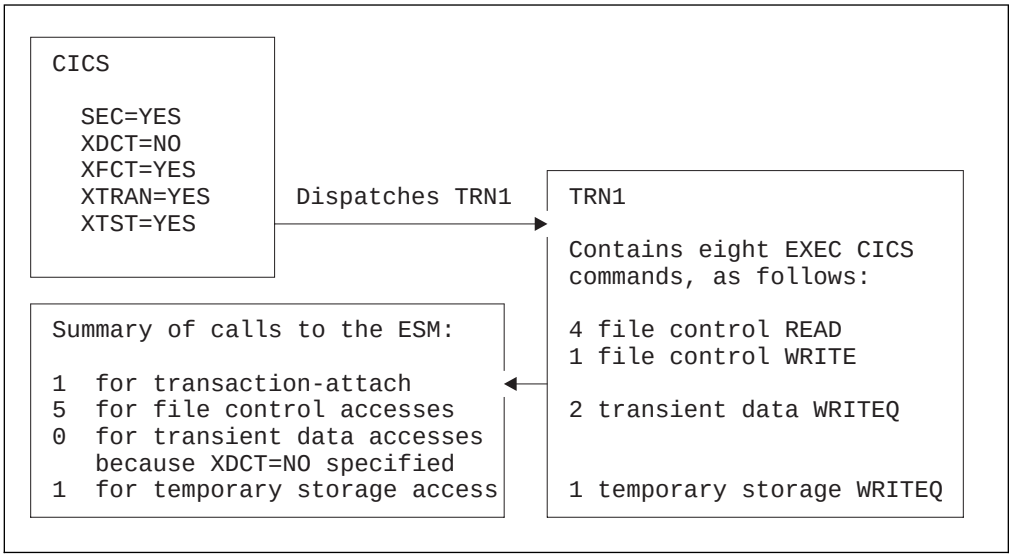


Figure 3. Multiple calls to the ESM with resource security checking

The RESSEC system initialization parameter

You can force the effect of RESSEC=YES for all CICS transactions by specifying the RESSEC=ALWAYS system initialization parameter. In general, this is not recommended, for the following reasons:

- For most simple transactions, just controlling access to the transaction is enough to control everything that the transaction can do.
- Invoking a resource check for every CICS resource consumes extra overhead that reduces the performance of all your transactions.

- Some CICS-supplied transactions may access resources of which you are unaware. It is your responsibility to ensure that users of these transactions are given enough authority to allow the transactions to continue to work.

Authorization failures

If a terminal user is not authorized to access the resource specified on a CICS command, CICS returns the NOTAUTH condition to the application program. CICS indicates this authorization failure by setting the EIBRESP field of the EXEC interface block (DFHEIBLK) to a value of 70 (and X'46' in byte 0 of the EIBRCODE field). Design your CICS applications to handle security violations by passing control to an appropriate routine. They can do this in either of the following ways:

- Test the EIBRESP condition by adding the RESP option to each command that may receive a NOTAUTH condition. For example (in COBOL):

```
EXEC CICS READ FILE('FILEA')
   INTO(REC) RIDFLD(KEY)
        RESP(COMMAND-RESPONSE)
END-EXEC.


EVALUATE COMMAND-RESPONSE
   WHEN DFHRESP(NORMAL)
        CONTINUE
   WHEN DFHRESP(NOTAUTH)
    PERFORM SECURITY-ERROR
END-EVALUATE.
```

- Code an EXEC CICS HANDLE CONDITION NOTAUTH(*label*) command, where *label* is the name of the security violation routine.

If an application does not cater for security violations, CICS abends the transaction with an AEY7 abend code.

Security for general resource types

This section discusses some of the resource types for which security can be implemented. This includes:

- "Transient data"
- "Files" on page 50
- "Journals" on page 51
- "Started and XPCT-checked transactions" on page 52
- "Temporary storage" on page 55
- "Application programs" on page 54

Transient data

To implement security for transient data destinations (queues), do the following:

1. Specify RESSEC(YES) in the CSD resource definition of the appropriate transactions.
2. Define profiles to the ESM in the DCICSDCT or ECICSDCT resource classes (or their equivalent if you have user-defined resource class names), and

authorize users as appropriate. Transient data queue names are a maximum of 4 characters in length, such as CSMT, L86O, L86P, and so on.

3. Specify SEC=YES as a CICS system initialization parameter (and SECPRFX=YES if you define profiles with the CICS region userid as a prefix).
4. Specify XDCT=YES for the default resource class names of DCICSDCT and ECICSDCT (or XDCT=class_name for user-defined resource class names).

Defining profiles for transient data queues

When you are defining profile names to an ESM to control access to transient data queues, define profiles only for queues that are defined to CICS as follows:

TYPE=INTRA For an intrapartition transient data queue held on the CICS intrapartition (VSAM) data set, DFHNTRA. When the destination facility is a file, you can specify a USERID. See "Considerations for triggered transactions" on page 50 for more information about intrapartition TD queues in this category, and "Transient data trigger-level transactions" on page 61 for more information about the USERID specification.

TYPE=EXTRA For an extrapartition transient data queue on a sequential data set.

If you define an indirect queue, CICS directs this to another queue, which can be extrapartition, intrapartition, or remote. The redirection can even be to another indirect queue. See the *CICS Resource Definition Guide* for more information about defining CICS transient data queues.

If you are running CICS with security checking for transient data queues, CICS issues a call to the ESM for each command that specifies a queue name. However, the resource name that CICS passes to the ESM is the queue name of the final queue, which is not necessarily the name of the queue specified on the command.

For example, if an EXEC CICS command specifies queue QID2, which is defined as indirect to QID1, CICS calls the ESM for an authorization check on QID1, not QID2. This is illustrated as follows:

```
TDQ definition:      DFHDCT TYPE=EXTRA,
                            DESTID=QID1
                            DSCNAME=sdsciname

                     DFHDCT TYPE=INDIRECT,
                            DESTID=QID2,
                            INDDEST=QID1

CICS transaction:    EXEC CICS WRITEQ TD
                            QUEUE(QID2)
                            FROM(data-area)
                            LENGTH(length)


CICS calls the ESM: Does the terminal user of the CICS transaction
                    have UPDATE authorization for QID1?
```

Access authorization levels

You can read an item from a transient data queue only once, because whenever you read from a transient data queue, CICS deletes the entry (by performing a "destructive read"). Therefore, if you specify security with SEC=YES as a system initialization parameter, CICS requires a minimum authorization level of UPDATE for all TD commands (DELETEQ, WRITEQ, and READQ).

CICS-required destination control table entries

CICS itself uses a number of queues. These queues are defined in the sample copybook DFH$DCTR in the VSE/ESA sublibrary PRD1.BASE. You may want to protect access to these definitions from user application programs. In the sample table, most of the queue names are indirect, pointing to the final queue, CSSL. Therefore, if you use the definitions as supplied, you need define to the ESM only the queue name CSSL. You may want also to consider defining the queue names CESO and CESE, the EXTRA partition queues for LE/VSE to the ESM.

Considerations for triggered transactions

For intrapartition TD queues with a trigger level greater than zero, CICS derives the userid associated with the triggered transaction from the following sources:

- The USERID parameter specified on the intrapartition transient data resource definition (DESTFAC=FILE).
- The userid associated with the terminal (for queues that have been defined with a destination facility of terminal) (DESTFAC=TERMINAL). This can be the CICS default userid if no user is signed on at the terminal.
- The link userid on the connection definition (for queues that have been defined with a destination facility of system) (DESTFAC=SYSTEM).

Files

CICS application programs process files, which, to CICS, are logical views of physical VSAM or DAM data sets. You identify a file to CICS by a 7-character file name, and you can define many files to CICS that refer to the same physical data set, which is separately identified by a 44-character data set name (DSNAME). For example, you can define file resource definitions called FILEA, FILEB, and FILEC, all of which refer to one physical VSAM data set, but with each file definition specifying different attributes.

CICS transactions access the data in physical data sets using the CICS file control name. Therefore, you control access to CICS-managed files by defining profiles in the ESM general resource classes for CICS files, not in the ESM data set class. You define the profiles using the CICS 7-character file name to identify the resource. Data set authorization based on the 44-character data set name is used only during OPEN processing, to determine whether the CICS region userid is authorized to access the data set for which the OPEN has been requested. This does not depend on the userid running the transaction that caused the OPEN to be performed.)

To implement security for files managed by CICS file control:

1. Specify RESSEC(YES) in the CSD resource definition of the transactions that access the files.

2. Define profiles to the ESM in the FCICSFCT or HCICSFCT resource classes (or their equivalent if you have user-defined resource class names), using the CICS file names to identify the profiles.
3. Specify SEC=YES as a CICS system initialization parameter (and SECPRFX=YES if you define profiles with the CICS region userid as a prefix).
4. Specify XFCT=YES for the default resource class names of FCICSFCT and HCICSFCT (or XFCT=class_name for user-defined resource class names).

Note that RDO transactions do not use file commands to access the CSD, and are not, therefore, subject to these mechanisms.

Access authorization levels

If you specify security with SEC=YES as a system initialization parameter, CICS requires a level of authorization appropriate to the file access intended: a minimum of READ for read intent, and a minimum of UPDATE for update or delete intent.

Journals

CICS provides facilities to write to and read from:

- The CICS system log
- The CICS general logs, which comprise user journals, forward recovery logs, and autojournals

The system log is used only for recovery purposes—for example, during dynamic transaction backout, or during emergency restart. Do not use it for any other purpose. Do not, therefore, write to it from a user application using the EXEC CICS WRITE JOURNALNAME command.

CICS uses journal identifier **DFHJ01** for its primary system log.

In addition to the automatic journaling and forward recovery logging that CICS performs for user transactions (depending on the options in the file resource definitions), user applications can also write user journal records using the EXEC CICS WRITE JOURNALNAME command.

Users needing to write journal records must have authority to write to the JOURNALNAME (as defined in the general resource class for CICS journals). CICS calls the ESM to perform a security check only for attempts to access a user journal by a CICS API command, and not for the journaling it performs in response to journaling options in the file resource definition. The CICS API does not provide a READ command for reading journals from a CICS transaction. For this reason, with proper exercise of control over the installation of applications on your CICS systems, you might consider it unnecessary to add protection for journals that cannot be read from within CICS.

If you decide to implement security for CICS journals:

1. Specify RESSEC(YES) in the CSD resource definition of the transactions that write to journals.
2. Define profiles to the ESM in the JCICSJCT or KCICSJCT resource classes (or their equivalent if you have user-defined resource class names) using the CICS journal name to identify the profiles.

3. Specify SEC=YES as a CICS system initialization parameter (and SECPRFX=YES if you define profiles with the CICS region userid as a prefix).
4. Specify XJCT=YES for the default resource class names of JCICSJCT and KCICSJCT (or XJCT=class_name for user-defined resource class names).

Access authorization levels

If you specify security with SEC=YES as a system initialization parameter, CICS requires a minimum authorization of UPDATE for journal access.

Started and XPCT-checked transactions

A CICS transaction initiated by a terminal user can start other transactions by means of an EXEC CICS START command. Transactions started in this way are known as **started transactions**, and you can use CICS external security to control who can start other transactions using the START command.

Started transactions are defined in the ACICSPCT and BCICSPCT resource class profiles. These profiles also control access to transactions specified in certain other EXEC CICS commands, if the transaction issuing the command is defined with RESSEC(YES). The commands affected are:

- COLLECT STATISTICS TRANSACTION
- CREATE TRANSACTION
- DISCARD TRANSACTION
- INQUIRE TRANSACTION
- SET TRANSACTION
- INQUIRE REQID
- CANCEL

When a transaction issues an EXEC CICS START TRANSID(*tranid*) command, CICS calls the ESM to check that the user of the transaction issuing the command is authorized for the started transaction.

To implement security for started transactions and for transactions checked against the XPCT class:

1. Specify RESSEC(YES) in the CSD resource definition of the transactions that issue START commands.
2. Define profiles to the ESM in the ACICSPCT or BCICSPCT resource classes (or their equivalent if you have user-defined resource class names) using the name of the started transaction to identify the profiles.
3. Specify SEC=YES as a CICS system initialization parameter (and SECPRFX=YES if you define profiles with the CICS region userid as a prefix).
4. Specify XPCT=YES for the default resource class names of ACICSPCT and BCICSPCT (or XPCT=class_name for user-defined resource class names).

Transactions started at terminals

The EXEC CICS START command enables a CICS application program to start another transaction associated with a terminal other than the one from which the start command is issued. For example, the following command issued in CICS transaction `tranid1`, invoked at `termid1`, starts another transaction called `tranid2` at `termid2`:

```
EXEC CICS START
     TRANSID(tranid2)
     AT HOURS('18') MINUTES('50')
     TERMID(termid2)
```

When a TERMID is specified for the started transaction, CICS performs a transaction-attach security check, using the classes TCICSTRN and GCICSTRN, on the userid associated with the terminal (`termid2` in this example). You must therefore ensure that the userid associated with the terminal (`termid2`) is authorized to invoke the transaction. This userid is that of the signed-on user, or the CICS default userid if no user is signed on. If `termid2` is **not** authorized, message DFHAC2033 is issued to the user of `termid2`. The user of the terminal that issued the START command gets a "normal" response. If the started transaction is defined with RESSEC(YES), also ensure that the userid associated with the terminal (`termid2` in this example) is suitably authorized to access protected resources.

Starting tasks at terminals defined with preset security: Typically, started transactions associated with a terminal are printing tasks, where the specified terminal is a printer. In this case, to associate a specific userid with the terminal, you define the terminal with preset security. See "Preset terminal security" on page 5 for more information.

Transactions started without terminals

The EXEC CICS START command enables a CICS application program to start another transaction that is not associated with any terminal. When no TERMID is specified for the started transaction, the userid associated with the new transaction depends on whether you also specify the USERID option.

Userid of a non-terminal started transaction: The USERID option of the EXEC CICS START command determines the userid for a non-terminal started transaction. Without the USERID option, the non-terminal started transaction has the same userid as the transaction that executed the EXEC CICS START command. When an EXEC CICS START command is executed without the TERMID option, CICS performs a surrogate user check to ensure that the transaction is authorized for the userid to be used by the non-terminal started transaction. For information about the link authorization of surrogate users, see "Link security" on page 90.

Access to resources by a non-terminal started transaction: If the USERID option is not specified on an EXEC CICS START command, the non-terminal started transaction does not always inherit all of the security of the transaction that executed the command. Also, it does not inherit resource access determined by link security, or resource access determined by a userid for EDF when used in dual-screen mode. This means:

- If a transaction-routed transaction executes an EXEC CICS START command, or if an EXEC CICS START command is function shipped, the non-terminal started transaction is not subject to link security.
- If EDF is used in dual-screen mode for a transaction that issues an EXEC CICS START command, the non-terminal started transaction is not subject to resource access determined by the userid of the EDF terminal.

If you want the started transaction to have exactly the same security capabilities as the starting transaction, omit the USERID option. Without the USERID option, resource access by the non-terminal started transaction is determined by the sign-on parameters of the terminal transaction, including the port of entry at which the terminal user signed on; that is, the terminal or console used to sign on, as shown in the following example:

> A terminal user signs on using the CESN transaction at a terminal with netname `NETNAMEX`. For the ESM, therefore, the port of entry is `NETNAMEX`. At the CESN screen the terminal user enters userid `USERID1`. The terminal user then runs a terminal transaction which executes an EXEC CICS START command without the TERMID option or the USERID option specified. The non-terminal started transaction has resource access determined by userid `USERID1`, and port of entry `NETNAMEX`.

If a non-terminal transaction is denied access to a resource by the ESM the error message produced can include the terminal sign-on parameters, and the userid. It can also include a port of entry. The userid, and port of entry can be those inherited from the terminal transaction that started the non-terminal transaction.

It is recommended that you do not specify the current userid of a terminal transaction on the USERID option. The non-terminal started transaction may not have the same resource access as the terminal transaction. The following example shows how the non-terminal started transaction can have different resource access:

> If an EXEC CICS START TRANSID USERID command is executed by a terminal transaction specifying the same userid that the terminal user entered when signing on with CESN, the started transaction has access to resources determined by the specified userid, but not to the resources determined by the port of entry.

Access authorization levels

CICS requires a minimum authorization of READ for started transactions.

Application programs

You control access to the initial program specified in the transaction resource definition by authorizing the user to initiate the transaction (transaction-attach security). However, CICS application programs can invoke other programs by means of the LINK, LOAD, and XCTL commands. Also, the load status of programs can be altered by the CICS RELEASE, ENABLE, and DISABLE commands. Note, however, that there is no separate security check on the RELEASE of programs loaded for task lifetime. This is done on the corresponding LOAD.

You control access to programs invoked using these commands by defining profiles in the CICS application program classes, and which you define to CICS on the XPPT system initialization parameter.

To control which users can invoke or change the load status of other programs:

1. Specify RESSEC(YES) on the RDO TRANSACTION resource definitions of the transactions that use the above commands.
2. Define profiles to the ESM in the MCICSPPT or NCICSPPT resource classes (or their equivalent if you have user-defined resource class names) using the

name of the program invoked on the LINK, LOAD, or XCTL command to identify the profiles.

3. Specify SEC=YES as a CICS system initialization parameter (and SECPRFX=YES if you define profiles with the CICS region userid as a prefix).
4. Specify XPPT=YES as a CICS system initialization parameter for the default resource class names of MCICSPPT and NCICSPPT (or XPPT=class_name for user-defined resource class names).

Exception for distributed program link (DPL) commands

If CICS finds that a program referenced on an EXEC CICS LINK command is a remote program, it does not perform the security check in the region in which the link command is issued. The security check is performed only in the CICS region in which the linked-to program finally executes.

For example, if CICSA function ships a DPL command to CICSB, where the program then executes, CICSB issues the security check. If the DPL request is function shipped again to CICSC for execution, it is CICSC that issues the security check.

Access authorization levels

CICS requires a minimum authorization of READ for programs.

Temporary storage

Implementing security for temporary storage queues

To implement security for temporary storage queues:

1. Specify RESSEC(YES) on the RDO TRANSACTION resource definitions of the appropriate transactions.
2. Specify DFHTST TYPE=SECURITY entries in the CICS temporary storage table for the queues on which you want CICS to perform security checking. CICS does not perform any security checks on temporary storage queues that are not defined by TYPE=SECURITY entries in the TST.
3. Define profiles to the ESM in the SCICSTST or UCICSTST resource classes (or their equivalent if you have user-defined resource class names), with access lists as appropriate.

   For more information about defining temporary storage profiles, see “Other temporary storage security considerations” on page 56.
4. Specify SEC=YES as a CICS system initialization parameter (and SECPRFX=YES if you define profiles with the CICS region userid as a prefix).
5. Specify XTST=YES as a CICS system initialization parameter for the default resource class names of SCICSTST and UCICSTST (or XTST=class_name for user-defined resource class names).

Other temporary storage security considerations

When a CICS application issues a temporary storage command (for example, DELETEQ TS, READQ TS, or WRITEQ TS) and temporary storage security is in effect, CICS searches the TST for a DATAID that corresponds to the leading characters of the queue name.

Note that if you include a temporary storage queue with hexadecimal characters in a temporary storage queue name, unpredictable results may occur. Also, if a TSQ name contains an imbedded blank, the ESM may truncate the resource name to that blank.

Access authorization levels

If you specify security with SEC=YES as a system initialization parameter, CICS requires a level of authorization appropriate to the temporary storage queue access intended: a minimum of READ for READQ TS, and a minimum of UPDATE for DELETEQ TS and WRITEQ TS.

Security checking of transactions running under CEDF

When a transaction is run under the CEDF transaction, CICS determines the security processing for the target transaction from the logical OR of RESSEC in the resource definitions for the target transaction and the CEDF transaction.

Table 11 shows the security checking performed for the transaction XSUB for different settings of RESSEC.

Table 11. Security checking of transactions running under CEDF

CEDF	XSUB	Security checking
RESSEC(YES)	RESSEC(YES)	Any access to CICS resources causes a security check.
RESSEC(YES)	RESSEC(NO)	Any access to CICS resources causes a security check. (Logical OR results in RESSEC on.)
RESSEC(NO)	RESSEC(YES)	Any access to CICS resources causes a security check. (Logical OR results in RESSEC on.)
RESSEC(NO)	RESSEC(NO)	Access to CICS resources does not cause a security check. (Logical OR results in RESSEC off.)

To achieve the expected security processing for a transaction when it runs under CEDF, ensure that RESSEC for the CEDF transaction definition is set to NO. The IBM-supplied definition of CEDF in the DFHEDF group specifies RESSEC(YES). Definitions in the IBM-supplied groups cannot be modified, so to change the definition, copy it to another group.

When the CEBR and CECI are invoked from within EDF they are transaction-attach checked. The CMDSEC and RESSEC definitions used will be those currently installed for CEBR and CECI.

When CEDF is used in **two-terminal mode** (the CEDF is entered at a different terminal from the transaction being tested), the authorities of the user executing the CEDF transaction are taken into account, as well as those of the user executing the transaction being tested. For each resource accessed by the tested transaction, both users must have access authority, otherwise a NOTAUTH condition is raised. This applies to all resource checks:

- Transaction attach
- CICS resource
- CICS command
- Non-CICS resources accessed through the QUERY SECURITY command
- Surrogate user

Chapter 7. Surrogate user security

This chapter is in two main sections:

- "Where surrogate user checking applies"
- "ESM definitions for surrogate user checking" on page 62

Where surrogate user checking applies

CICS performs surrogate user security checking in a number of situations, using the surrogate user facility of an ESM. A surrogate user is one who has the authority to start work on behalf of another user. A surrogate user is authorized to act for that user without knowing that other user's password. To enable surrogate user checking, XUSER=YES must be specified as a system initialization parameter.

If surrogate user checking is in force, it applies to:

- The CICS default user
- PLT post-initialization processing
- Preset terminal security
- Started transactions
- The userid associated with a transient data destination
- The userid supplied as a parameter on an EXCI call

CICS default user

CICS performs a surrogate user security check against its own userid (the CICS region userid) to ensure that it is properly authorized as a surrogate of the default userid specified on the DFLTUSER system initialization parameter.

Post-initialization processing

If you specify a program list table on a PLTPI system initialization parameter, CICS checks that the region userid is authorized as a surrogate user of the userid specified in the PLTPIUSR system initialization parameter.

The PLTPIUSR system initialization parameter specifies the userid that CICS is to use for PLT programs that run during CICS initialization. All PLT programs run under the authority of the specified userid, which must be authorized to all the resources referenced by the programs.

The scope of PLT security checking is defined by the PLTPISEC parameter. This specifies whether command security checks and resource security checks are to apply to PLTPI programs.

If you do not specify the PLTPIUSR parameter, CICS runs PLTPI programs under the authority of the CICS region userid, in which case CICS does not perform a surrogate user check. However, the CICS region userid must then be authorized to all the resources referenced by the PLT programs. Furthermore, the CICS region userid is associated with any transactions started by PLT programs, and therefore must be authorized to run such transactions.

Preset terminal security

When you install a terminal that is defined with a preset security userid, CICS checks that the userid performing the install is authorized as a surrogate user of the preset userid. This is discussed in “Controlling the use of preset-security” on page 36.

Started transactions

CICS performs surrogate user checks when you use the EXEC CICS START command to start a transaction that is not associated with a terminal.

In the following, the userid under which the transaction issuing the START command runs is called the *starting-userid*, and the userid under which the started transaction runs is called the *started-userid*:

- If the TERMID option is specified on the START command, surrogate user checking does not apply. The *started-userid* is inherited from the terminal at which the transaction runs.
- If the USERID option is specified on the START command, the *started-userid* is set to that specified userid.
- If neither TERMID nor USERID is specified on the START command, the *started-userid* is set to be the same as the *starting-userid*.

CICS requires that all the userids associated with the transaction issuing the START are surrogates of the *started-userid*. CICS also assumes that any userid is always a surrogate of itself. So userids that are the same as *started-userid* are regarded as surrogates already, and the external security manager is not called for them.

A transaction can be associated with userids that are different from *starting-userid* when it is using CICS intercommunication, and when it is using EDF in the two-terminal mode.

Intercommunication and started transactions

If an EXEC CICS START command (without TERMID) is function shipped or is executed from a transaction-routed transaction, the command can be subject to link security. If link security is in effect, CICS also performs a surrogate user check to verify that the userid for link security is authorized as a surrogate user to the userid for the started transaction. The surrogate check is done at this stage even if the USERID is omitted (if the *started-userid* is different from the link userid). For more information see “Link security” on page 90.

EDF in dual-screen mode and started transactions

If an EXEC CICS START command (without TERMID) is executed under control of EDF in dual-screen mode, CICS also performs a surrogate user check, to verify that the userid for the EDF terminal is authorized as a surrogate user of the userid for the started transaction. This check is done even if USERID is omitted, if the *started-userid* is different from the EDF userid.

Surrogate user checking can be subject to link security. If EDF is in use in dual-screen mode, the security of the user executing EDF is also checked. If a NOTAUTH condition occurs with an EXEC CICS START command, this can be because of link security or because of EDF user security.

Transient data trigger-level transactions

When a transient data queue is defined by a DFHDCT macro with a non-terminal trigger-level transaction and a USERID parameter, CICS checks that its own userid (the CICS region userid) is authorized as a surrogate user of the userid specified on the trigger-level transaction, during the installation of the transient data resource definition.

The userid for a transient data trigger-level transaction that is not associated with a terminal can be specified on the transient data definition or on the EXEC CICS SET TDQUEUE system programming command.

Intrapartition transient data resources.

CICS uses the userid specified on transient data queue definition for security checking in any trigger-level transactions that are not associated with a terminal. Code the USERID operand with the userid that you want CICS to use for security checking for the trigger-level transaction specified on the TRANSID operand. USERID is valid only when the destination facility is a file.

The trigger-level transaction runs under the authority of the specified userid, which must be authorized to all the resources used by the transaction.

If you omit the userid from a qualifying trigger-level entry, CICS uses the default userid specified on the DFLTUSER system initialization parameter. Ensure that the userid of any CICS region in which the transient data queue definition is installed is defined as a surrogate of all the userids specified in the DCT. This is because, during a cold start, CICS performs a surrogate user security check for the CICS region userid against all the userids specified in transient data queue definitions that are being installed. If the surrogate security check fails, the transient data queue definition is not installed.

EXEC CICS SET TDQUEUE ATIUSERID

The system programming command, EXEC CICS SET TDQUEUE ATIUSERID, specifies the userid for a transient data trigger-level transaction that is not associated with a terminal. The destination facility must be a file.

CICS performs a surrogate user security check against the userid of the transaction that issues the EXEC CICS SET TDQUEUE command, to verify that the transaction userid is authorized as a surrogate user of the userid specified on the ATIUSERID parameter.

Userid passed as parameter on EXCI calls

A surrogate user check is performed to verify that the batch region's userid is authorized to issue DPL calls for another user (that is, is authorized as a surrogate of the userid specified on the DPL_request call).

If you want your external CICS interface (EXCI) client jobs to be subject to surrogate user checking, specify SURROGCHK=YES in the EXCI options table, DFHXCOPT. If you specify SURROGCHK=YES, authorize the batch region's userid as a surrogate of the userid specified on all DPL_request calls. This means the batch region's userid must have READ access to a profile named “userid.DFHEXCI” in the SURROGAT general resource class (where “userid” is the userid specified on the DPL call).

If surrogate user checking is enabled (SURROGCHK=YES), but no userid is specified on the DPL call, no surrogate user check is performed, because the userid on the DPL call defaults to the batch region's userid.

If you do not want surrogate user security checking, specify SURROGCHK=NO in the DFHXCOPT options table.

Surrogate user checking is useful when the batch region's userid is the same as the CICS server region userid, in which case the link security check is bypassed. In this case, a surrogate user check is recommended, because the USERID specified on the DPL call is not an authenticated userid (no password is passed).

If the batch region's userid and the CICS region userid are different, link security checking is enforced. With link security, an unauthenticated userid passed on a DPL call cannot acquire more authority than that allowed by the link security check. It can acquire only the same, or less, authority than allowed by the link security check.

ESM definitions for surrogate user checking

To enable CICS surrogate user checking:

- Define the appropriate SURROGAT class profiles for CICS in the ESM database.
- Authorize CICS surrogate users to the appropriate SURROGAT profiles.

There are two forms of surrogate class profile names that you can define for CICS surrogate user checking. The names of these SURROGAT class profiles must conform to the following naming conventions:

*userid***.DFHSTART**
: *userid* is the userid under which a started transaction is to run.

*userid***.DFHINSTL**
: *userid* represents on of the following:

  - The PLT userid specified on the PLTPIUSR system initialization parameter
  - The userid associated with a trigger-level transaction
  - The CICS default userid specified on the DFLTUSER system initialization parameter
  - The userid specified for preset terminal security

There is also a form of surrogate class profile that you can define for external CICS interface (EXCI) security checking:

*userid***.DFHEXCI**
: *userid* represents the user specified on the DPL call in the client batch region.

  To authorize a surrogate to this EXCI profile, grant the EXCI batch region's userid READ access.

  Note that surrogate security checks in an EXCI batch region are independent of security definitions in the target CICS region. If SURROGCHK is specified in the EXCI options table (DFHXCOPT),

surrogate security checks are performed in the EXCI client program's address space regardless of the CICS security settings.

To authorize a surrogate user to one of these profiles, you must grant READ access.

You do not need to define a user as that user’s own surrogate. CICS bypasses the surrogate check in this case.

Chapter 8. CICS command security

CICS command security applies to System Programming (SP)-type commands; that is, commands that require the special CICS translator option, SP. Security checking is performed for these commands when they are issued from a CICS application program, and for the equivalent commands that you can issue with the CEMT master terminal transaction. Table 12 on page 66 shows the commands that are subject to command security checking.

This chapter discusses security for these commands as follows:

- "CICS resources subject to command security checking"
- "Parameters for specifying command security" on page 67
- "Security checking of transactions running under CEDF" on page 69
- "CEMT considerations" on page 70
- "Authorization failures" on page 70

CICS Front End Programming Interface security uses the same mechanism for authorization as the SP-type commands, using the FEPIRESOURCE resource name. Front End Programming Interface security is not discussed in this book. See the *CICS Front End Programming Interface User's Guide* for details.

The system programming commands are:

- COLLECT
- CREATE
- DISABLE
- DISCARD
- ENABLE
- EXTRACT
- INQUIRE
- INSTALL
- PERFORM
- RESYNC
- SET

You should refer to the ESM documentation for the appropriate ESM resource class access levels required to use these commands.

To determine who is allowed to use the (SP) option on the CICS translator, you can use the ESM to control who is allowed to load the DFHEITBS table at translation time. DFHEITBS is the language definition table that defines the SP-type commands, and is loaded only on demand.

CICS resources subject to command security checking

For transaction and resource security checking, you identify the resources to the ESM using the identifiers you have assigned to them, such as file names, queue names, transaction names, and so on. However, in the case of command security, the resource identifiers are all predefined by CICS, and you use these predefined names when defining resource profiles to the ESM. The full list of resource identifiers that are subject to command security checking, together with the

associated commands, is shown in Table 12 on page 66. Note that most of these commands are common to both the CEMT and EXEC CICS interfaces; where they are unique to one or the other they are prefaced with **CEMT**, or **EXEC CICS**, as appropriate.

Note: Refer to the ESM documentation to determine the ESM resource name that corresponds to the CICS predefined resource name.

Table 12 (Page 1 of 2). CICS resources subject to command security checking

Resource name (see note 1)	**Related CICS command(s)**
AUTINSTMODEL	INQUIRE|DISCARD AUTINSTMODEL
AUTOINSTALL	INQUIRE|SET AUTOINSTALL
CONNECTION	INQUIRE|SET|CREATE|DISCARD CONNECTION
DELETSHIPPED	INQUIRE|SET|PERFORM DELETSHIPPED
DSNAME	INQUIRE|SET DSNAME
DUMP	PERFORM DUMP **CEMT** PERFORM SNAP
DUMPDS	INQUIRE|SET DUMPDS
EXCI	**CEMT** INQUIRE EXCI
EXITPROGRAM	**EXEC CICS** ENABLE PROGRAM **EXEC CICS** DISABLE PROGRAM **EXEC CICS** EXTRACT EXIT **EXEC CICS** RESYNC ENTRYNAME
FEPIRESOURCE	Certain EXEC CICS FEPI commands (see note 3)
FILE	INQUIRE|SET|CREATE|DISCARD FILE
IRC	INQUIRE|SET IRC
JOURNALNUM	INQUIRE|SET JOURNALNUM
LSRPOOL	CREATE LSRPOOL
MAPSET	CREATE|DISCARD MAPSET
MODENAME	INQUIRE|SET MODENAME
MONITOR	INQUIRE|SET MONITOR
PARTITIONSET	CREATE|DISCARD PARTITIONSET
PARTNER	INQUIRE|CREATE|DISCARD PARTNER
PROFILE	INQUIRE|CREATE|DISCARD PROFILE
PROGRAM	INQUIRE|SET|CREATE|DISCARD PROGRAM
REQID	**EXEC CICS** INQUIRE|SET REQID
RESETTIME	PERFORM RESETTIME (see note 4)
SECURITY	PERFORM SECURITY REBUILD
SESSIONS	CREATE|DISCARD SESSIONS
SHUTDOWN	PERFORM SHUTDOWN (see note 2)
STATISTICS	INQUIRE|SET STATISTICS **EXEC CICS** COLLECT STATISTICS, and PERFORM STATISTICS RECORD
STORAGE	INQUIRE STORAGE
SYSDUMPCODE	INQUIRE|SET SYSDUMPCODE (see note 4)
SYSTEM	INQUIRE|SET SYSTEM
TASK	INQUIRE|SET TASK and TASK LIST

Table 12 (Page 2 of 2). CICS resources subject to command security checking	
Resource name (see note 1)	**Related CICS command(s)**
TCLASS	INQUIRE|SET|DISCARD TCLASS and INQUIRE|SET|CREATE|DISCARD TRANCLASS
TDQUEUE	INQUIRE|SET TDQUEUE
TERMINAL	INQUIRE|SET|CREATE|DISCARD TERMINAL and INQUIRE|SET NETNAME
TRACEDEST	**EXEC CICS** INQUIRE|SET TRACEDEST
TRACEFLAG	**EXEC CICS** INQUIRE|SET TRACEFLAG
TRACETYPE	**EXEC CICS** INQUIRE|SET TRACETYPE
TRANDUMPCODE	INQUIRE|SET TRANDUMPCODE (see note 4)
TRANSACTION	INQUIRE|SET|DISCARD|CREATE TRANSACTION
TSQUEUE	**EXEC CICS** INQUIRE TSQUEUE
TYPETERM	CREATE|DISCARD TYPETERM
VTAM	INQUIRE|SET VTAM

Notes:

1. If you are using prefixing, the CICS region userid must be prefixed to the command resource name.
2. Be particularly cautious when authorizing access to these and any other CICS commands that include a SHUTDOWN option.
3. For more information about FEPI security, see the *CICS Front End Programming Interface User's Guide*.
4. See "Resource names for CEMT" on page 70.

If you are running CICS with command security, define resource profiles to the ESM, using the ESM-specific resource names corresponding to the CICS resource names listed in Table 12 on page 66 as the profile names in the CCICSCMD OR VCICSCMD resource classes (or their equivalent if you have user-defined resource class names).

If you are running CICS with SEC=YES, users require access equivalent to the type of command being issued, for example, INQUIRE or SET, because this access level is dependent on the ESM.

Parameters for specifying command security

In addition to the SEC and SECPRFX system initialization parameters, which are described in "SEC" on page 24, and "SECPRFX" on page 25, CICS provides the XCMD system initialization parameter and the CMDSEC resource definition option to enable you to specify that you want command security.

XCMD system initialization parameter

The XCMD security parameter is a CICS system initialization parameter. You can specify whether you want command security active in the CICS region, and optionally specify the ESM resource class name in which you have defined the command security profiles.

If you are using the IBM-supplied resource class names for CICS command profiles (CCICSCMD and VCICSCMD), specify XCMD=YES. CICS then requests the ESM to build the in-storage profiles from these default resource classes.

If you are using installation-defined resource class names for CICS command profiles, specify XCMD=*user_class*, and CICS requests the ESM to build the in-storage profiles from your own installation-defined resource classes.

If you do not want command security in a CICS region, specify XCMD=NO.

The CMDSEC system initialization parameter

You can force the effect of CMDSEC=YES for all CICS transactions by specifying the CMDSEC=ALWAYS system initialization parameter. In general, this is not recommended, for the following reasons:

- For most simple transactions, just controlling access to the transaction is enough to control everything that the transaction can do.
- Invoking a command check for every CICS command consumes extra overhead that reduces the performance of all your transactions.

The CMDSEC option is recommended for installations that need total control of the SP-type commands.

The default value is CMDSEC=ASIS, which causes command security to recognize the specification of CMDSEC in the individual transaction definitions.

The CMDSEC transaction definition parameter

As described earlier in this section, the XCMD parameter enables command security to be active. You specify which transactions you want command security to apply to by using the CMDSEC option on the transaction resource definition, as follows:

CMDSEC(NO) You do not want command security checking the transaction.

CMDSEC(YES) You want command security checking on the SP commands in Table 12 on page 66.

For each of these commands issued in a user application or by the CICS-supplied transactions CEMT and CECI, CICS calls the ESM to check that the terminal operator who initiated the transaction has authority to use the command for the specified resource.

Security checking of transactions running under CEDF

When a transaction runs under the CEDF transaction, CICS determines the security processing for the target transaction from the logical OR of the CMDSEC settings in the resource definitions for the target transaction and the CEDF transaction.

Table 13 shows the security checking performed for the transaction XSUB for different settings of CMDSEC.

Table 13. Security checking for transactions running under CEDF

CEDF	XSUB	Security checking
CMDSEC(YES)	CMDSEC(YES)	Any access to CICS commands causes a security check.
CMDSEC(YES)	CMDSEC(NO)	Any access to CICS commands causes a security check. (Logical OR results in CMDSEC on.)
CMDSEC(NO)	CMDSEC(YES)	Any access to CICS commands causes a security check. (Logical OR results in CMDSEC on.)
CMDSEC(NO)	CMDSEC(NO)	Access to CICS commands does not cause a security check. (Logical OR results in CMDSEC off.)

To achieve the expected security processing for a transaction when it runs under CEDF, ensure that CMDSEC for the CEDF transaction definition is set to NO. The IBM-supplied definition of CEDF in the DFHEDF group specifies CMDSEC(YES). Definitions in the IBM-supplied groups cannot be modified, so to change the definitions, copy them to another group.

When CEBR or CECI is invoked from within EDF it is transaction-attach checked. The CMDSEC and RESSEC definitions used will be those currently installed for CEBR or CECI.

When CEDF is used in **two-terminal mode** (the CEDF is entered at a different terminal from the transaction being tested), the authorities of the user executing the CEDF transaction are taken into account, as well as those of the user executing the transaction being tested. For each resource accessed by the tested transaction, both users must have access authority, otherwise a NOTAUTH condition is raised. This applies to all resource checks:

- Transaction-attach
- CICS resource
- CICS command
- Non-CICS resources accessed through the QUERY SECURITY command
- Surrogate user

Note: When an EXEC CICS SIGNON, EXEC CICS VERIFY PASSWORD, or EXEC CICS CHANGE PASSWORD command is issued by a transaction running under CEDF, the password (and new password, where applicable) is blanked out.

CEMT considerations

In general, the resources that the CICS-supplied CEMT master terminal transaction operates on are the same as the equivalent SP-type commands of the CICS API shown on on page 65. If, in addition to normal transaction-attach security, you are using command security, you must ensure that authorized users of CEMT are also authorized for the CICS commands, as appropriate. If a user is authorized to initiate the CEMT transaction, but is not authorized for the resources on which the SP commands depend, CICS returns a NOTAUTH condition. To allow your system programmers to use the CEMT command in a command security environment, give them UPDATE access to the commands on which you want them to issue the PERFORM, SET, and DISCARD commands, and give them READ access to the commands on which you want them to issue only INQUIRE and COLLECT commands.

UPDATE authority should be given to users specifying XPPT=YES and XCMD=YES when they issue a CEMT SET PROG(xxx) NEWCOPY command.

Resource names for CEMT

In general, the resource names of the CEMT commands correspond to the resource names of the equivalent CICS API command. However, there are some exceptions, and in all these cases it is the API resource name that is shown in Table 12 on page 66, and for which you should define the corresponding ESM profile.

- The CEMT system dump option is spelled differently from the EXEC CICS equivalent. CEMT INQUIRE|SET SYDUMPCODE corresponds to EXEC CICS INQUIRE|SET SYSDUMPCODE.
- The CEMT transaction dump option is spelled differently from the EXEC CICS equivalent. CEMT INQUIRE|SET TRDUMPCODE corresponds to EXEC CICS INQUIRE|SET TRANDUMPCODE.
- The CEMT PERFORM RESET option corresponds to the EXEC CICS PERFORM RESETTIME command.
- The AUXTRACE, and INTTRACE options of the CEMT INQUIRE and SET commands all correspond to the TRACEDEST option of the API.

To use the CEMT INQUIRE|SET NETNAME command, you need access to the resource TERMINAL, not NETNAME.

Authorization failures

If you are running with CICS command security, CICS returns the NOTAUTH condition (RESP value 70) to your application, which is the same condition as for a resource security failure. (CICS also issues message DFHXS1111 to the CICS security transient data destination CSCS.) To test for this value in your application, we recommend you code DFHRESP(NOTAUTH) rather than explicitly coding a value. To distinguish between a command security failure and a resource security failure, check the RESP2 value. For a command security failure, CICS returns a

value of 100 in RESP2. For a resource security failure, a value of 101 is returned in RESP2.

For background information on using RESP and RESP2, see the *CICS Application Programming Guide*; for programming information, see the *CICS Application Programming Reference* and the *CICS System Programming Reference* manuals.

Chapter 9. Security checking using the QUERY SECURITY command

This chapter describes security checking by the user application using the EXEC CICS QUERY SECURITY command. The following topics are included:

- "How the QUERY SECURITY mechanism works"
- "QUERY SECURITY RESTYPE" on page 74
- "QUERY SECURITY RESCLASS" on page 77
- "Querying a user's surrogate authority" on page 77
- "Logging for QUERY SECURITY RESTYPE and RESCLASS" on page 78
- "Uses for QUERY SECURITY RESTYPE and RESCLASS" on page 78

An application can use the EXEC CICS QUERY SECURITY to request from the ESM the level of access a user has to a particular resource. The user in this context is the user invoking the transaction that contains the QUERY SECURITY command.

Issuing the QUERY SECURITY command does not actually grant or deny access to a resource (by issuing a NOTAUTH condition), but instead enables the application program to determine what action to take based on the CICS-value data area (CVDA) values that CICS returns. (For programming information on CVDAs, see the *CICS Application Programming Reference* manual.)

Note: QUERY SECURITY is **not** affected by the RESSEC and CMDSEC keywords on the transaction definition.

There are two distinct forms of the QUERY SECURITY command, depending on the options chosen.

- QUERY SECURITY RESTYPE
- QUERY SECURITY RESCLASS

(For programming information on the QUERY SECURITY command, see the *CICS Application Programming Reference* manual.)

How the QUERY SECURITY mechanism works

How the QUERY SECURITY mechanism works depends on:

- Whether SEC=YES or SEC=NO is specified in the system initialization parameters
- Whether SECPRFX=YES or SECPRFX=NO is specified in the system initialization parameters
- Which resource classes are active
- Whether the transaction issuing the request is subject to transaction routing, and if so:
  - Which ATTACHSEC parameter was specified on the connection definition

SEC and resource class system initialization parameters

Assuming SEC=YES is specified as a system initialization parameter, and the relevant resource class is active, for example, XFCT=YES is specified when issuing QUERY SECURITY RESTYPE(FILE). This command returns the level of access in terms of READ, UPDATE, CONTROL and ALTER. These access levels will correspond to levels specified in the ESM resource classes, and may vary between ESMs.

If, however, the relevant X*name* parameter is **not** active (for example, if XFCT=NO has been specified), or SEC=NO, the resource is READABLE, UPDATABLE, CTRLABLE and ALTERABLE.

SECPRFX system initialization parameter

If SECPRFX=YES is specified, CICS prefixes the resource with the CICS region userid. For example, issuing:

```
QUERY SECURITY RESTYPE('FILE') RESID('PAYFILE')
```

calls the ESM to check the terminal user's access to *cics_region_userid.PAYFILE* if SECPRFX=YES is specified. If SECPRFX=NO is specified, *PAYFILE* is checked.

Transaction routing

When the QUERY SECURITY command is issued from a transaction that has been routed to a remote system, CICS checks the link user's access to the specified resource, and the terminal user's access to the resource, if appropriate. For more information, see “Link security with LU6.2” on page 99, “Link security with LU6.1” on page 137, environment you are using. or “Link security with MRO” on page 145 according to the environment you are using.

QUERY SECURITY RESTYPE

Use the QUERY SECURITY RESTYPE command to query access levels to CICS resources contained in the classes activated at initialization by the ESM. The response to the QUERY SECURITY command indicates the result of a resource check on this resource. If the resource is not defined to the ESM, CICS does not grant access and the response is NOTREADABLE. Note that responses returned for category 3 transactions may not reflect that there is no attach time (TRANSATTACH) checking performed on category 3 transactions. Ensure the length of the resource name passed to the ESM with a RESTYPE request is the actual maximum length for that resource type.

RESTYPE values

RESTYPE is a resource type that corresponds to one of the X*name* system initialization parameters, and can take any of the values shown in Table 14 on page 75.

Table 14. QUERY SECURITY RESTYPE values	
RESTYPE value	**Xname parameter**
FILE	XFCT
JOURNALNUM	XJCT
PROGRAM	XPPT
SPCOMMAND	XCMD
TDQUEUE	XDCT
TRANSACTION	XPCT
TRANSATTACH	XTRAN
TSQUEUE	XTST

RESID values

In all cases (except for the SPCOMMAND resource type), the resource identifiers (RESID values) are defined by your installation.

When defining RESID values, be aware of the effects of using blanks (X'40') in resource identifiers. For example, in:

```
QUERY SECURITY RESTYPE('FILE') RESID('A B')
```

the blank delimits the RESID and causes the ESM to use a resource name of A.

For SPCOMMAND, the identifiers are predetermined by CICS. Table 15 lists the possible RESID values for SPCOMMAND. The ESM may use its own identifiers corresponding to the CICS predetermined names listed here.

Table 15. RESID values for RESTYPE(SPCOMMAND)		
AUTINSTMODEL	AUTOINSTALL	CONNECTION
DELETESHIPPED	DSNAME	DUMP
DUMPDS	EXCI	EXITPROGRAM
FEPIRESOURCE	FILE	IRC
JOURNALNUM	LINE	MODENAME
MONITOR	PARTNER	PROFILE
PROGRAM	REQID	RESETTIME
SECURITY	SHUTDOWN	STATISTICS
STORAGE	SYSDUMPCODE	SYSTEM
TASK	TCLASS	TDQUEUE
TERMINAL	TRACEDEST	TRACEFLAG
TRACETYPE	TRANDUMPCODE	TRANSACTION
TSQUEUE	VTAM	

QUERY SECURITY RESTYPE enables an application program to request from the ESM the level of access a terminal user has to the specified resource for the environment in which the transaction is running.

Before calling the ESM, CICS checks that the resource is installed. If the resource does not exist, CICS does not call the ESM and returns the NOTFND condition.

When the RESTYPE is TRANSATTACH and the transaction specified on the RESID parameter is unknown in the local region, a NOTFND condition is returned. However, if dynamic transaction routing is being used, there is no need for the

transaction to be installed in the terminal-owning region. The transaction specified on the DTRTRAN system initialization parameter is attached if an unknown transaction identifier is entered.

Application programmers should be aware that the NOTFND condition does not necessarily indicate that a terminal user will be unable to enter a transaction identifier, because the transaction may be routed dynamically.

Examples of values returned by QUERY SECURITY RESTYPE

This section gives a number of examples of the values returned by QUERY SECURITY RESTYPE, depending on what has been specified in the system initialization parameters.

SEC=NO

When SEC=NO is specified, issuing:

```
QUERY SECURITY RESTYPE('FILE') RESID('PAYFILE') ALTER(alter_cvda)
```

returns:

```
alter_cvda = DFHVALUE(ALTERABLE)
```

because SEC=NO means that no security checking is done for the entire CICS region.

SEC=YES and XFCT=NO

When SEC=YES and XFCT=NO are specified, issuing:

```
QUERY SECURITY RESTYPE('FILE') RESID('PAYFILE') ALTER(alter_cvda)
```

returns:

```
alter_cvda = DFHVALUE(ALTERABLE)
```

because XFCT=NO means that no security checking is done for files.

SEC=YES, XDCT=YES, and SECPRFX=NO

When SEC=YES, XDCT=YES, and SECPRFX=NO are specified, issuing:

```
QUERY SECURITY RESTYPE('TDQUEUE') RESID('TDQ1') READ(read_cvda)
```

returns:

```
read_cvda = DFHVALUE(READABLE)
```

if the user has READ (or higher) access to 'TDQ1' in the DCICSDCT class or the ECICSDCT group class.

SEC=YES, XTRAN=YES, and SECPRFX=YES

When SEC=YES, XTRAN=YES, and SECPRFX=YES are specified, issuing:

```
QUERY SECURITY RESTYPE('TRANSATTACH') RESID('TRN1') READ(read_cvda)
```

returns:

```
read_cvda = DFHVALUE(NOTREADABLE)
```

if the user **does not** have READ (or higher) access to `cics_region_userid.TRN1` in the TCICSTRN class or GCICSTRN group class.

SEC=YES, XCMD=$USRCMD, and SECPRFX=NO

When SEC=YES, XCMD=$USRCMD, and SECPRFX=NO are specified, issuing:

```
QUERY SECURITY RESTYPE('SPCOMMAND') RESID('VTAM') UPDATE(updt_cvda)
```

returns:

```
updt_cvda = DFHVALUE(UPDATABLE)
```

if the user has UPDATE access (or higher) to 'VTAM' in the C$USRCMD or V$USRCMD class.

QUERY SECURITY RESCLASS

Use the QUERY SECURITY RESCLASS command when you want to query access levels for non-CICS resources. RESCLASS is the name of a valid ESM general resource class, such as TERMINAL, FACILITY, or a similar installation-defined resource class. See “Other IBM-supplied resource class names affecting CICS” on page 14. The class name identified by RESCLASS is treated literally, with no translation.

Prefixing, as specified in the SECPRFX system initialization parameter, does not apply to QUERY SECURITY RESCLASS. That is, CICS does **not** prefix the RESID with the CICS-region userid before calling the ESM.

If SEC=NO is specified in the system initialization parameters, QUERY SECURITY RESCLASS always returns READABLE, UPDATABLE, CTRLABLE and ALTERABLE.

For QUERY SECURITY RESCLASS, both the RESID **and** the RESIDLENGTH option must be specified. The maximum length of a resource (RESID) is dependent upon the ESM. When defining RESID values, you should be aware of the effects of including blanks (X'40') in RESIDs. For example, in:

```
QUERY SECURITY RESCLASS('MYCLASS') RESID('MY PROFILE') RESIDLENGTH(10)
```

the presence of a blank may cause an INVREQ condition. This is because the ESM may not allow blanks to be embedded in a profile name.

Note: To determine access to CICS resources you should normally use RESTYPE, when the resource class is determined by the *Xname* system initialization parameter. However, if, for special reasons, you want to inquire about specific CICS resource classes, you should note that the class name must be the member class, and **not** the group class; that is, CCICSCMD, and not VCICSCMD.

Querying a user’s surrogate authority

To query a user’s surrogate authority, you can use the QUERY SECURITY command with the RESCLASS('SURROGAT') option. You also need to specify the RESID and RESIDLENGTH options. The RESID value you should provide is described in “ESM definitions for surrogate user checking” on page 62. However, this command is **not** controlled by the XUSER system initialization parameter, so you might obtain an unexpected response of NOTREADABLE if XUSER=NO has been specified. For example, to check whether the current user is allowed to start a transaction with a new userid of NEWUSER, when XUSER=YES is specified, issue the command:

```
QUERY SECURITY RESCLASS('SURROGAT') RESID('NEWUSER.DFHSTART')
RESIDLENGTH(16)  READ(cvda)
```

Logging for QUERY SECURITY RESTYPE and RESCLASS

You can control logging on the QUERY SECURITY command by specifying one of the following options:

- LOG
- NOLOG
- LOGMESSAGE(*cvda*), where *cvda* value is 54 for LOG, or 55 for NOLOG

The default is LOG.

If logging is in effect, and the terminal user does not have the requested access to the specified resource, message DFHXS1111 is issued to the CICS security transient data destination CSCS.

For programming information about CVDAs, refer to the *CICS System Programming Reference* manual.

Uses for QUERY SECURITY RESTYPE and RESCLASS

You can use the two forms of the QUERY SECURITY command in a number of different ways to customize resource security checking within an application. This section gives a number of examples of doing so.

Changing the level of security checking

You can use QUERY SECURITY to perform a different level of security checking from that which CICS would perform for application programs that specify RESSEC(YES) or CMDSEC(YES).

For example, suppose a transaction has RESSEC(YES) and contains a number of EXEC CICS READ FILE commands and a number of EXEC CICS WRITE FILE commands. For each command, CICS performs a security check to ensure that the terminal user has access to the relevant file, even though the same file may be being accessed each time. An alternative to this is to switch off security checking at the transaction level by specifying RESSEC(NO) on the transaction definition and then, when the application starts, execute a command such as:

```
EXEC CICS QUERY SECURITY RESTYPE('FILE') RESID(file_name) UPDATE(cvda)
```

This command allows the transaction to continue without any further calls to the ESM.

Note: Switching resource security checking off, using RESSEC(NO), means that **all** resource checks—not just of files as in the above example—are bypassed.

Checking which transactions to offer a user

You can use the QUERY SECURITY command to check whether a user is authorized to use a particular transaction **before** displaying the transaction code as part of an introductory menu. When you use the command for this purpose, you will probably want to avoid logging the checks for users who are not allowed to use certain transactions. To do this, use the NOLOG option.

Example of use of QUERY SECURITY RESCLASS

Normal CICS resource security checking for files operates at the file level only. You can use QUERY SECURITY to enable your application to control access to data at the **record** or **field** level.

To do this, define resource names (which represent records or fields within particular files) with the appropriate access authorizations for the records or fields you want to control. You could define these resources in an installation-defined ESM general resource class and then use the QUERY SECURITY RESCLASS command to check a terminal user's access to a specific field within a file before displaying or updating the field. (The application logic would determine which field.) For example:

```
QUERY SECURITY RESCLASS('$FILEREC') RESID('PAYFILE.SALARY')
RESIDLENGTH(14) READ(cvda) NOLOG
```

where '$FILEREC' is an installation-defined ESM general resource class. For more information, see "Designing applications to use the user-defined resources" on page 170.

Chapter 10. Security for CICS-supplied transactions

This chapter discusses security for CICS-supplied transactions, and contains a number of recommendations to ensure that your CICS regions are adequately protected. Where applicable, it describes the recommended security specifications that you will need for the CICS-supplied transactions defined in the group list DFHLIST, and stored in the CICS system definition data set (CSD). These recommendations cover all CICS-supplied transactions—those that are intended for use from a user terminal or console, and those that are for CICS internal use only. (For information about the CICS-supplied groups of resource definitions, and the DFHLIST group list, see the *CICS Resource Definition Guide*.)

CICS-supplied transactions are subject to transaction security checking in the same way as any other transactions.

Transaction security checking is controlled by the SEC= and XTRAN= system initialization parameters. For more information, see Chapter 5, “Transaction security” on page 41.

There is no parameter on the transaction resource definition that allows you to run with transaction security on some transactions but not others. If you are running with transaction security (SEC=YES and XTRAN=YES), CICS issues a security check for each transaction attach, to establish whether the user is permitted to run that transaction.

Categories of CICS-supplied transactions

For the purposes of this description, we divide the CICS-supplied transactions into three categories. Each transaction is identified within a category that describes its use within CICS. Each category specifies the recommended security specifications you need, in terms of both the CICS transaction definitions and the corresponding ESM profiles.

The three categories contain all the required CICS transactions, which are generated in their designated groups when you initialize your CICS system definition data set (CSD). The CSD does not include the CICS sample transactions (those that are in groups starting with DFH$). Sample applications should not require protection, because you are unlikely to install them on a CICS production system.

Category 1 transactions

Category 1 transactions are never associated with a terminal—that is, they are for CICS internal use only, and should not be invoked from a user terminal. CICS checks that the region userid has the authority to attach these transactions.

However, if the region userid is not authorized to access all of the category 1 transactions, CICS issues message DFHXS1113 and fails to initialize.

For category 1 transactions, specify the following:

To CICS RESSEC(NO) and CMDSEC(NO) on the transaction resource definition.

To the ESM Add the required definitions to the ESM to ensure that only your region userids have access to these transactions.

By doing this, you prevent any terminal user initiating these transactions (accidentally or otherwise). It is important that you do this, because permitting the initiation of these transactions at a terminal has unpredictable results.

Table 16 lists the category 1 transactions.

Table 16. Category 1 transactions

CSD group	Transaction	Program invoked	Description
DFHAKP	CSKP	DFHAKP	Writes system log activity keypoint
DFHBMS	CSPQ	DFHTPQ	Performs terminal page cleanup (BMS)
DFHFEPI	CSZI	DFHSZRMP	Implements Front End Programming Interface
DFHISC	CRSQ	DFHCRQ	Provides remote schedule purging (ISC)
	CSNC	DFHCRNP	Provides interregion control program (MRO)
DFHOPCLS	CSFU	DFHFCU	Opens user file-control managed files
DFHRMI	CRSY	DFHRMSY	Resynchronizes resource manager
DFHRSPLG	CSLG	DFHZRLG	Logs responses for protected message support
DFHSIGN	CESC	DFHCESC	Processes time-out and sign-off for idle terminals
DFHSPI	CATA	DFHZATA	Defines autoinstall terminal
	CATD	DFHZATD	Deletes autoinstall terminal
	CDTS	DFHZATS	Provides remote single delete transaction
	CITS	DFHZATS	Provides remote autoinstall transaction
	CMTS	DFHZATS	Remote mass delete transaction
	CFTS	DFHZATS	Provides remote mass flag transaction
	CRMD	DFHZATMD	Provides remote mass delete transaction
	CRMF	DFHZATMF	Provides remote mass flag transaction
DFHSTAND	CSTE	DFHTACP	Processes terminal abnormal conditions
	CXCU	DFHCXCU	Performs XRF tracking catch-up
	CXRE	DFHZXRE	Reconnects terminals following XRF takeover
DFHVTAM	CSNE	DFHZNAC	Provides VTAM node error recovery
N/A	CPLT	DFHSIPLT	Initializes PLT processing
N/A	CSSY	DFHAPATT	Provides entry point attach
N/A	CGRP	DFHZCGRP	Provides VTAM persistent sessions transaction
N/A	COVR	DFHZCOVR	Provides open VTAM retry transaction
N/A	CSTP	DFHZCSTP	Provides terminal control transaction

Category 2 transactions

Category 2 transactions either are initiated by the terminal user, or are associated with a terminal.

For the CICS resource definitions, the IBM-supplied transactions are defined with the recommended RESSEC and CMDSEC options. In particular, CECI, CEDF, CEMT, and CEST are all supplied with RESSEC(YES) and CMDSEC(YES). The mirror transactions are defined with RESSEC(YES). If you need to change any of these definitions, you can do so by copying them to another group. We recommend that you do **not** change the supplied definitions of any other transactions.

It is unlikely that you will want to give all users access to all of the transactions in this category; consider defining them in several subcategories. In the examples that follow, the category 2 transactions are further subdivided into a number of groups. Please note that these are only examples. You can choose to group CICS transactions in the ways that best suit your installation's needs.

- SYSADM, containing: CEDA, CEMT, CETR, and CRMF.
- DEVELOPER, containing: CEBR, CECI, CECS, CEDB, and CEDF
- INQUIRE, containing: CEDC
- OPERATOR, containing: CEOT, CEST, CMSG, and CWTO
- INTERCOM, containing: CEHP, CEHS, CPMI, CRTE, CSMI, CSM1, CSM2, CSM3, CSM5, and CVMI

  If function shipping is being used, the mirror transactions must be available to remote users in a function shipping environment. When a database or file resides on another CICS region, CICS function ships the request to access the data, and this request runs under one of the CICS-supplied mirror transactions. This means that:

  – The terminal user running the application must be authorized to use the mirror transaction. (See Chapter 5, "Transaction security" on page 41.)

  – The terminal user must also be authorized to use the data that the mirror transaction accesses. (See Chapter 6, "Resource security" on page 45.) The mirror transactions are supplied with RESSEC(YES) defined; so, even if the user's transaction specifies RESSEC(NO), the mirror transaction fails if the user is not authorized to access the data.

    If you do not use resource security checking, change the mirror transaction definitions to specify RESSEC(NO). Because the mirror transactions are an IBM-protected resource, first copy these definitions into your own groups and then change them. It is recommended that you define CSGM (or, if your installation does not use CSGM, whatever transaction is defined as GMTRAN) in a group of its own, because all users need access to it. If your installation uses CSGM as its "good morning" transaction, users who are not authorized to use CSGM will receive message DFHAC2002 when they attempt to use CICS. Also include your "goodnight" transaction in this group, if you defined one with the GNTRAN system initialization parameter

  With RESSEC(YES) and CMDSEC(YES) defined for these transactions, you must ensure that the users authorized to use the transactions are also authorized to access the CICS resources and commands that the transactions use.

If you protect a resource with a resource group profile, you should avoid protecting the same resource with another profile, because the way your ESM handles multiple profiles can have a performance impact, or it can be difficult to determine exactly which access authority applies to a particular user.Table 17 lists the category 2 transactions.

Table 17. Category 2 transactions

CSD group	Transaction	Program invoked	Description
DFHCONS	CWTO	DFHCWTO	Writes to console operator
DFHEDF	CEDF	DFHEDFP	Provides execution diagnostic facility
	CEBR	DFHEDFBR	Browse temporary storage
DFHFE	CSFE	DFHFEP	Tests Field Engineering terminal
DFHINTER	CECI	DFHECIP	CICS command interpreter
	CECS	DFHECSP	Checks CICS command syntax
DFHISC	CEHP	DFHCHS	Provides CICS OS/2 remote server mirror
	CEHS	DFHCHS	Provides CICS/VM remote server mirror
	CPMI	DFHMIRS	Provides CICS OS/2 LU6.2 mirror
	CRTE	DFHRTE	Provides start transaction routing session
	CRTX	N/A	Provides default dynamic routing transaction
	CSMI	DFHMIRS	Provides ISC mirror transaction
	CSM1	DFHMIRS	Provides ISC SYSMSG model
	CSM2	DFHMIRS	Provides ISC scheduler model
	CSM3	DFHMIRS	Provides ISC queue model
	CSM5	DFHMIRS	Provides ISC DL/I model
	CVMI	DFHMIRS	Provides LU6.2 synclevel 1 mirror
DFHMSWIT	CMSG	DFHMSP	Provides message switching
DFHOPER	CEMT	DFHEMTP	Processes master terminal command
	CEOT	DFHEOTP	Inquires on user's own terminal status
	CEST	DFHESTP	Processes supervisor terminal command
	CETR	DFHCETRA	Provides inquire and set trace options
DFHSPI	CEDA	DFHEDAP	Provides perform resource definition online—full
	CEDB	DFHEDAP	Provides perform resource definition online—restricted
	CEDC	DFHEDAP	Views resource definitions online
DFHVTAM	CSGM	DFHGMM	Provides CICS good morning message

Category 3 transactions

Category 3 transactions are either initiated by the terminal user or associated with a terminal. **All CICS terminal users**, whether they are signed on or not, require access to transactions in this category. For this reason, it is recommended that category 3 transactions are made exempt from any security check, and CICS permits any terminal user to initiate these transactions.

For category 3 transactions you are recommended to specify RESSEC(NO) and CMDSEC(NO) on the CICS transaction resource definition. These transactions may be defined to the ESM, for use by the QUERY SECURITY command, even if all users are permitted access.

Table 18 lists the category 3 transactions.

Table 18. Category 3 transactions

CSD group	Transaction	Program invoked	Description
DFHHARDC	CSPP	DFHP3270	Provides 3270 print support
DFHBMS	CSPG	DFHTPR	Provides BMS terminal paging
	CSPS	DFHTPS	Schedules BMS paging transaction
DFHISC	CLS1	DFHZLS1	Provides ISC LU services model
	CLS2	DFHLUP	Provides ISC LU services model
	CLS3	DFHCLS3	ISC LU services model
	CLS4	DFHCLS4	Manages password expiry
	CMPX	DFHMXP	Ships ISC local queuing
	CRSR	DFHCRS	Provides ISC remote scheduler
	CSSF	DFHRTC	Cancels CRTE transaction routing session
	CXRT	DFHCRT	Provides Transaction routing relay
DFHRSEND	CSRS	DFHZRSP	Synchronizes 3614 message
DFHSIGN	CESN	DFHSNP	Signs on terminal user
	CESF	DFHSNP	Signs off terminal user
	CEGN	DFHCEGN	Schedules goodnight transaction
DFHSPI	CATR	DFHZATR	Deletes autoinstall restart terminal
DFHSTAND	CQRY	DFHQRY	Provides ATI query support
	CSAC	DFHACP	Processes program abnormal condition
DFHVTAMP	CSCY	DFHCPY	Provides 3270 screen print
	CSPK	DFHPRK	Provides 3270 screen print support
	CSRK	DFHRKB	Provides 3270 screen print—release keyboard

Part 3. Intercommunication security

This part discusses how to plan and implement security in an intersystem communication (ISC) environment, using LU6.2 or LU6.1, or in a multiregion operation (MRO) environment. This part contains the following chapters:

- Chapter 11, “Overview of intercommunication security” on page 89, which introduces the concepts of bind-time, link, user, transaction, and resource security in an intercommunication environment
- Chapter 12, “Implementing LU6.2 security” on page 95, covering bind-time, link, user, transaction, resource, and command security; plus transaction routing, and function shipping
- Chapter 13, “APPC password expiration management” on page 115, which contains information on evaluating and using APPC password expiration management
- Chapter 14, “Implementing LU6.1 security” on page 137, covering link, transaction, resource, and command security; plus function shipping
- Chapter 15, “Implementing MRO security” on page 143, covering bind-time, link, user, transaction, resource, and command security; plus transaction routing and function shipping
- Chapter 16, “Security for shared data tables” on page 157, covering provision made for security of CICS shared data tables; plus logon security checks, and connection security checking for bind security and file security.
- Chapter 17, “Security for the Report Controller facility and CICS SPOOL interface” on page 161, discusses the necessity of using an ESM to secure reports and all facilities of the RCF.

Chapter 11. Overview of intercommunication security

This chapter gives an overview of how security works when CICS systems are interconnected or connected to other compatible systems.

It is organized under the following main topics:

- "Introduction"
- "Planning for intercommunication security" on page 90
- "Summary of intercommunication security levels" on page 92
- "Implementing intercommunication security" on page 92.

Introduction

In a single CICS system, you use security to make sure that terminal users can access only those parts of the system they need to work with. For interconnected systems, the same basic principles apply, but now you also include definitions for connections, sessions, and partners. You also need to allow for the fact that users of one CICS system can initiate transactions and access resources in another CICS system.

This chapter assumes that you are already familiar with setting up security for a single CICS system, as described in Part 1, "Introduction" on page 1 and Part 2, "Implementing protection for a single-region CICS" on page 17.

In particular, you should understand the following concepts:

- User signon. (See "Sign-on process" on page 31.)
- How the relationship between user security and transaction security determines which transactions a particular user is allowed to invoke. (See Chapter 4, "Verifying CICS users" on page 31 and Chapter 5, "Transaction security" on page 41.)
- How resource security determines which other resources a user is allowed to access. (See Chapter 6, "Resource security" on page 45.)

An interconnected group of CICS systems differs from a single CICS system in that you may have to define a user profile or group profile more than once. (See "ESM user definitions" on page 9, for information on defining these profiles.) That is, you may have to define these profiles in each CICS system that is using a separate ESM database, and in which a user is likely to want to attach a transaction or access a resource. When planning these profiles, you must consider all cases in which a user could initiate function shipping, transaction routing, asynchronous processing, distributed program link, distributed transaction processing, or external call interface (EXCI). (For descriptions of these methods of intercommunication, see the *CICS Intercommunication Guide* and the *CICS Distributed Transaction Programming Guide*.)

Planning for intercommunication security

Intercommunication security in a CICS system is concerned with incoming requests for access to CICS resources, rather than with requests that are sent to other systems.

The security problem with incoming requests occurs when a particular user at a particular remote system is trying to access resources of your CICS system. Is this access authorized, or should it be rejected?

The following sections describe the points in the processing of an incoming request at which you can apply security checks.

Bind-time security

The first requirement is for a session to be established between the two systems. This does not, of course, happen on every request; a session, once established, is usually long-lived. Also, the connection request that establishes the session can, depending on the circumstances, be issued either by the remote system or by your CICS system. However, the establishment of a session presents the first potential security exposure for your system.

Your security concern is to prevent unauthorized remote systems from being connected to your CICS system; that is, to ensure that the remote system is really the system that it claims to be. This level of security is called **bind-time security** (also known as **systems network architecture (SNA) session security**). It can be applied when a request to establish a session is received from, or sent to, a remote system.

Note: We use the term **bind** to refer both to the **SNA BIND** command that is used to establish SNA sessions between systems, and to the **CICS connection request** that is used to establish MRO sessions for CICS interregion communication.

You can specify bind-time security for LU6.2 and multiregion operation (MRO) links, but **not** for LU6.1 links. For information on defining bind-time security in your system, see either “Bind-time security with LU6.2” on page 95 or “Bind-time security with MRO” on page 143, depending on the environment you are using.

Link security

Each link between systems is given an authority defined by a userid.

It is important to note that users cannot access any transactions or resources over a link that is itself unauthorized to access them. This means that each user’s authorization is a subset of the link’s authority as a whole.

To limit the remote system’s access to your transactions and resources, you use **link security**. Link security is concerned with the single user profile that you assign to the remote system as a whole. Like user security in a single-system environment, link security governs:

- **Transaction security**. This controls the link’s authority to attach specific transactions.
- **Resource security**. This controls the link’s authority to access specific resources. This applies for transactions, executing on any of the sessions from

the remote system, that have RESSEC(YES) specified in their transaction definition.

- **Command security**. This controls the link's authority for the commands that the attached transaction issues. This applies for transactions, executing on any of the sessions from the remote system, that have CMDSEC(YES) specified in their transaction definition.
- **Surrogate user security**. This controls the link's authority to START transactions with a new userid, and to install resources with an associated userid.

For more information, see "Transaction, resource, command, and surrogate user security."

Link security with MRO

See the section "Link security with MRO" on page 145, in Chapter 15, "Implementing MRO security" on page 143.

Link security with LU6.2

See the section "Link security with LU6.2" on page 99, in Chapter 12, "Implementing LU6.2 security" on page 95.

Link security for LU6.1

See the section "Link security with LU6.1" on page 137, in Chapter 14, "Implementing LU6.1 security" on page 137.

User security

In addition to the security profile that you set up for the link, you may want to further restrict each remote user's access to the transactions, commands, and resources in your system. This is done by specifying the appropriate ATTACHSEC parameters in the CONNECTION definition. This **user security**, like link security, distinguishes between transaction, resource, command, and surrogate security. User security can never **increase** a user's authority above that of the link. For more information, see "Transaction, resource, command, and surrogate user security."

For information on defining user security in your system, see either "User security with LU6.2" on page 100 or "User security with MRO" on page 146, depending on the environment you are using.

You cannot specify user security for LU6.1 links. For LU6.1, the user security is taken to be the same as the link security.

Transaction, resource, command, and surrogate user security

The last step in defining security for your system is to make sure that the access parameters match the profiles you have defined for your transactions, resources, commands, and surrogate users for the link and the individual remote users. For information on defining these levels of security in a single-system environment, see Chapter 5, "Transaction security" on page 41, Chapter 6, "Resource security" on page 45, and Chapter 8, "CICS command security" on page 65.

Resources and commands are unsecured unless you explicitly request security protection in your transaction definitions.

For information on defining transaction and resource security in your system, see one of the following, depending on the environment you are using:

- “Transaction, resource, and command security with LU6.2” on page 107
- “Transaction, resource, and command security with LU6.1” on page 138
- “Transaction, resource, and command security with MRO” on page 149

Summary of intercommunication security levels

Figure 4 on page 93 shows bind-time, transaction, resource, and command security, and how CICS enforces these levels of security under the LU6.2, MRO, and LU6.1 protocols. It also shows how the two levels of authorization (user and link) are involved at the three security levels.

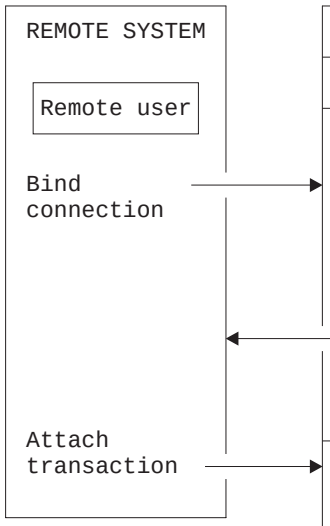
For guidance on choosing between these environments, see the *CICS Intercommunication Guide*.

Implementing intercommunication security

Security in the intercommunication environment is implemented through resource definition and ESM profiles. The following chapters tell you how to define your intersystem links, according to the environment you are using:

- Chapter 12, “Implementing LU6.2 security” on page 95
- Chapter 14, “Implementing LU6.1 security” on page 137
- Chapter 15, “Implementing MRO security” on page 143

Figure 4 on page 93 shows a summary of intercommunication security.



SECURE CICS SYSTEM	
Security level	Enforced by
Bind-time security (BIND received)	
1. Accept bind request?	LU6.1 No check. LU6.2 Session key from the ESM. MRO DFHAPPL profiles in the ESM
Bind-time security (BIND sent)	
1. Is the remote system the correct one?	LU6.1 No check. LU6.2 Session key from the ESM. MRO DFHAPPL profiles in the ESM.
Transaction security:	
1. Has the link authority to attach transaction?	Link authority established just after session is bound, by sign-on (by either SECURITYNAME on the connection (not for MRO), or USERID on session definition).
2. Has the remote user authority to access this system?	LU6.1 No check. LU6.2 User authority of remote user established at signon time. MRO AS LU6.2
3. Has the remote user authority to attach transaction?	LU6.1 Link authority. LU6.2 User authority established at this attach request (or possibly at earlier attach request from same user) by sign-on. MRO As LU6.2.
Resource, command and surrogate security:	
1. Has the session authority to access other resources that the transaction uses?	Link authority established just after session is bound, by ESM sign-on of userid specified in SECURITYNAME of link definition.
2. Has the remote user authority to access other resources that the transaction uses?	LU6.1 Link authority. LU6.2 User authority established at this attach request (or possibly at earlier attach request from same user) by ESM sign-on. MRO As LU6.2.

Figure 4. Summary of intersystem and interregion security

Chapter 12. Implementing LU6.2 security

This chapter tells you how to implement security for LU6.2. It is organized under the following topics:

- "Bind-time security with LU6.2"
- "Link security with LU6.2" on page 99
- "User security with LU6.2" on page 100
- "SNA profiles and attach-time security" on page 105
- "Attach-time security and the USEDFLTUSER option" on page 106
- "Transaction, resource, and command security with LU6.2" on page 107
- "Transaction routing security with LU6.2" on page 108
- "Function shipping security with LU6.2" on page 109
- "Distributed program link security with LU6.2" on page 110
- "Security checking done in AOR with LU6.2" on page 112
- "Summary of resource definition options for LU6.2 security" on page 114.

Bind-time security with LU6.2

A security check can be applied when a request to establish an APPC session is received from, or sent to, a remote system; that is, when the session is bound. This is called **bind-time security** (or, in SNA terms, **session security**), and is part of the CICS implementation of the LU6.2 architecture. Its purpose is to prevent an unauthorized system from binding a session to one of your CICS systems.

Bind-time security is optional in the LU6.2 architecture. Of course, do not specify bind-time security if the remote system does not support it. SNA defines how session security is to be applied, and CICS conforms to this architecture. If you want to connect to a CICS system other than CICS Transaction Server for VSE/ESA or CICS/VSE, make sure the other system is also compatible with this architecture.

When you define an LU6.2 connection to a remote system, you assume that all inbound bind requests originate in that remote system, and that all outbound bind requests are routed to the same system. However, where there is a possibility that a transmission line might be switched or broken into, guard against unauthorized session binds by specifying session security at both ends of the connection.

For a bind request to succeed, both ends must hold the same **session key**, which is defined to the ESM. When a session is bound, the action CICS takes depends on:

- How you specified the SEC and XAPPC system initialization parameters.
- How you specified the BINDSECURITY option on the CONNECTION resource definition in the CSD. The CONNECTION definitions on both systems must specify the same value for BINDSECURITY.
- Whether you have defined an APPCLU security profile for the link.

If you have specified the SEC=YES and XAPPC=YES system initialization parameters, and BINDSECURITY(YES) in your CSD connection definition, and BINDSECURITY(YES) is also specified for the partner system, a bind security validation will be attempted.

If you have BINDSECURITY(NO), then the system initialization parameter specifications are immaterial.

Table 19 summarizes what happens.

Table 19. APPC bind-time security—relationship to resource definition

SEC value	**XAPPC value**	**BINDSECURITY value**	**ESM APPCLU profile**	**Resulting CICS action**
YES	YES	YES	Defined (See note 1)	CICS extracts the APPCLU profile from the ESM at bind-time to verify the remote system.
YES	YES	YES	Not defined	CICS is unable to extract the APPCLU profile from the ESM and therefore rejects the bind.
YES	YES	NO	Any value	CICS defaults to its own internal checking and compares the BINDPASSWORDs. If these match, or are not specified, the bind is successful (See note 2)
YES	NO	Any value	Any value	CICS defaults to its own internal checking and compares the BINDPASSWORDs. If these match, or are not specified, the bind is successful (See note 2)
NO	Any value	Any value	Any value	CICS does not perform any bind security validation.

Notes:

1. If the ESM APPCLU profile is defined, but the SESSKEY on each side does not match, or the session segment is locked or expired, and no value is specified for SESSKEY, the bind request is always rejected.
2. For more information about CICS internal checking, see “Internal LU6.2 bind time security” on page 98.

For an example of defining an APPCLU profile, see your ESM documentation.

Defining bind-time security

You define bind-time security in the CONNECTION definition, although you must also choose the appropriate system initialization parameters. Figure 5 shows how to define APPC external session security, for which you need to specify the BINDSECURITY option.

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

Note: For APPC terminals defined as a TERMINAL-TYPETERM pair, the BINDSECURITY operand is on the TERMINAL definition.

Figure 5. Bind-time security

Auditing bind-time security

If security is active (SEC=YES is specified in the system initialization parameters), CICS performs bind security auditing. The following conditions are considered bind failures:

- Session key does not match partner's.
- Session segment is locked.
- Session segment has expired.
- Session key is null.
- Session segment does not exist.
- Session segment retrieval was unsuccessful.
- Session bind was unsuccessful.

The following conditions are considered bind successes:

- Session was successfully bound.
- Session key will expire in less than six days.

Changing ESM profiles that are in use—caution

Take care when changing ESM profiles for APPC connections that are in use. CICS recognizes the change in the profile after the ESM database has been refreshed Bind-time security processing occurs when each session in a connection is acquired. If not all the sessions in a connection are acquired and the APPC profile becomes invalid, then an attempt to establish any of the unacquired sessions causes a bind security failure. This can cause transactions that attempt to allocate one of these unused sessions to be suspended indefinitely.

Reasons for invalid profiles

An APPC profile can become invalid for a number of reasons; for example:

- The session key expires
- The session key changes and an ESM database refresh takes place in one system without the corresponding change and refresh occurring in the other system
- The profile is locked while an ESM database refresh takes place.

Sessions that are already acquired still continue to function normally if bind security fails in another session. If you are using expiring session keys, then the connection can still be used after the expiry date, if any of the sessions on the connection were acquired before the date of expiration, and have remained acquired. Hence, you see the effect of an expiring session key only when the connection (or session) is acquired.

Note: If your ESM allows for expiring session keys, see the associated documentation for methods of finding out when these keys are due to expire. You must also take appropriate action to minimise any disruption that may occur because the connection is unavailable because of an expired sessionkey.

You can avoid the problem of APPC profiles becoming invalid while the connection is in use by specifying AUTOCONNECT(YES) or AUTOCONNECT(ALL) on the SESSIONS definition. This causes all sessions to be established (acquired) when the connection is acquired.

Internal LU6.2 bind time security

You define LU6.2 internal (CICS) bind-time security using the CONNECTION definition. Figure 6 shows an example of this using CEDA.

```
CEDA DEFINE CONNECTION(name)
  GROUP(groupname)
  ACCESSMETHOD(VTAM)
  SECURITYNAME(name)
  PROTOCOL(APPC)
  NETNAME(name)
  BINDPASSWORD(password)
  BINDSECURITY(NO)
```

Note: For APPC terminals defined as a TERMINAL-TYPETERM pair, the BINDPASSWORD operand is on the TERMINAL definition.

Figure 6. Internal (CICS) bind-time security

The BINDPASSWORD on an RDO CONNECTION resource definition is only used for LU6.2 bind-time security validation, when ESM bind security validation has not been requested. That is, when XAPPC=NO has been specified in the system initialization parameters, or BINDSECURITY(NO) has been specified on the RDO CONNECTION resource definition.

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

Each pair of communicating systems must have the same password for the link between them. For example, if you are defining a link between two CICS systems, the CONNECTION definition in each system must specify the same password. You are advised to select a unique password for each system with which your CICS system communicates, rather than using the same password for two or more remote systems.

Specifying a bind password causes CICS to perform password checking each time a session is bound. CICS rejects bind requests from systems that do not know the password. It also prevents your bind requests from being intercepted and accepted by systems that do not know your password.

If the two bind passwords do not match, a message is sent to the transient data queue, CSNE. The session is not bound, and the system reacts to a user request for a session with a **SYSIDERR**.

Link security with LU6.2

Link security further restricts the resources a user can access, depending on the remote system from which they are accessed. The practical effect of link security is to prevent a remote user from attaching a transaction or accessing a resource for which the link userid has no authority.

Link security can be associated with a connection or with a session, depending on whether you want to control the link security for each group of sessions separately:

- To define link security for a connection as a whole, specify the SECURITYNAME parameter in the CONNECTION definition.
- To define link security for individual groups of sessions within a connection, specify the USERID in the SESSIONS definition as a user identifier.

Each link between systems is given an authority defined by a link userid. A link userid for LU6.2 is a userid defined on your session's definition for this connection. If not defined, the link userid is the SECURITYNAME userid specified on the connection definition. If there is no SECURITYNAME, the link userid is the default userid.

You can never transaction route or function ship to CICS without having at least one security check, but the security checks are minimized if the two regions involved are ***equivalent systems***. This term means the same thing for LU6.1, LU6.2, and MRO. If the link userid matches the local region userid, you have equivalent systems.

If you do have equivalent systems, only one security check is made. This will either be against the default user (for ATTACHSEC=LOCAL) or against the userid that is in the received FMH-5 attach request (ATTACHSEC=non-LOCAL).

If you do not have equivalent systems for ATTACHSEC=LOCAL, resource checks are done only against the link userid. For ATTACHSEC=non-LOCAL there are always two resource checks. One is against the link userid, and the second is against the userid received from the remote user in the attach request.

If a failure occurs in establishing link security, the link is given the security of the local region's default user. This would happen, for example, if the preset session userid had been revoked.

If a value is present on the USERID parameter of the SESSIONS definition, the value overrides any value specified on the SECURITYNAME parameter in the CONNECTION definition.

User security with LU6.2

User security causes a second check to be made against a user signed onto a terminal, in addition to the link security described in “Link security with LU6.2” on page 99. After reading the following descriptions, consider whether you want the extra level of security checking that user security provides.

You can specify the following levels of user security using the ATTACHSEC parameter of the CONNECTION definition:

- *LOCAL*, which you specify if you do **not** want to make a further check on users by requiring a user identifier or password to be sent. Choose LOCAL if you do not want user security because you consider that the authority of the link is sufficient for your system. See “Specifying user security in link definitions” on page 102 for information on doing so.
- *Non-LOCAL*, which you specify if you *do* want to make a further check on users by requiring a user identifier, or a user identifier and a password, to be sent. Non-LOCAL includes the following types of checking:
  - IDENTIFY

    A user identifier must be sent, but no password is requested
  - VERIFY

    A password must also always be sent
  - PERSISTENT VERIFICATION

    Password is sent on the first attach request for a user
  - MIXIDPE

    Either identify or persistent verification

Note: “Non-LOCAL user security verification” further describes these types of user checking checking. See “Specifying user security in link definitions” on page 102 for information on specifying them.

Non-LOCAL user security verification

In a CICS-to-CICS system connection, where you have a terminal-owning region (TOR), an application-owning region (AOR), and a data-owning region (DOR), the terminal operator signs on to the TOR, attaches a transaction in the AOR, and accesses resources in the DOR. If all three systems implement non-local user security, then the same operator is registered as a user in each of them. The usual procedure is for the operator to sign on to the TOR with a password. CICS assumes that the password is valid for the entire systems complex, and that it does not need to be passed on to the AOR and the DOR for further verification. All that is needed is for the AOR and the DOR to IDENTIFY the user, who is then signed on without a password. Therefore, the password is not sent with the attach request to the AOR. This is considered to be more secure, because the password is not passed on a network.

Specify IDENTIFY when you know that CICS can trust the remote system to verify its users (by some sort of sign-on mechanism) before letting them use the link. Use IDENTIFY if you want user security for CICS-to-CICS communication (CICS does not support password flows on CICS-to-CICS connection) which includes the following:

- CICS/VSE Version 2 Release 3
- CICS Transaction Server for VSE/ESA Version 1 Release 1
- CICS Transaction Server for OS/390

If the front end does not have a security manager—for example, if it is a programmable workstation (PWS)—it is often not possible to VERIFY the user by means of a user identifier and password before the attach request reaches the AOR. The AOR must then receive both user identifier and password from the front end so that it can verify the user itself by a sign-on with password.

Specify VERIFY if you have reasons for wanting your own system to verify the remote system's users even if they have already been checked by the remote system itself, or if the remote system does not have a security manager and therefore cannot verify its own users. VERIFY must be used if the request comes from CICS for OS/2, which does not support PERSISTENT.

If programmable workstations make repeated requests to attach transactions in the AOR, performance suffers because of many verifications. The LU6.2 architecture, which defines these security procedures, allows persistent verification to reduce the software overhead. Using this protocol, the first attach request contains a user identifier and a password, but once the user has signed on, only the user identifier is needed for all the attach requests that follow.

Specify PERSISTENT to reduce the verification overhead if remote users repeatedly send attach requests. However, the remote system must be able to cooperate in the management of persistent verification by keeping a list of users who are currently signed on.

Some remote APPC systems have mixed sign-on requirements that vary from conversation to conversation (for example, CPI communications conversations). In this case, CICS must accept incoming identify or persistent requests.

To decide which of these types of user verification to use, you need to know how far the remote system is capable of managing its own security and, if it cannot, to what extent it must depend on the CICS system you are defining.

- Do you need to know the user identifier? If not, use LOCAL.
- Can the remote system verify its own users? If so, use IDENTIFY. If not, can it send a user identifier and a password with the attach request? If so, use VERIFY for PWS-to-CICS communication.
- Does the remote system support persistent verification by keeping track of its user identifiers and passwords? If you are using PWS-to-CICS communication, you may want to specify PERSISTENT, or MIXIDPE if you are using both CICS-to-CICS and PWS-to-CICS.

You specify these levels of checking for each connection using the ATTACHSEC operand of the RDO CONNECTION resource definition, as described in "Specifying user security in link definitions" on page 102.

Specifying user security in link definitions

The level of user security you require for a remote system is specified in the ATTACHSEC operand in the RDO CONNECTION resource definition, as shown in Figure 7.

This topic describes how CICS interprets the parameters of the ATTACHSEC operand of the CONNECTION definition. However, special rules apply for CICS transaction routing, as described in “Transaction routing security with LU6.2” on page 108. Figure 7 shows an example of defining ATTACHSEC using CEDA.

```
CEDA DEFINE CONNECTION(name)
  GROUP(groupname)

  ATTACHSEC(LOCAL|IDENTIFY|VERIFY|PERSISTENT|MIXIDPE)
```

Note: For APPC terminals defined as a TERMINAL-TYPETERM pair, the ATTACHSEC operand is on the TERMINAL definition.

Figure 7. Defining sign-on level for user security

The ATTACHSEC operand specifies the sign-on requirements for incoming transaction attach requests. It has no effect on attach requests that are issued by your system to a remote system; these are dealt with in the remote system.

When an APPC session is bound, each side tells the other the level of attach security user verification that will be performed on its incoming requests. There is no negotiation on this.

Meanings of ATTACHSEC operand

The following are the possible operands of ATTACHSEC:

LOCAL
specifies that a user identifier is not to be supplied by the remote system. If one is received, the attach fails. CICS makes the user security profile equivalent to the link security profile. You do not need to specify ESM profiles for the remote users. LOCAL is the default value.

IDENTIFY
specifies that a user identifier is expected on every attach request. All remote users of a system must be identified to the ESM.

If an attach request with both a user identifier and a password is received on a link with ATTACHSEC(IDENTIFY), CICS does not reject the attach request. CICS handles the attach request as if the connection was defined with ATTACHSEC(VERIFY).

If a null (X'00') user identifier or an unknown user identifier is received, CICS rejects the attach request.

VERIFY
specifies that, in addition to a user identifier, a user password is required for verification against the local ESM database. All remote users of a system must be identified to ESM.

The rules that apply to the checking of the user identifier for ATTACHSEC(IDENTIFY) also apply for ATTACHSEC(VERIFY). If a valid user identifier is received but the password verification fails, CICS rejects the attach request.

All CICS systems except CICS OS/2 and CICS for Windows NT can verify the security attributes of their users with an external security manager. CICS OS/2 does not have an external security manager and so is regarded as an insecure system. CICS OS/2 only supports LOCAL(VERIFY). If CICS for OS/2 is the terminal-owning region (TOR) connected to CICS Transaction Server for VSE/ESA, use the ATTACHSEC=VERIFY option in the LU6.2 connection definition on the CICS Transaction Server for VSE/ESA application owning region (AOR). The appropriate adjustments should also be made to the Communications Manager on the CICS OS/2 system so that the password and userid of the user signing on to CICS OS/2 are sent. (See the *CICS OS/2 Intercommunication Guide*, SC33-0826, for details of the Communications Manager changes that need to be made.) CICS Transaction Server for VSE/ESA is then able to VERIFY the user by performing a signon with password. If the communicating system is CICS for AIX, ATTACHSEC=IDENTIFY should be used.

Note: Products other than CICS can connect to a CICS Transaction Server for VSE/ESA AOR via an LU6.2 link. They then use the SNA LU6.2 FMH-5 ATTACH mechanism to start a transaction on the CICS AOR. Where this mechanism is being used from an insecure system, the ATTACHSEC=VERIFY option should be used on the connection definition to protect the transaction on the AOR. (See “SNA profiles and attach-time security” on page 105).

PERSISTENT
specifies that a user identifier and a user password are required with the first attach request for a new user, but all following attach requests for the same user need supply only a user identifier. (All remote users of a system must be identified to the ESM.) The first attach signs on the user, even if the attach request is later unsuccessful because the user is not authorized to attach the transaction.

Note: PERSISTENT cannot be used for CICS-to-CICS communication.

MIXIDPE
specifies that the sign-on level for the remote user is determined by parameters sent with the attach request. The possibilities are: PERSISTENT or IDENTIFY.

Sign-on status

With the ATTACHSEC parameters IDENTIFY, MIXIDPE, PERSISTENT, and VERIFY, the remote user remains signed on after the conversation associated with the first attach request is complete. CICS then accepts attach requests from the same user without a new sign-on until either of the following occurs:

- The period specified in the system initialization parameter USRDELAY elapses after completion of the last transaction associated with the attach request for this user.

  When you are running remote transactions over ISC and IRC links USRDELAY defines the time for which entries can remain signed onto the remote CICS region. For information on specifying USRDELAY, see the *CICS System Definition Guide*. See the *CICS Performance Guide* for information on tuning.

- The CICS system is terminated.

If you alter the ESM profile of a signed-on remote user (for example, by revoking the user), CICS continues to use the authorization established at the first attach request until the entry is removed from the sign-on list.

Password checking

If you are using ATTACHSEC(PERSISTENT) (or ATTACHSEC(MIXIDPE) being treated as ATTACHSEC(PERSISTENT)), CICS maintains a table for each remote system called the **persistent verification (PV) signed-on-from list**. This is a list of users whose passwords have been checked and who do not require a further password check as long as the entry remains in the list. Entries remain in the list until:

- The period specified in the system initialization parameter PVDELAY elapses after the user's sign-on entry was last used.

  PVDELAY defines how long entries can remain in the PV signed-on-from list for the remote system, which means that their passwords do not need to be revalidated for each attach request. For information on specifying a value for PVDELAY, see the *CICS System Definition Guide*. See the *CICS Performance Guide* for information on tuning.

- The connection with this system is terminated because: CICS is restarted, the connection is lost, or CICS receives an invalid attach request from the user.

When persistent verification is in operation for a remote user, and that user is removed from the PV signed-on-from list, CICS informs the remote system by issuing a sign-off request for the user to remove the entry from the PV signed-on-to list in the remote system.

If you specify ATTACHSEC(VERIFY), the remote user's password is checked for *every attach request*. This is to ensure that the user has authority to access this system, to verify that this password is correct, and to establish security authorities for the user.

Information about remote users

Information about the user can be transmitted with the attach request from the remote system. This means that you can protect your resources not only on the basis of which remote system is making the request, but also on the basis of which user at the remote system is making the request.

This topic describes some of the concepts associated with remote-user security, and how CICS sends and receives user information.

You have to define your users to the ESM. If a remote user is not defined to the ESM, any attach requests from that remote user are rejected.

CICS remote-user security for LU6.2 links implements the LU6.2 architecture. The LU6.2 architecture allows user identifiers, user passwords, and user profiles to be transmitted with requests to attach a transaction.

User profiles can be transmitted instead of, or in addition to, user identifiers. The profile name, if supplied, is treated as the group ID.

If the user has been added to the front-end system with a group ID explicitly specified, (for example, in EXEC CICS SIGNON or by filling in the GROUPID

parameter on the CESN panel) this will be propagated by CICS in outbound attach FMHs for LU6.2 links when ATTACHSEC(IDENTIFY) has been specified in the CONNECTION definition.

If the group ID defaults when the user was originally added to the front-end system, the profile field will not be included in the outbound FMH5. If the group ID is passed to the back-end system, the group ID will be used as part of ADD_USER proceeding on the back-end. The user ID must be defined as a member of the group passed in the ESM on the back-end for the ADD_USER to be successful.

CICS sends user IDs on ATTACHSEC(IDENTIFY) conversations. Table 20 shows how CICS decides which user ID to send.

Table 20. Attach-time user identifiers—LU6.2

Characteristics of the local task	**User identifier sent by CICS to the remote system**
Task with associated terminal—user identifier	Terminal user identifier
Task with associated terminal—no user signed on and no USERID specified in the terminal definition	Default user identifier for the local system
Task with no associated terminal or USERID started by interval control START command (if using function shipping or distributed transaction processing (DTP))	User identifier for the task that issued the START command
Task started with USERID option	User identifier specified on the START command
CICS internal system task	CICS region userid
Task with no associated terminal started by transient data trigger	User identifier specified on the DCT that defines the queue
Task with associated terminal started by transient data trigger	Terminal user identifier
Task started from PLTPI	PLTPIUSR

Signing on the remote user has two purposes:

- To ensure that the remote user is allowed to access the CICS system
- If the sign-on is successful, to establish the authority for the remote user

CICS signs off the remote user under the circumstances described in “Sign-on status” on page 103.

SNA profiles and attach-time security

Implementation of the LU6.2 attach-time security in CICS Transaction Server for VSE/ESA Release 1 conforms strictly to the architecture. In particular, note the following:

- The introduction of SNA profile support and the conformance to SNA attach-time security processing may cause migration problems.

- Profile support means that badly coded profiles sent in an attach FMH-5 cause certain attach requests to be rejected.
- The checks to prevent problems in the access security subfields of an FMH-5 are:
  - Check for unrecognized subfield
  - Check for invalid length subfield
  - Check for multiple subfields of the same type
- The full 10-character userid and password are accepted. Any trailing blanks (X'40') are removed before being passed to the security manager, which either rejects the attach request, or converts the userid and password into 8-character form before proceeding.
- Attach requests are rejected if they have a blank or zero-length user ID parameter in the attach FMH-5. See “Attach-time security and the USEDFLTUSER option” for an explanation of the exception where zero-length user IDs may be accepted for ATTACHSEC(VERIFY) and ATTACHSEC(IDENTIFY).
- Attach requests are rejected if they have a blank, or zero-length, userid parameter in the attach FMH-5.
- Valid SNA profiles received are treated as the ESM groupid with which the userid in the FMH-5 will be associated after the userid in the FMH-5 is signed on.
- When a SNA profile is received and the connection had ATTACHSEC=PERSISTENT, it is validated to conform to the architecture. It is not used to further qualify users in the signed-on-from list. This also applies to persistent signed-on flows received on a connection that has ATTACHSEC=MIXIDPE specified.

Attach-time security and the USEDFLTUSER option

In releases earlier than CICS Transaction Server for VSE/ESA, Release 1, a user who was not signed on would not have an associated userid. In CICS Transaction Server for VSE/ESA, coding USEDFLTUSER on the connection indicates that the default user can be used. The following types of incoming attach FMH-5 are accepted by CICS Transaction Server for VSE/ESA only if the USEDFLTUSER option is coded on the connection:

- An FMH-5 received on a connection defined with ATTACHSEC(IDENTITY) not containing a user ID subfield, for example, from a CICS for VSE/ESA system.
- An FMH-5 received on a connection defined with ATTACHSEC(VERIFY) containing userid and password subfields which have zero-length; for example, from certain non-EBCDIC based systems.
- An FMH-5 received on a connection defined with ATTACHSEC(VERIFY) containing an access security information field (ASIF) length field of zero.
- An FMH-5 received on a connection defined with ATTACHSEC(IDENTIFY) containing a userid access security information subfield (ASIS) which specifies a zero-length userid.
- An FMH-5 received on a connection defined with ATTACHSEC(IDENTIFY) containing a user ID ASIS which specifies a zero-length user ID.

If the user does not specify the USEDFLTUSER option in these exceptions, the expected protocol violation occurs, a message is generated, and the attach fails.

Transaction, resource, and command security with LU6.2

As in a single-system environment, users must be authorized to:

- Attach a transaction (**transaction security**)
- Access all the resources that the transaction is programmed to use. These levels are called **resource security**, **surrogate user security**, and **command security**

Transaction security

As in a single-system environment, the security requirements of a transaction are specified when the transaction is defined, as described in Chapter 5, “Transaction security” on page 41.

In an LU6.2 environment, two basic security requirements must be met before a transaction can be initiated:

- The link must have sufficient authority to initiate the transaction.
- If anything other than ATTACHSEC(LOCAL) has been specified, user security is in force. The user who is making the request must therefore have sufficient authority to access the system and to initiate the transaction.

Note: Transaction security also applies to the mirror transactions. See “Function shipping security with LU6.2” on page 109.

Resource and command security

Resource and command security in an intercommunication environment are handled in much the same way as in a single-system environment.

Resource and command security checking are performed only if the installed TRANSACTION definition specifies that they are required; for example, on the CEDA DEFINE TRANSACTION command, as shown in Figure 8.

```
CEDA DEFINE TRANSACTION
  .
  RESSEC(YES)
  CMDSEC(YES)
  .
```

Figure 8. Specifying resource and command security for transactions

If a TRANSACTION definition specifies resource security checking, using RESSEC(YES), both the link and the user must also have sufficient authority for the resources that the attached transaction accesses.

If a TRANSACTION definition specifies command security checking, using CMDSEC(YES), both the link and the user must also have sufficient authority for

the SP commands shown in Table 12 on page 66 that the attached transaction issues.

For further guidance on specifying resource and command security, see Chapter 6, “Resource security” on page 45 and Chapter 8, “CICS command security” on page 65.

NOTAUTH exceptional condition

If a transaction tries to access a resource, but fails the resource security checks, the NOTAUTH condition occurs.

When the transaction is the CICS mirror transaction, the NOTAUTH condition is returned to the requesting transaction, where it can be handled in the usual way.

Transaction routing security with LU6.2

In transaction routing, the authority of a user to access a transaction can be tested in both the TOR and the AOR.

In the TOR, a test is made to ensure that the user has authority to access the transaction defined as remote, just as if it were a local transaction. This test determines whether the user is allowed to run the relay program.

The terminal through which the transaction is invoked must be defined on the remote system (or defined as “shippable” in the local system), and the terminal operator needs authority if the remote system is protected. The way in which the terminal on the remote system is defined affects the way in which user security is applied:

- If the definition of the remote terminal does not specify the USERID parameter:
  - For links defined with ATTACHSEC(IDENTIFY), the transaction security and resource security of the user are established when the remote user is signed on. The userid under which the user is signed on, whether explicitly or implicitly (in the DFLTUSER system initialization parameter), has this security capability assigned in the remote system.
  - For links defined with ATTACHSEC(LOCAL), transaction security, command security, and resource security are limited by the authority of the link.

In both cases, tests against the link security are made as described in “Link security with LU6.2” on page 99.

Note: During transaction routing, the 3-character operator identifier from the TOR is transferred to the surrogate terminal entry in the AOR. If the surrogate terminal was shipped in, this identifier is not used for security purposes, but it may be referred to in messages.

Preset-security terminals and transaction routing

A terminal has preset-security if a value is specified on the USERID parameter of the terminal definition. When considering the security aspects of transaction routing from a preset-security terminal, you must remember that preset-security is an attribute of the terminal rather than of the user who initiated the transaction routing request.

During transaction routing, a surrogate terminal is created in the AOR to represent the terminal at which the transaction routing request was issued. Whether the surrogate terminal has preset security or not depends upon a number of factors:

- If a remote terminal definition exists in the AOR for the terminal at the TOR, and specifies the USERID parameter, the surrogate terminal is preset with this userid. If the USERID parameter is not specified in the remote terminal definition, the surrogate terminal does not have preset security.
- If a remote terminal definition does not exist in the AOR, the preset security characteristics of the surrogate terminal are determined from the terminal definition shipped from the TOR. If the shipped terminal definition has preset security, the surrogate also has preset security, unless the connection to the AOR is defined with ATTACHSEC=LOCAL, in which case any preset security information shipped to the AOR is ignored.

CICS routing transaction, CRTE

You can use the CICS routing transaction, CRTE, with LU6.2 to run transactions that reside on a connected remote system, instead of defining these transactions as remote in the local system. CRTE is particularly useful for infrequently used transactions, or for transactions such as CEMT that reside on all systems.

Security checking done in the AOR for transactions executed under CRTE does not depend on what is specified by ATTACHSEC, or on the userid signed on in the TOR. Instead, security checking depends on whether the user signs on while using CRTE:

- If the user does **not** sign on, the surrogate terminal created is associated with the AOR default user. When a transaction is run, the security checks are carried out against this default user. A check is also done against the link userid to see whether the routing application itself has authority to access the resource.
- When a user **does** sign on to the AOR, using the CESN transaction while running CRTE, the surrogate already created then points to the userid of the signed-on user. For transactions attempting to access resources, security checking is done against the signed-on user's userid in the surrogate and the link userid.

For more information on CRTE, see the *CICS-Supplied Transactions* manual and the *CICS Intercommunication Guide*.

Function shipping security with LU6.2

When CICS receives a function-shipped request, the transaction that is invoked is the **mirror transaction**. The CICS-supplied definitions of the mirror transactions all specify resource, but not command, security checking. This means that you are prevented from accessing the remote resources if either the link or your userid profile on the other system does not have the necessary authority.

If the CICS-supplied definitions of the mirror transactions are not what your security strategy needs, you can change them by copying the definitions in group DFHISC into your own group, changing them and then reinstalling them. For more information, see "Category 2 transactions" on page 83.

If you include a remote resource in your resource definitions, you can arrange for security checking to be done locally, just as if the resource were a local one. Also, the system that owns the resource can be made to apply an independent check, if it is able to receive the user identifier. You can therefore choose to apply security restrictions on both sides, on either side, or not at all.

Note: Be aware that if you specify the SYSID option on a function-shipped request, security checking is done in the remote system but is **bypassed in the local system**. Figure 9 summarizes what happens.

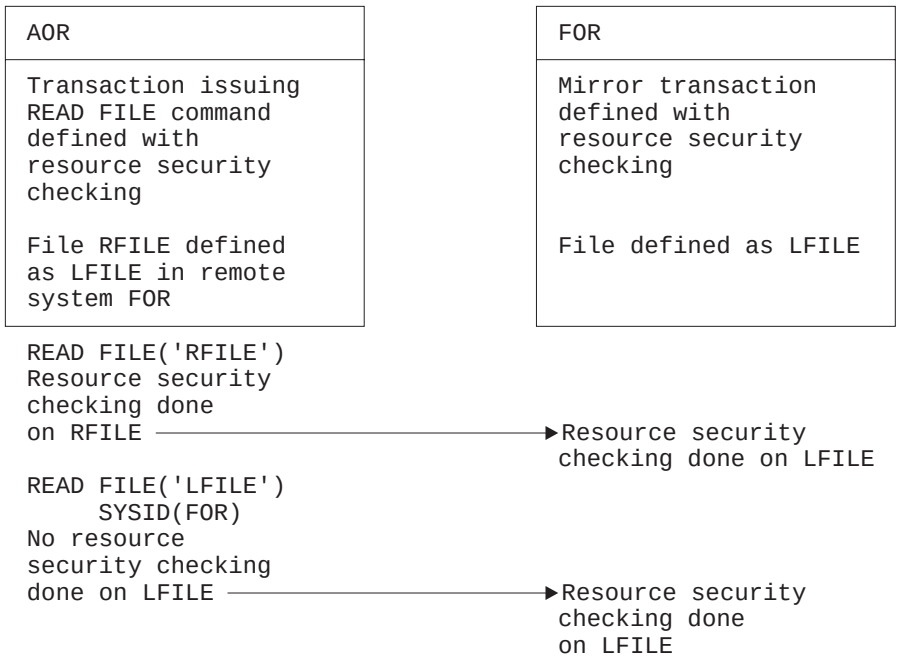


Figure 9. Security checking done with and without SYSID

For programming information on specifying the SYSID option, see the *CICS Application Programming Reference* manual.

Distributed program link security with LU6.2

The CICS distributed program link (DPL) facility enables a CICS program (the client program) to call another CICS program (the server program) in a remote CICS region. DPL is used when the SYSID option on the EXEC CICS LINK PROGRAM command, or the REMOTESYSTEM option of the program resource definition, specifies a remote CICS region.

When the SYSID option on the EXEC CICS LINK command specifies a remote CICS system, the client region does not perform any resource security checking, but leaves the resource check to be performed in the server region.

The server program in the remote region is executed by a mirror transaction, in a similar way to other function-shipped CICS requests. However, the transaction name associated with the mirror depends on how the EXEC CICS LINK PROGRAM is invoked in the client region. You must be aware of the transaction name because normal attach security applies to the mirror transaction:

- If the TRANSID option is specified on the DPL command, the specified transaction name is used for the mirror.
- If the TRANSID option is omitted from the DPL command, but the TRANSID option is used in the program resource definition in the client region, the name for the mirror is taken from the program's TRANSID specification.

Otherwise, a default name for the mirror transaction is used, and this depends on the origin and LU6.2 sync level of the conversation:

- If the client program is executing in a CICS OS/2 system, the transaction name for the mirror is **CPMI**.
- If synclevel 1 is being used, the default transaction name for the mirror is **CVMI**. This transaction name is used:
  - If the SYNCONRETURN option is specified on the DPL command in the client region
  - If the LU6.2 CONNECTION definition specifies SINGLESESS(YES)
  - If the connection is by means of an LU6.2 terminal; that is, a terminal whose resource definition uses a TYPETERM with a specification of DEVICE(APPC)
- If sync level 2 is being used, the default transaction name is **CSMI**. This transaction name is used when none of the other previous conditions is met.

Authorize your users to access the transaction name that the mirror runs under. The userids to be authorized depend on whether LOCAL or non-LOCAL attach security is being used, and are described in "Security checking done in AOR with LU6.2" on page 112. If the mirror transaction is defined with RESSEC(YES) in the server region, these userids must also be authorized to access the server program that is being linked to by the mirror. If the server program accesses any CICS resources, the same userids must be authorized to access them. If the server program invokes any SP-type commands, and the mirror transaction is defined with CMDSEC(YES) in the server region, the same userids must be authorized to access the commands.

If the mirror transaction cannot be attached because of security reasons, the NOTAUTH condition is not raised, but the TERMERR condition is returned to the issuing application in the client region. If the mirror transaction is successfully attached, but it is not authorized to link to the distributed program in the server region, the NOTAUTH condition is raised. The NOTAUTH condition is also raised if the server program fails to access any CICS resources for security reasons.

The server program is restricted to a DPL-subset of the CICS API commands when running in a server region. The commands that are not supported include some that return security-related information. For programming information about which commands are restricted, see the *CICS Application Programming Reference*. For information about surrogate user checking on DPL calls, see "Userid passed as parameter on EXCI calls" on page 61. For further information about DPL, refer to the *CICS Intercommunication Guide*.

Security checking done in AOR with LU6.2

This section summarizes how security checking is done in the AOR depending on how SECURITYNAME is specified in the AOR and TOR.

The link userid referred to in Table 21 on page 113 and Table 22 on page 114 is the one specified in the SECURITYNAME on the CONNECTION definition, or the USERID on the SESSIONS definition.

If a USERID is specified on the SESSIONS definition, and a link check is done, the userid used is the one on the SESSIONS definition.

If no userid is specified in SECURITYNAME, then the default userid of the AOR is used instead. However, if the SECURITYNAME userid is the same as the region userid for the AOR, then the link is deemed to have the same security as the AOR, and **link security is omitted altogether**. The effect of omitted link security depends on whether LOCAL or non-LOCAL attach security is specified for the link:

- For LOCAL attach security, the security specified in the USERID on the SESSIONS definition is used. If this too is omitted, then the default userid for the AOR is used.
- For non-LOCAL attach security, the security specified in the USERID on the sessions definition is **not** used. Only the userid received from the TOR is used to determine security.

Note: Neither the region userid for the TOR, nor the SECURITYNAME in the TOR's CONNECTION definition for the AOR, is relevant to security checking in the AOR.

Table 21 on page 113 shows how checking is done when ATTACHSEC(LOCAL) is specified.

Table 21. LU6.2 and ATTACHSEC(LOCAL)			
Region userid for AOR	**SECURITYNAME in connection definition**	**USERID in SESSION definition**	**Checking in AOR**
USERIDA	Not specified	Not specified	Check against AOR DFLTUSER
USERIDA	Not specified	USERIDA	Check against AOR DFLUTSER
USERIDA	Not specified	USERIDB	Check against USERIDB
USERIDA	USERIDA	Not specified	Check against AOR DFLTUSER
USERIDA	USERIDB	Not specified	Check against USERIDB
USERIDA	USERIDA	USERIDA	Check against AOR DFLTUSER
USERIDA	USERIDA	USERIDB	Check against USERIDB
USERIDA	USERIDB	USERIDA	Check against DFLTUSER
USERIDA	USERIDB	USERIDB	Check against USERIDB
USERIDA	USERIDB	USERIDC	Check against USERIDC

Table 22 on page 114 shows how checking is done when the ATTACHSEC parameter IDENTIFY (or PERSISTENT, or MIXIDPE) has been specified.

Table 22. LU6.2 and ATTACHSEC(IDENTIFY|PERSISTENT|MIXIDPE)			
Region userid for AOR	**SECURITYNAME in connection definition**	**USERID in SESSION definition**	**Checking in AOR**
USERIDA	Not specified	Not specified	Transmitted userid and AOR DFLTUSER
USERIDA	Not specified	USERIDA	Transmitted userid only
USERIDA	Not specified	USERIDB	Transmitted userid and USERIDB
USERIDA	USERIDA	Not specified	Transmitted userid only
USERIDA	USERIDA	USERIDA	Transmitted userid only
USERIDA	USERIDA	USERIDB	Transmitted userid and USERIDB
USERIDA	USERIDB	Not specified	Transmitted userid and USERIDB
USERIDA	USERIDB	USERIDC	Transmitted userid and USERIDC

Summary of resource definition options for LU6.2 security

The following is a summary of the resource definition options you need to define for LU6.2 security:

- On the CONNECTION definition:
  - ATTACHSEC, with any one of the following options:

    IDENTIFY
    LOCAL
    MIXIDPE
    PERSISTENT
    VERIFY

  - BINDPASSWORD
  - BINDSECURITY
  - SECURITYNAME
- On the SESSIONS definition:
  - USERID

For guidance on defining CONNECTION and SESSION resources, see the *CICS Resource Definition Guide*.

Chapter 13. APPC password expiration management

This chapter contains information on advanced program-to-program communications (APPC) password expiration management (PEM).

To use PEM you should understand APPC conversation-level security. To code the requester sign-on transaction, you also need to have basic APPC programming skills.

To find out what APPC PEM offers, read "Introduction to APPC password expiration management." System programmers responsible for coding the **PEM client** (requester) should also read "APPC PEM processing" on page 119, which explains the requirements of the PEM client and CICS PEM server.

Note: In this chapter the word 'sign-on' is used in the sense defined in the APPC architecture, which is different from the meaning used elsewhere in this book.

This chapter includes the following topics:

- "Introduction to APPC password expiration management"
- "What you require to use APPC PEM" on page 116
- "Roles of PEM client and CICS PEM server" on page 116
- "APPC PEM processing" on page 119
- "Overview of APPC PEM processing" on page 119
- "Setting up the PEM client" on page 124
- "PEM client input and output data" on page 127

Introduction to APPC password expiration management

This section introduces, and describes the benefits of, APPC password expiration management. For examples of PEM requester and CICS PEM server user data produced by a program, see:

- "Sign-on with correct userid and password" on page 132
- "Sign-on with new password" on page 133
- "Response to correct sign-on data" on page 134
- "Response to incorrect data format" on page 136

What APPC PEM does

APPC PEM with CICS provides receive support for an APPC architected sign-on transaction that verifies userid, password pairs, and processes requests for a password change by:

- Identifying a user and authenticating that user's identification
- Notifying specific users during the authentication process that their passwords have expired
- Letting users change their passwords when (or before) the passwords expire
- Telling users how long their current passwords will remain valid
- Providing information about unauthorized attempts to access the system using a particular user identifier

Benefits of APPC PEM

APPC PEM has the following benefits:

- It enables users to update passwords on APPC links.

  This can be particularly useful in the case of expired passwords. On APPC links that do **not** support APPC PEM, when users' passwords expire on remote systems, they are unable to update them from their own systems. The only alternative on a non-APPC PEM system is to log on directly to the remote system using a non-APPC link, such as an LU2 3270-emulation session, to update the password.
- It provides APPC users with additional information regarding their sign-on status; for example, the date and time at which they last signed on.
- It informs users whether their userid is revoked, or the password has expired, when they provide the correct password or PassTicket.

What you require to use APPC PEM

To use APPC PEM, you need a **PEM client** (requester) and a **PEM server** linked by an APPC session. Your external security manager must also be available to the PEM server. Figure 10 shows a sample configuration.

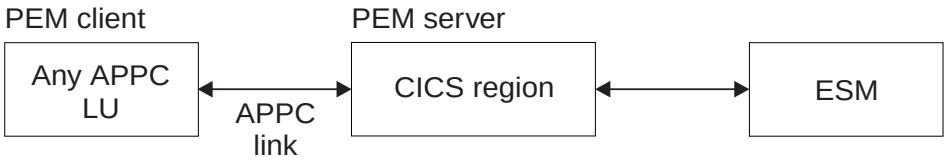


Figure 10. Sample APPC PEM configuration

The PEM client can be any APPC logical unit (LU) or node that is capable of initiating a conversation with the architected sign-on transaction. However, the benefits of using APPC PEM are increased when using an LU or node that does not have its own ESM; for example, a programmable workstation. APPC PEM enables users of such LUs or nodes to manage their password values within the ESM used by CICS.

The PEM server can be any APPC LU that supports APPC PEM. This chapter assumes that the PEM server is the one provided by CICS Transaction Server for VSE/ESA Release 1. It is referred to in the rest of this book as the CICS PEM server.

Roles of PEM client and CICS PEM server

CICS Transaction Server for VSE/ESA Release 1 does not send passwords on APPC conversations. This means that it can **attach**, but not **initiate** the sign-on transaction, and must always act as the PEM server. Therefore, in your configuration always include an LU that is capable of initiating the sign-on transaction to act as the PEM client.

The PEM client collects sign-on information and sends it to the CICS PEM server via a sign-on transaction program. The sign-on transaction program is a SNA service transaction program, as described in *SNA LU 6.2 Peer Protocols* manual.

Note that a PEM signon is not to be confused with a CICS signon. In CICS, PEM signon is a way for the APPC LU to verify and manage passwords. Following verification or updating, the userid or password is intended to be included as the ASIS part of the FMH5 attach header. When this FMH5 is sent into CICS through the APPC link (provided ATTACHSEC is non-local) the userid is signed on to CICS. Therefore, a PEM signon does not result in the ESM last-connect, last-access information being updated.

The CICS PEM server then processes the sign-on request, updates the user's password (if necessary), and returns a reply to the PEM client containing responses and other data, such as a password expiry and information about unauthorized attempts to sign on. The PEM client then processes the data, as appropriate.

An example of signing on with APPC PEM

Figure 11 on page 118 shows an example sign-on for APPC PEM.

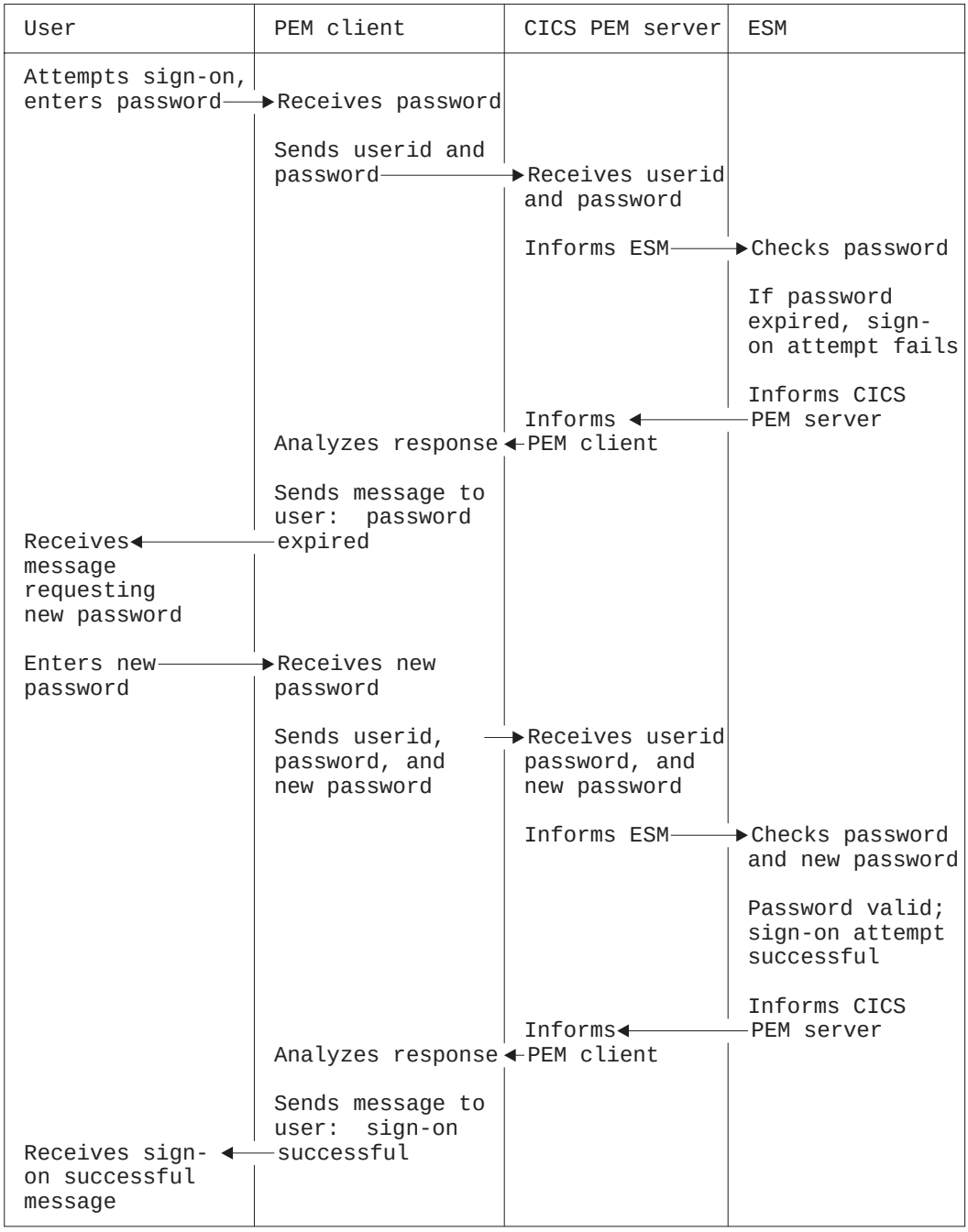


Figure 11. Example of signing on with APPC PEM

APPC PEM processing

In order to code the sign-on transaction program for the PEM client to send the sign-on details to the CICS PEM server, you need to know the following:

- What happens on each side of the link—see “Overview of APPC PEM processing.”
- How to code the PEM client—see “Setting up the PEM client” on page 124, “Format of user data” on page 125, and “Examples of PEM client and CICS PEM server user data” on page 132.
- The data the CICS PEM server requires from the PEM client—see “Sign-on input data sent by PEM client” on page 127
- The data the CICS PEM server sends in response to the PEM client—see “Sign-on output data returned by CICS PEM server” on page 129.

Overview of APPC PEM processing

CICS provides the PEM server, the receive side of APPC PEM as a CICS transaction (CLS4) that is started when an ATTACH for the sign-on transaction program is received from the PEM client.

CICS retrieves the sign-on data associated with the request, calls the ESM to perform a sign-on, and retrieves sign-on details for the userid. If the sign-on data includes a new password value, CICS includes this value when it calls the ESM to request a sign-on.

If PV is being used, and sign-on completes correctly, the user is added to the PV “signed-on-*from* list” in CICS, and the PV “signed-on-*to* list” in the PEM client. The “signed-on-” lists keep track of verified user IDs.

The CICS PEM server builds a reply and returns it to the PEM client, after which the CICS PEM server transaction terminates normally.

PEM client processing

The PEM client sign-on transaction program:

1. Obtains sign-on information, for example by:
   - Displaying a message to the user requesting sign-on information; that is, userid, password, and, if required, new password; or
   - Accessing sign on information from a user who has already been authenticated locally.
2. Sends the sign-on information to the CICS PEM server via an APPC conversation.
3. Receives replies from the CICS PEM server on the same APPC conversation.
4. If PV is being used (either ATTACHSEC=PERSISTENT or ATTACHSEC=MIXIDPE is specified on the CONNECTION definition), and sign-on is successful, adds the user’s name to the PV signed-on-to list.
5. Processes the reply information from the CICS PEM server; for example, by:
   - Displaying the information to the user

- Processing the data and saving it in a file to which only the user has access.

CICS PEM server processing

The CICS PEM server performs the following processing:

1. Accepts the userid and password, with optional new password, from the sign-on PEM client.
2. Tries to validate the user with its ESM.

   If the userid and password are valid and the password has not expired, the CICS PEM server extracts the following information from its ESM:

   - Date and time of the last successful sign-on
   - Data and time the password will expire (calculated by data extracted from the ESM by the CICS PEM server itself)
   - Number of unsuccessful sign-on attempts since the last successful sign-on.
3. Returns a response to the PEM client (described in Table 24 on page 129, and shown in "Examples of PEM client and CICS PEM server user data" on page 132 and Figure 19 on page 136), indicating whether the sign-on was succeeded or failed, and the reason for any failure:

   ```
   Status           = (X'00') OK
   Date-Time        = Current date and time
   Last-Date-Time   = Date and time of previous successful sign-on
   Expiry-Date-Time = Date and time password will expire
   Revoke-Count     = Number of unsuccessful sign-on attempts made with
                      this userid since the previous successful sign-on
   ```

   Note: The ESM increments the revoke count whenever it processes an invalid sign-on attempt.

   If sign-on is unsuccessful, CICS returns to the PEM client a sign-on completion status value (as described in Table 26 on page 131) and, if appropriate, a formatting error value (as described in Table 27 on page 131).
4. If PV is being used (either ATTACHSEC=PERSISTENT or ATTACHSEC=MIXIDPE is specified on the CONNECTION definition), and sign-on is successful, adds the user's name to the PV signed-on-from list.

Expected flows between PEM client and CICS PEM server

Figure 12 on page 121 through Figure 15 on page 124 show expected flows for successful and unsuccessful sign-on attempts with and without PV. These examples do not include information on setting up the connection. For more information on doing this, see the *CICS Intercommunication Guide*.

Note: CICS support for the PEM client sign-on transaction assumes that the request for sign-on (or sign-on and change password) is for a single user. Batching of sign-on requests for different userids within a single sign-on transaction is not supported. If multiple sign on requests are passed in the input data, the CICS PEM server processes only the first one.

Successful sign-on—non-PV connection: Figure 12 shows the expected flows between the PEM client and CICS PEM server during a successful sign-on when PV is not being used.

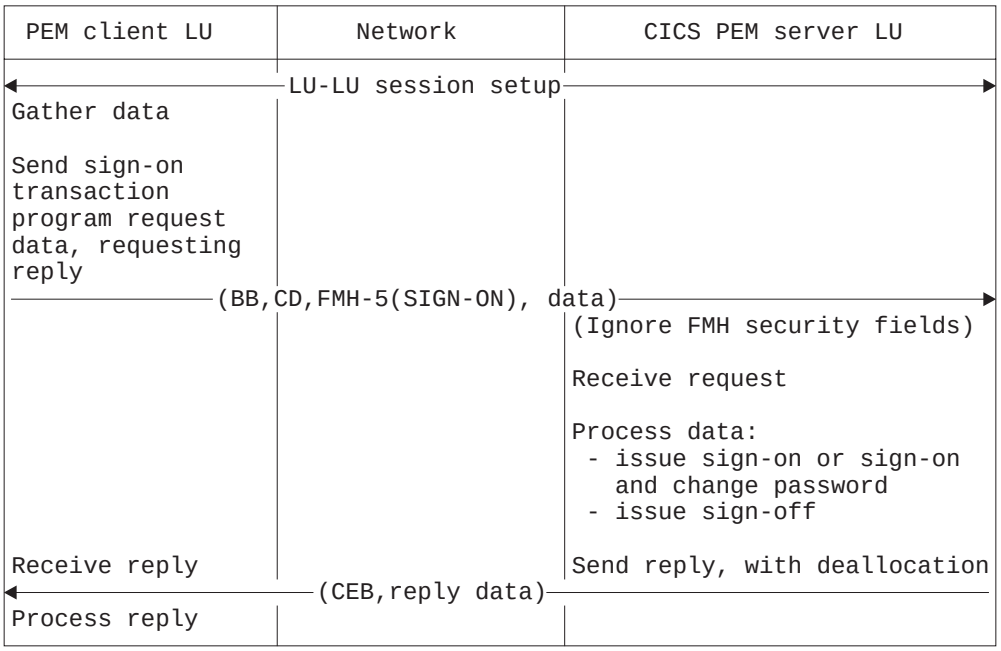


Figure 12. Successful sign-on—non-PV connection

Note: All security fields in the FMH-5 (userid, password and UP, AV, PV1 and PV2 bits) are ignored by the CICS PEM server when it attaches the sign-on transaction.

Unsuccessful sign-on—non-PV connection: Figure 13 shows the expected flows for an unsuccessful sign-on between a PEM client and CICS PEM server when PV is not being used.

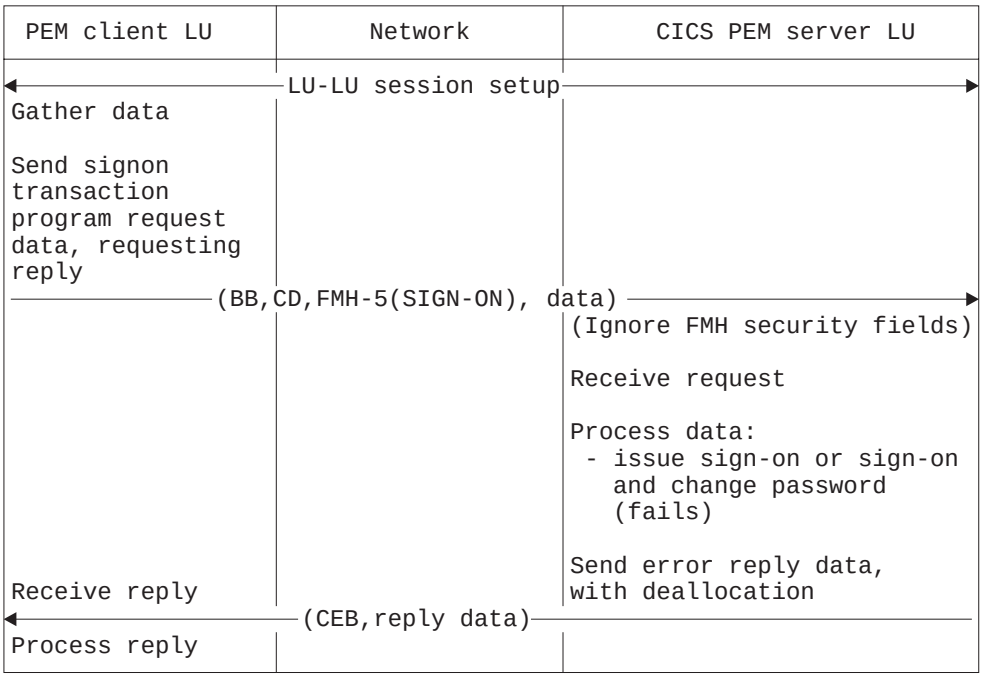


Figure 13. Unsuccessful sign-on—non-PV connection

Note: The CICS PEM server schedules sign-off against the PEM client if a sign-on request for a userid fails.

Successful sign-on—PV connection: Figure 14 shows the expected flows between the PEM client and CICS PEM server during a successful sign-on on a PV connection.

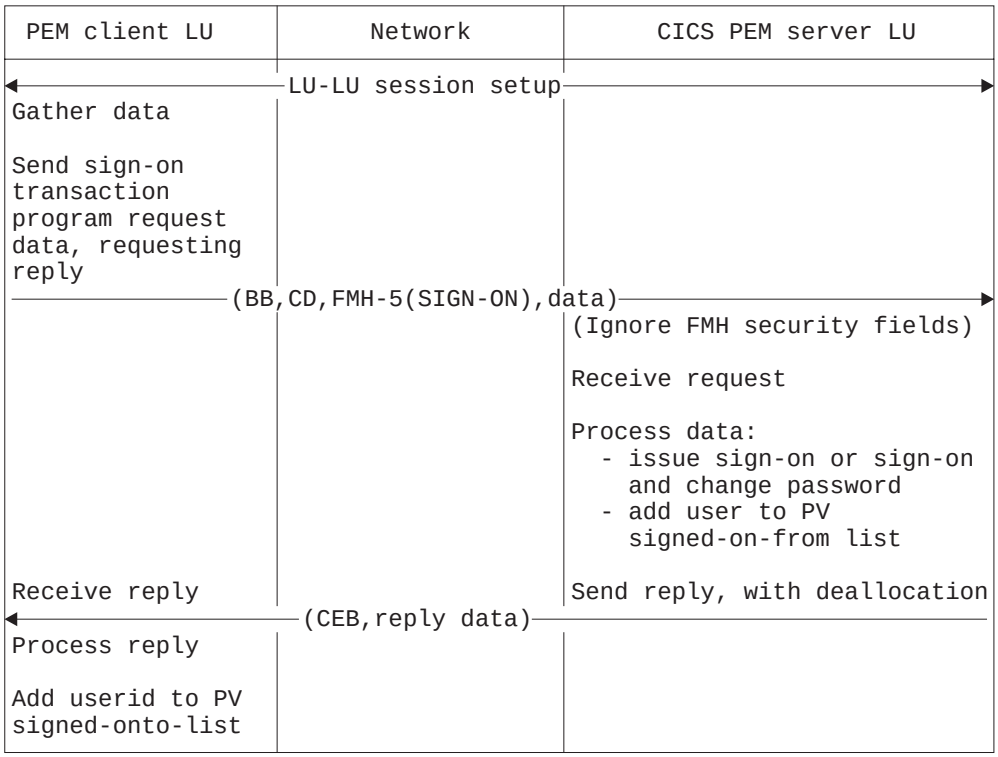


Figure 14. Successful sign-on—PV connection

Notes:

1. All security fields in the FMH-5 (userid, password and UP, AV, PV1 and PV2 bits) are ignored by the CICS PEM server when it attaches the sign-on transaction.
2. The CICS PEM server adds the userid to its PV signed-on-from list only if the sign-on and change password request is successful and either ATTACHSEC=MIXIDPE or ATTACHSEC=PERSISTENT is specified in the CONNECTION definition.
3. The PEM client must add the userid to its PV signed on-to list only if a successful sign-on reply is received from the CICS PEM server. The userid has been added to the PV signed on from list in the CICS PEM server, so all subsequent attach requests from this userid can flow as signed on.

Unsuccessful sign-on—PV connection: Figure 15 shows the expected flows between a PEM client and CICS PEM server during an unsuccessful sign-on on a PV connection.

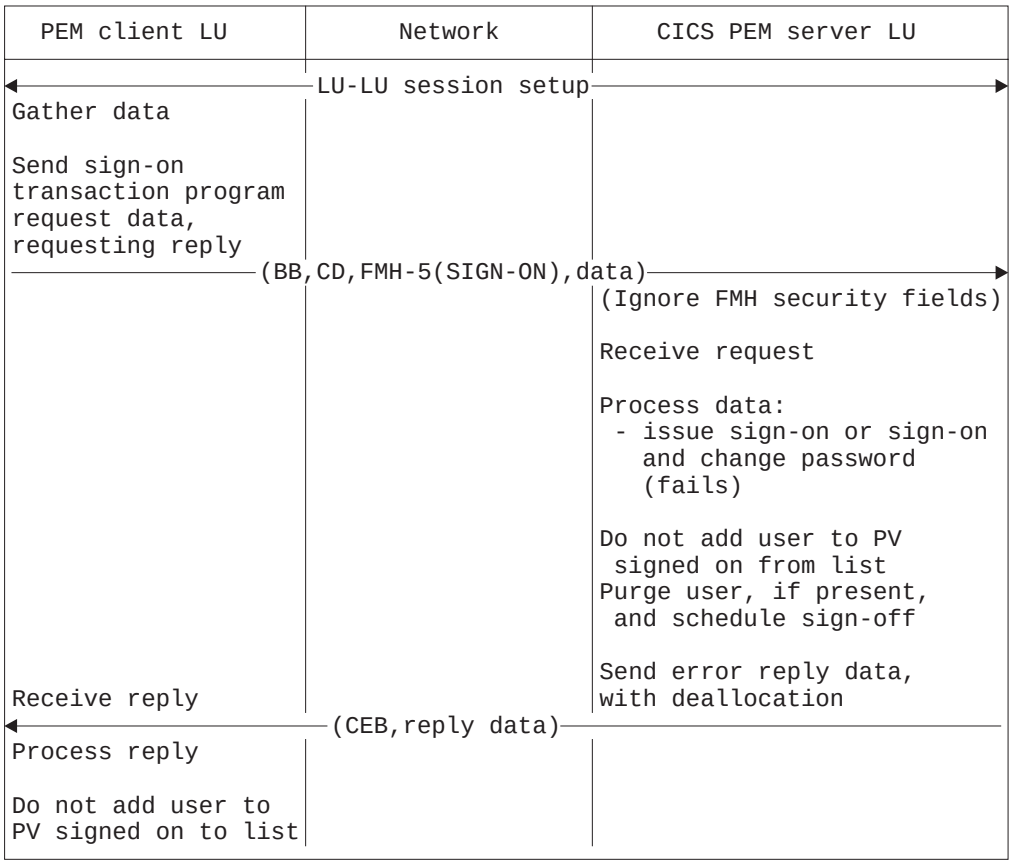


Figure 15. Unsuccessful sign-on—PV connection

Note: CICS schedules sign-off against the PEM client if a sign-on request for a userid fails, and that user is in the PV signed on from list. In this case, CICS sends the sign-off transaction program output data to the PEM client, where it is processed and the userid removed from the PV signed on to list.

Setting up the PEM client

When setting up your PEM client, note the following:

- Use the basic (also known as **unmapped**) conversation type. In addition to sending the data you want the partner to receive, you must add control bytes (in Assembler language or C) to convert the data into an SNA-defined format called a **generalized data stream** (GDS). Include the keyword GDS in any EXEC CICS commands used. See the *CICS Intercommunication Guide* for introductory information on basic conversations, and the *CICS Distributed Transaction Programming Guide* for information on using them.

- The SNA service transaction program name for the sign-on transaction program is **X'06F3F0F1'**, which is also the transaction id (XTRANID) that must be used for the CICS transaction CLS4. You specify XTRANID in the CICS TRANSACTION definition.
- Run the CICS PEM server sign-on transaction as a **sync level 0** transaction. If it is initiated with a sync level other than 0, it sends an ISSUE ABEND and frees the conversation.
- Translate the userid and password into EBCDIC; if they are not in this form, the ESM cannot recognize them and issues an error.

  Check whether the ESM requires the userid and password to be in uppercase characters.
- On the ATTACH request for the sign-on transaction program specify SECURITY(NONE). CICS ignores any ATTACH security fields passed in the ATTACH function management header, FMH-5, for this transaction.
- CICS does not support the receipt of the PROFILE option in the sign-on transaction program. If data identifier (ID) X'00' is provided, CICS returns status value X'06' (incorrect data format) with formatting error X'0002' (precluded structure present), as described in Table 27 on page 131.
- The new password ID, X'06', is permitted and required only with the X'FF01' request data ID. If the new password is provided with a data ID other than X'FF01', CICS returns status value X'06' (incorrect data format) with formatting error X'0002' (precluded structure present), as described in Table 27 on page 131.
- CICS only supports userids and passwords up to 8 characters long. If the userid or password length (after stripping blanks and nulls) exceeds 8, CICS returns status value X'06' (incorrect data format) with formatting error X'000F' (data value out of range), as described in Table 27 on page 131.
- Program initialization parameter (PIP) data is optional on the ALLOCATE for the sign-on transaction, and is ignored if sent.
- If the sign-on transaction receives a GDS ISSUE SIGNAL command, it is ignored.
- If the CICS PEM server receives a GDS ISSUE ERROR command, it replies with ERROR and frees the conversation.
- If the CICS PEM server receives a GDS FREE command, it frees the conversation. (It does not provide diagnostic information about the type of conversation error.)
- The CICS PEM server transaction does not support the receipt of data exceeding the maximum buffer size. If the concatenation bit in the initial LL is set, the command is ignored; concatenated data is also ignored.

Format of user data

As part of the general rules for APPC basic conversations, the user data must be in LL-ID-data format (where LL and ID are each two bytes long), and must follow the attach FMH-5 header. As described in Table 23 on page 128, the CICS DFHCLS4 program requires the user input data stream to fit into the format shown in Figure 16 on page 126; if it does not, CICS rejects the data.

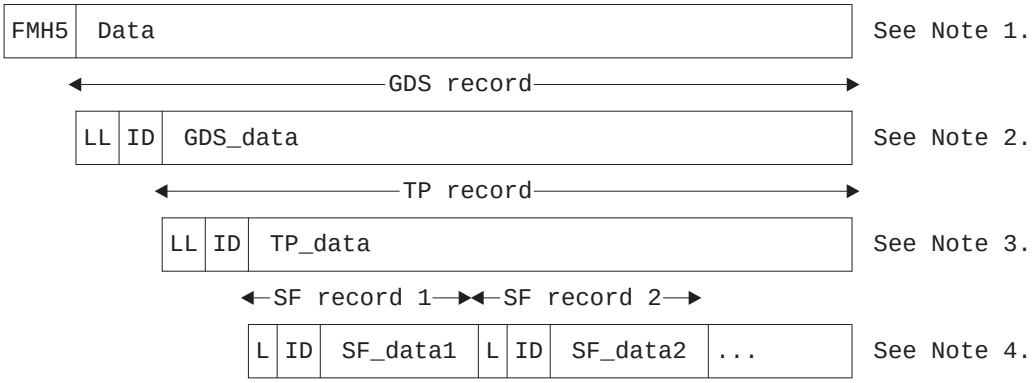


Figure 16. Format of user data

Notes:

1. This is an attach FMH-5 header with its data. Data is passed between the PEM client and the CICS PEM server via GDS variables. (For information on GDS, see the *SNA LU 6.2 Peer Protocols* manual.)
2. The **GDS record** contains GDS data in the format LL-ID-data where:
   - LL, which is two bytes long, is the length of the GDS record, including the LL and ID lengths.
   - ID, which is two bytes long, indicates what the data record represents; for example, X'1221' (sign-on data).
3. The GDS data record is itself an LL-ID-data record; in this example, a transaction program record (or **TP record**) where:
   - LL, which is two bytes long, is the length of the TP record including the LL and ID lengths.
   - ID which is two bytes long, indicates the function the TP is to perform; for example, X'FF00' (sign-on) or X'FF01' (signon and change password).
4. The TP data record is divided up into L-ID-data records (where L and ID are each **one** byte long). These are known as subfield (or **SF records**) where:
   - L is the length of the SF record, including the L and ID lengths.
   - ID indicates the subfield being passed; for example, X'01' (userid), X'02' (password), and X'06' (new password).

PEM client input and output data

To perform the functions described in “CICS PEM server processing” on page 120, the CICS PEM server takes input data from, and sends output data to, the PEM client sign-on transaction program:

- The PEM client sends data to the CICS PEM server, as described in Table 23 on page 128.
- The CICS PEM server sends data to the PEM client, as described in Table 24 on page 129 through Table 27 on page 131.

Ensure the data conforms to the standards described in “Setting up the PEM client” on page 124, and that its format is as described in “Format of user data” on page 125. See “Sign-on with correct userid and password” on page 132 and “Sign-on with new password” on page 133 for examples of sign-on output data in GDS flows.

Basic conversation information and data are contained in the attach FMH, as described in “Format of user data” on page 125. The sign on request attaches a transaction X'06F3F0F1', which is the SNA service transaction program name for the sign-on transaction program.

Sign-on input data sent by PEM client

Table 23 on page 128 shows the input data that the CICS PEM server needs from the PEM client sign-on transaction program. See “Sign-on with correct userid and password” on page 132 and “Sign-on with new password” on page 133 for examples of sign-on input data in GDS flows.

Table 23. Sign-on request and data sent to CICS PEM server

Length (bytes)	Value	Meaning
2	X'nnnn'	Length of entire GDS data, including this 2-byte length value.
2	X'1221'	Data ID for sign-on data.
2	X'nnnn'	Length of this second (nested) data structure (length, data ID, and data), including this 2-byte length value.
2	X'FF00' or X'FF01'	Data ID for sign-on or sign-on and change password request data, respectively. (New password subfield is not permitted for X'FF00'.)
1	X'nn'	Length of subfield for userid, including this 1-byte length value.
1	X'01'	Data ID of subfield for userid.
X'nn'–2	C'xxxxxxxx'	Userid.
1	X'mm'	Length of subfield for password, including this 1-byte length value.
1	X'02'	Data ID of subfield for password.
X'mm'–2	C'xxxxxxxx'	Password.
1	X'pp'	Length of subfield for new password, including this 1-byte length value.
1	X'06'	Data ID of subfield for new password.
X'pp'–2	C'xxxxxxxx'	New password.

Sign-on output data returned by CICS PEM server

Table 24 lists the sign-on output data that the CICS PEM server returns to the PEM client. See “Response to correct sign-on data” on page 134 and “Response to incorrect data format” on page 136 for examples of sign-on output data in GDS flows.

Table 24 (Page 1 of 2). Sign-on output data returned to PEM client

Length (bytes)	Value	Required or optional	Meaning
2	X'nnnn'	Required	Length of entire GDS data, including this 2-byte length value.
2	X'1221'	Required	Data ID of subfield for sign-on data.
2	X'nnnn'	Required	Length of this second (nested) data structure (length, data ID, and data), including this 2-byte length value.
2	X'FF02'	Required	Data ID for sign-on reply data.
1	X'03'	Required	Length of subfield for sign-on completion status, including this 1-byte length value.
1	X'00'	Required	Data ID of subfield for sign-on completion status.
1	X'00' through X'06'	Required	Sign-on completion status—see Table 26 on page 131.
1	X'04'	Optional	Length of subfield for sign-on request formatting error, including this 1-byte length value.
1	X'01'	Optional	Data ID of subfield for sign-on request formatting error.
2	X'0000' through X'0003', X'0005' through X'0007', X'000F'	Optional	Sign-on request formatting error—see Table 27 on page 131.
1	X'0A'	Optional	Length of subfield for date and time of current successful sign-on, including this 1-byte length value.
1	X'02'	Optional	Data ID of subfield for date and time of current successful sign-on.
8	See Table 25 on page 130 for format	Optional	Date and time of current successful sign-on. The date and time returned are extracted by the ESM from the user profile.
1	X'0A'	Optional	Length of subfield for date and time of last successful sign-on, including this 1-byte length value.
1	X'03'	Optional	Data ID of subfield for date and time of last successful sign-on.

Table 24 (Page 2 of 2). Sign-on output data returned to PEM client			
Length (bytes)	**Value**	**Required or optional**	**Meaning**
8	See Table 25 on page 130 for format	Optional	Date and time of last successful sign-on. The date and time returned are extracted by the ESM from the user profile.
1	X'0A'	Optional	Length of subfield for date and time password will expire, including this 1-byte length value.
1	X'04'	Optional	Data ID of subfield for date and time password will expire.
8	See Table 25 on page 130 for format.	Optional	Date and time password will expire. (The date and time returned are calculated from data obtained from the ESM.)
1	X'04'	Optional	Length of subfield for revoke count, including this 1-byte length value.
1	X'05'	Optional	Data ID of subfield for revoke count.
2	X'nnnn'	Optional	Revoke count.

Format of date and time subfields: Table 25 lists the format of the date and time subfields that the CICS PEM server can return to the PEM client, as referred to in Table 24 on page 129. See “Response to correct sign-on data” on page 134 for an example of date and time subfields in a GDS flow.

Table 25. Format of date and time subfields using 24-hour clock	
Position	**Meaning**
00	Two-byte year value; for example, 1994=X'07CB'.
02	One-byte month value; January=X'01', December=X'0C'.
03	One-byte day value; first day=X'01', thirty-first day=X'1F'.
04	One-byte hour value; midnight=X'00', 23rd hour=X'17'.
05	One-byte minute value; on the hour=X'00', 59th minute=X'3B'.
06	One-byte second value; on the minute=X'00', 59th second=X'3B'.
07	One-byte 100ths of a second value; on the second=X'00', maximum=X'63'.

Note: The maximum time value for a given day is 23 hours, 59 minutes, and 59.99 seconds (decimal). Midnight is 0 hours, 0 minutes, and 0 seconds on the following day.

Sign-on completion status values returned to PEM client: Table 26 on page 131 describes the sign-on completion status values (referred to in Table 24 on page 129) that the CICS PEM server can return to the PEM client in the status completion subfield in the sign-on reply data. See “Response to correct sign-on data” on page 134 for an example of sign-on completion status values in a GDS flow.

Table 26. Sign-on completion status values returned to PEM client	
Status value	**Meaning**
X'00'	All of the following conditions apply: • Userid valid • Password valid • Password not expired or new valid password specified When this status value is returned, the new password is set if specified, and PV processing (if used) is complete.
X'01'	Userid not known to the receiver.
X'02'	Userid valid, password incorrect.
X'03'	Userid valid, password correct but expired. New password must be set.
X'04'	Userid valid, password correct, new password not acceptable to receiving security system.
X'05'	Security function failure. Function not performed.
X'06'	Incorrect data format. Specific error is contained in the sign-on request formatting error subfield described in Table 27 on page 131.

Note: The CICS PEM server never returns either of the following sign-on status values to the PEM client:

- X'07'—general security error
- X'08'—password change completed, but sign-on failed.

Sign-on request formatting errors returned to PEM client: Table 27 lists the sign-on request formatting error values (referred to in Table 24 on page 129) that the CICS PEM server can return to the PEM client. Each is a 2-byte binary value. See “Response to incorrect data format” on page 136 for an example of sign-on request formatting errors in a GDS flow.

Table 27. Sign-on request formatting error values returned to PEM client	
Error value	**Description**
X'0000'	Undefined error not described below.
X'0001'	Required structure absent.
X'0002'	Precluded structure present.
X'0003'	Several occurrences of a nonrepeatable structure.
X'0005'	Unrecognized structure present where precluded.
X'0006'	Length outside specified range. This value assumes that the length arithmetic balances and that the sender intended to send the structure at that length.
X'0007'	Length exception. Length arithmetic is out of balance.
X'000F'	Data value out of range.

Application design

Design your applications to run the sign-on transaction before any other transaction. This keeps that any password check and any password changing separate from the application's own functions. In multitasking systems, it is possible for more than one sign-on transaction to start on parallel sessions. It is important that the code dealing with application-level ALLOCATE requests, serializes the sign-on process to completion, thus ensuring both flow as signed-on.

To record the date and time of a previous successful sign-on, the CICS PEM server sign-on program extracts password data from the ESM **before** it performs sign-on. If your system uses shared userids, and two users attempt to sign on at the same time, or if a user is multitasking, the time values returned to the PEM client for the current sign-on may not be the same as the timestamp recorded on the ESM database. Remember this if you are writing an application that is to run on multiple systems, and depends on the sign-on time returned to the PEM client. (This situation should not apply on a single system, provided the sign-on process is serialized as suggested.)

If PV is being used, and the interval specified in PVDELAY is exceeded, and the userid is removed from the PV sign on from list, applications must allow for the sign-on process to be serialized again.

General-use programming interface

Examples of PEM client and CICS PEM server user data

Data is passed between the PEM client and the CICS PEM server via GDS variables. To help you check the data being sent by your PEM client, the examples that follow show:

- “Sign-on with correct userid and password”
- “Sign-on with new password” on page 133
- “Response to correct sign-on data” on page 134
- “Response to incorrect data format” on page 136.

These examples are produced by the sample PEM client program shown in “Introduction to APPC password expiration management” on page 115. That program uses a **partner_LU_alias** of `hostcics`, an **LU_alias** of `ps2lua`, and a **mode_name** of `lu62ss`. When writing your own PEM client program, use the values defined in your communications manager configuration.

Sign-on with correct userid and password: The following shows a sample flow for a successful sign-on using the correct userid and password, with no new password.

```
PEM hostcics ps2lua lu62ss sec2r01 drtnnom
```

Figure 17. Sign-on with correct userid and password, no new password

A valid userid (sec2r01) and password (drtnnom) are correctly entered. No new password is entered.

The PEM client sends the following hexadecimal user data to the CICS PEM server:

```
001A12210016FF000901E2C5C3F2D9F0F10902C4D9E3D5D5D6D4
```

This contains the following values, as described in Table 23 on page 128:

001A Length of the entire GDS data, including this 2-byte length value

1221 Data ID for sign on data

0016 Length of this second (nested) data structure (length, data ID, and data), including this 2-byte length value

FF00 Data ID for sign-on request data

09 Length of subfield for userid, including this 1-byte length value

01 Data ID of subfield for userid

E2C5C3F2D9F0F1 Userid (SEC2R01) in EBCDIC

09 Length of subfield for password, including this 1-byte length value

02 Data ID of subfield for password

C4D9E3D5D5D6D4 Password (DRTNNOM) in EBCDIC

Sign-on with new password: The following is an example of a successful sign-on using a new password.

```
PEM hostcics ps2lua lu62ss sec2r01 drtnnom hursley
```

A userid, password, and new password are correctly entered.

The PEM client sends the following hexadecimal user data to the CICS PEM server:

```
0231221001FFF010901E2C5C3F2D9F0F10902C4D9E3D5D5D6D40906C8E4D9E2D3C5E8
```

This contains the following values, as described in Table 23 on page 128:

0023 Length of entire GDS variable, including this 2-byte length value

1221 Data ID for sign

001F Length of this second (nested) data structure (length, data ID, and data), including this 2-byte length value

FF01 Data ID for sign-on and change password request data

09 Length of subfield for userid, including this 1-byte length value

01 ID of subfield for userid

E2C5C3F2D9F0F1 Userid (SEC2R01) in EBCDIC

09 Length of subfield for password, including this 1-byte length value

02 ID of subfield for password

C4D9E3D5D5D6D4 Password (DRTNNOM) in EBCDIC

09 Length of subfield for new password, including this 1-byte length value

06 ID of subfield for new password

C8E4D9E2D3C5E8 New password (HURSLEY) in EBCDIC

Response to correct sign-on data: Figure 18 shows an example of the response to the correct sign-on data being entered.

```
PEM_OK
GDS LLID
00 2d 12 21
Sign-on Reply LLID
00 29 ff 02
Sign-on Completion Status Subfield
03 00 00
Date & Time of Current Successful Sign-on Subfield
0a 02 07 ca 01 14 0d 24 31 62
Date & Time of Last Successful Sign-on Subfield
0a 03 07 ca 01 11 16 1b 23 3e
Date & Time Password Will Expire Subfield
0a 04 07 ca 02 03 00 00 00 00
Revoke Count Subfield
04 05 00 00
```

Figure 18. Response to correct sign-on data

The first three lines of hexadecimal user data returned to the PEM client show the following *required* values, as described in Table 24 on page 129.

002d Total length of the GDS variable, including this 2-byte length value

1221 Data ID for sign-on data

0029 Length of this second (nested) data structure (length, data ID, and data), including this 2-byte length value

FF02 Data ID for sign-on reply data

03 Length of subfield for sign-on completion status, including this 1-byte length value

00 Data ID for sign-on completion status

00 Sign-on completion status. 00 indicates that the userid and password were valid, and the password had not expired. (See Table 26 on page 131 for a list of sign-on completion status values.)

In Figure 18, the last four lines of hexadecimal user data returned to the PEM client show the following optional values, as described in Table 24 on page 129. (Note that the formatting error subfields shown in Table 24 on page 129 are not included, indicating that there are no errors.)

0A Length of subfield for date and time of current successful sign-on including this 1-byte length value

02 Data ID for date and time of current successful sign-on

Date and time of current successful sign-on, as described in Table 25 on page 130:

07CA Year (1994)

01 Month (January)

14 Day (20)

0D Hour (13)

24 Minutes (36)

31 Seconds (49)

62 Hundredths of a second (98)

0A Length of subfield for date and time of previous successful sign-on,

03 Data ID for date and time of previous successful sign-on

Date and time of previous successful sign-on, as described in Table 25 on page 130:

07CA Year (1994)

01 Month (January)

11 Day (17)

16 Hour (22)

1B Minutes (27)

23 Seconds (35)

3E Hundredths of a second (62)

0a Length of subfield for date and time password will expire (including this 1-byte length value)

04 Length of subfield for data ID for date and time password will expire

Date and time password will expire, as described in Table 25 on page 130:

07ca Year (1994)

02 Month (February)

03 Day (14)

00 Hour (00)

00 Minutes (00)

00 Seconds 00)

00 Hundredths of a second (00)

04 Length of subfield for revoke count, including this 1-byte length value

05 Data ID of subfield for revoke count

0000 Revoke count. (0000 means that there have been no unsuccessful sign-on attempts since the last successful sign-on with this userid.)

Response to incorrect data format: Figure 19 shows an example response to incorrect data being entered.

```
PEM_OK
GDS LLID
00 0F 12 21
Sign-on Reply LLID
00 0B FF 02
Sign-on Completion Status Subfield
03 00 06
Sign-on Request Formatting Error Subfield
04 01 00 0F
```

Figure 19. Response to incorrect data format

The first three lines of hexadecimal user data returned to the PEM client show the following required values, as described in Table 24 on page 129:

`000F` Length of entire GDS data, including this 2-byte length value

`1221` Data ID for sign-on data

`000B` Length of this second (nested) data structure (length, data ID, and data), including this 2-byte length value

`FF02` Data ID for sign-on reply data

`03` Length of subfield for sign-on completion status, including this 1-byte length value

`00` Data ID of subfield for sign-on completion status

`06` Sign-on completion status 06 indicating incorrect data format (see Table 26 on page 131 for a list of signon completion status values.)

The last line of hexadecimal user data returned to the PEM client shows the following **optional** values, which are returned only if there is an error. (The optional values are described in Table 24 on page 129.)

`04` Length of subfield for sign-on request formatting error, including this 1-byte length value

`01` Data ID of subfield for sign-on request formatting error

`000F` Sign-on request formatting error, indicating “data value out of range” (see Table 27 on page 131 for a description of other possible formatting errors)

End of General-use programming interface

Chapter 14. Implementing LU6.1 security

This chapter tells you how to implement link security for LU6.1, and covers the following topics:

- "Link security with LU6.1"
- "Specifying ATTACHSEC with LU6.1" on page 138
- "Transaction, resource, and command security with LU6.1" on page 138
- "Function shipping security with LU6.1" on page 139
- "Security checking done in AOR with LU6.1" on page 140
- "Summary of resource definition options for LU6.1 security" on page 141

For LU6.1 links, CICS cannot check the identity of the requesting system, and the bind request is never rejected on security grounds. You are advised to use the intersystem security offered by LU6.2 links whenever possible. Note that no bind-time or user security can be applied to LU6.1 links.

Link security with LU6.1

Link security restricts the resources that a user can access, depending on the remote system from which they are accessed. The practical effect of link security is to prevent a remote user from attaching a transaction or accessing a resource for which the link userid has no authority.

Each link between systems is given an access authority defined by a link userid. A link userid for LU6.1 is a userid defined on your sessions definition for this connection. If not defined there, the link userid is taken to be the SECURITYNAME userid specified on the connection definition. If there is no SECURITYNAME, the link userid is the local region's default userid.

You cannot function ship to CICS without having a security check. However, the security check is minimized if the two regions involved are **equivalent systems**. This term means the same for LU6.1, LU6.2 and MRO: that the link userid matches the local region's userid.

If you have equivalent systems, the resource check is made against the local region's default user. If you do not have equivalent systems, the resource check is carried out against the link userid.

If a failure occurs in establishing link security, the link is given the security of the local region's default user. This would happen if, for example, the preset session userid had been revoked.

Specifying ATTACHSEC with LU6.1

With LU6.1 links, information about the remote user is not available for security purposes. In this case, the authority of the user is taken to be that of the link itself, and you must rely on link security alone to protect your resources.

With LU6.1, you can specify only ATTACHSEC(LOCAL) in the CONNECTION definition. Figure 20 shows an example of doing this using CEDA.

```
CEDA DEFINE CONNECTION(name)
  GROUP(groupname)
  .
  ATTACHSEC(LOCAL)
```

Figure 20. Defining sign-on level for user security with LU6.1

LOCAL is the default value. It specifies that a user identifier is not required from the remote system, and, if one is received, it is ignored. Here, CICS makes the user security profile equivalent to the link security profile. You do not need to specify ESM profiles for the remote users.

Transaction, resource, and command security with LU6.1

As in a single-system environment, links must be authorized to:

- Attach a transaction
- Access all the resources that the transaction is programmed to use.

This results in security levels called **transaction security**, **resource security**, and **command security**.

Transaction security

As in a single-system environment, the security requirements of a transaction are specified when the transaction is defined, as described in Chapter 5, “Transaction security” on page 41.

In an LU6.1 environment, a transaction can be initiated only if the link has sufficient authority.

Resource and command security

Resource and command security in an intercommunication environment are handled in much the same way as in a single-system environment.

CICS performs resource and command security checking only if the installed TRANSACTION definition specifies that they are required; for example, on the CEDA DEFINE TRANSACTION command, as shown in Figure 21 on page 139.

```
CEDA DEFINE TRANSACTION
  .
  RESSEC(YES)
  CMDSEC(YES)
  .
```

Figure 21. Specifying resource and command security for transactions

If a transaction definition specifies resource security checking, using RESSEC(YES), the link must have sufficient authority for the resources that the attached transaction accesses.

If a transaction definition specifies command security checking, using CMDSEC(YES), the link must have sufficient authority for the commands (COLLECT, CREATE, DISCARD, INQUIRE, PERFORM, and SET) that the attached transaction issues.

For further guidance on specifying resource and command security, see Chapter 6, “Resource security” on page 45 and Chapter 8, “CICS command security” on page 65.

NOTAUTH exceptional condition

If a transaction tries to access a resource, but fails the resource security checks, the NOTAUTH condition is raised.

When the transaction is the CICS mirror transaction, the NOTAUTH condition is returned to the requesting transaction, where it can be handled in the usual way.

Function shipping security with LU6.1

When CICS receives a function-shipped request, the transaction that is invoked is the **mirror transaction**. The CICS-supplied definitions of the mirror transactions all specify resource security checking, but not command security checking. This means that you are prevented from accessing the remote resources if the link does not have the necessary authority.

Note that **transaction routing** across LU6.1 links is not supported.

If the CICS-supplied definitions of the mirror transactions are not what your security strategy needs, you can change them by copying the definitions in group DFHISC into your own group, changing them, and then reinstalling them. For more information, see “Category 2 transactions” on page 83.

If you include a remote resource in your resource definitions, you can arrange for security checking to be done locally, just as if the resource were a local one. Also, the system that owns the resource can be made to apply an independent check, if it is able to receive the user identifier. You can therefore choose to apply security restrictions on both sides, on either side, or not at all.

Take care when you specify the SYSID option on a function-shipped request. Security checking is done in the remote system but is **bypassed in the local system**. Figure 22 on page 140 summarizes what happens.

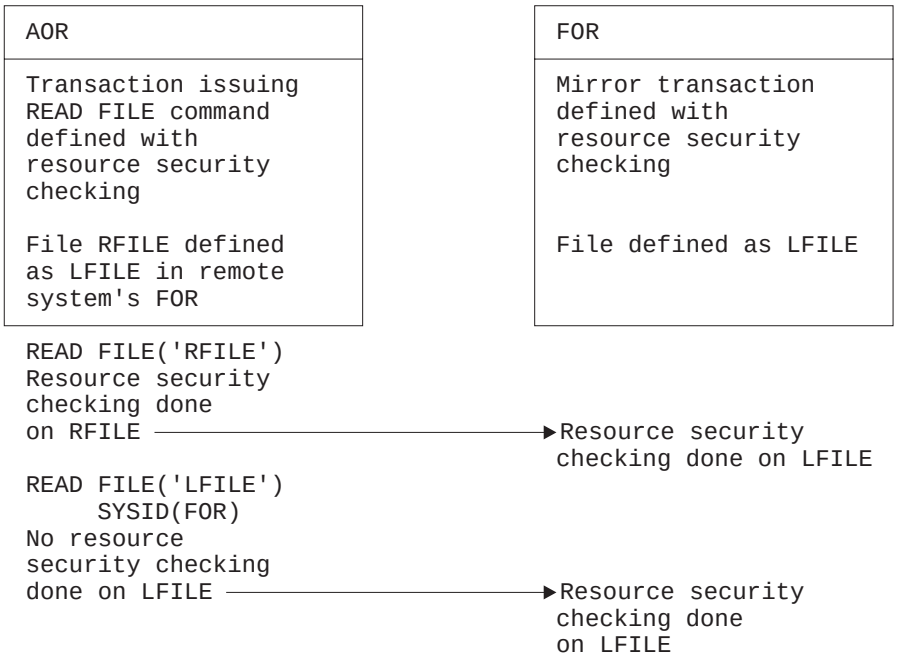


Figure 22. Security checking done with and without SYSID

For programming information on specifying the SYSID option, see the *CICS Application Programming Reference* manual.

Security checking done in AOR with LU6.1

This section summarizes how security checking is done in the AOR according to how SECURITYNAME is specified in the AOR and TOR, in an LU6.1 environment.

The link userid referred to in Table 28 on page 141 is the one specified in the SECURITYNAME on the CONNECTION definition, or the USERID on the SESSIONS definition.

If a USERID is specified on the SESSIONS definition, and a link check is done, the userid used is the one on the SESSIONS definition.

Table 28 on page 141 shows how checking is done when ATTACHSEC(LOCAL) is specified.

Neither the region userid for the TOR, nor the SECURITYNAME in the TOR's CONNECTION definition for the AOR, is relevant to security checking in the AOR.

Table 28. Security checking done in AOR

Region userid for AOR	SECURITYNAME in CONNECTION definition	USERID in SESSION definition	Checking in AOR
USERIDA	Not specified	Not specified	Check against AOR DFLTUSER
USERIDA	Not specified	USERIDA	Check against AOR DFLTUSER
USERIDA	Not specified	USERIDB	Check against USERIDB
USERIDA	USERIDA	Not specified	Check against AOR DFLTUSER
USERIDA	USERIDB	Not specified	Check against USERIDB
USERIDA	USERIDA	USERIDA	Check against AOR DFLTUSER
USERIDA	USERIDA	USERIDB	Check against USERIDB
USERIDA	USERIDB	USERIDA	Check against AOR DFLTUSER
USERIDA	USERIDB	USERIDB	Check against USERIDB
USERIDA	USERIDB	USERIDC	Check against USERIDC

Summary of resource definition options for LU6.1 security

The following is a summary of the resource definition options you need to define for LU6.1 security:

- On the CONNECTION definition:
  - ATTACHSEC, with the LOCAL option specified or allowed to default
  - SECURITYNAME
- On the SESSIONS definition:
  - USERID

For guidance on defining CONNECTION and SESSION resources, see the *CICS Resource Definition Guide*.

Chapter 15. Implementing MRO security

This chapter tells you how to implement CICS multiregion operation (MRO) security, and is organized as follows:

- "Bind-time security with MRO"
- "Logon security checking with MRO"
- "Link security with MRO" on page 145
- "User security with MRO" on page 146
- "Transaction, resource, and command security with MRO" on page 149
- "Transaction routing security with MRO" on page 150
- "Function shipping security with MRO" on page 152
- "Distributed program link security with MRO" on page 153
- "Security checking done in AOR with MRO" on page 154
- "Summary of resource definition options for MRO security" on page 155.

Bind-time security with MRO

The CICS interregion communication (IRC) facility supports MRO through the use of DFHAPPL.*applid* profiles in the FACILITY class.

There are two phases to bind security checking in DFHIRP, and these occur at:

- Logon time
- Connect time

These security checks, via RACROUTE calls to the SAF interface, are always performed, regardless of whether the or not MRO partner regions are running with external security active for CICS resource security checking (that is, for both SEC=YES and SEC=NO). In order for an MRO connection to be established between two regions, both the logon and connect security checks in both systems must be completed successfully. This security is applied to earlier releases of CICS using the CICS Transaction Server for VSE/ESA Release 1 of DFHIRP, the CICS interregion communication program.

Logon security checking with MRO

Logon security checking is performed whenever a CICS region logs on to the CICS-supplied interregion communication (IRC) program, DFHIRP.

CICS interregion communication uses the external security manager to check that CICS regions logging on to IRC are the regions they claim to be.

Each region that uses the IRC access method must be authorized to the ESM in a DFHAPPL.*applid* profile in the FACILITY class, (or equivalent). This requires the definition of a DFHAPPL.*applid* profile for each region that logs on to DFHIRP, and that each CICS region userid has UPDATE access to its own DFHAPPL.*applid* profile.

See Figure 23 for an illustration of logon checking.

Connect security

To perform MRO connect security checking, DFHIRP checks that each CICS region in the connection has read access to its partner's DFHAPPL.*applid* profile.

When CICS Transaction Server for VSE/ESA Release 1 DFHIRP is installed, all regions using earlier CICS releases in the VSE image use the DFHAPPL.*applid* form of MRO connect security. In addition, the SECURITYNAME parameter on the CONNECTION definition is not used for MRO and is ignored.

To authorize the MRO partner regions for bind security purposes, you must define the appropriate DFHAPPL profiles in the ESM FACILITY class. This means that each CICS region in an MRO interregion communication link must be given access to its partner's DFHAPPL.*applid* profile with READ access authority.

You cannot specify to CICS whether or not you want connect security checking for MRO connections—CICS always issues the RACROUTE calls. For example, the CICS TOR running under userid CICSRTOR (with APPLID CICSATOR), that connects to the AOR running under userid CICSRAOR (with APPLID CICSAAOR), will undergo the logon and connect security checks shown in Figure 23.

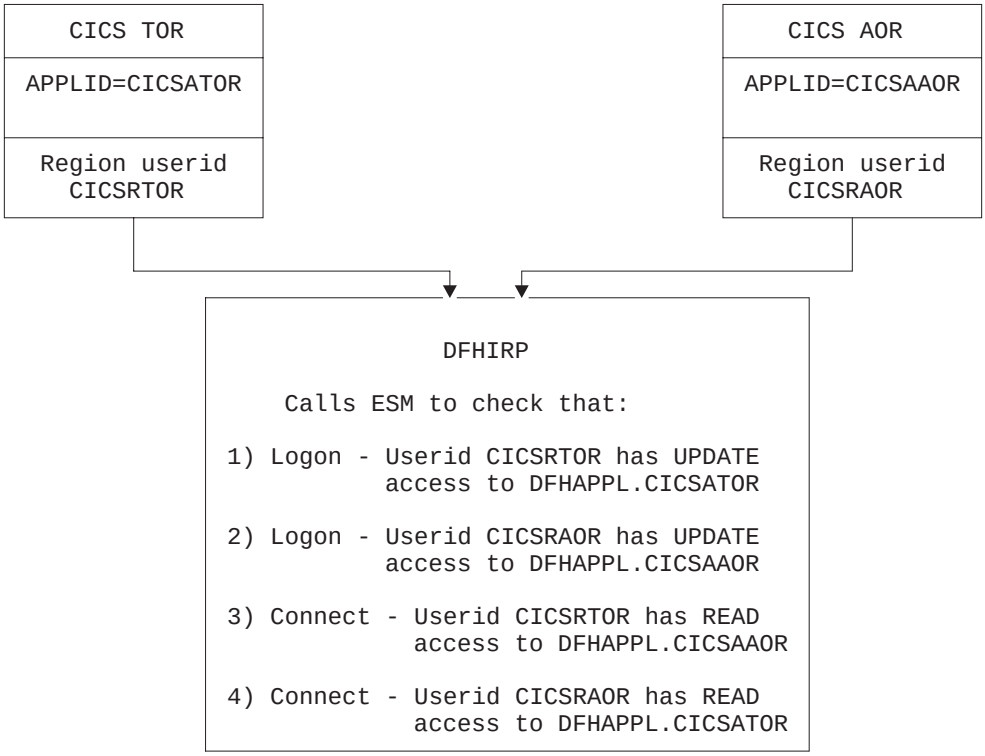


Figure 23. Illustration of the DFHIRP logon and connect security checks

Responses from the system authorization facility (SAF)

If the security profile for a specified resource is not retrieved, SAF neither grants nor refuses the access request. In this situation:

IRC rejects the logon or connect request if:

- A security manager was installed, but is either temporarily inactive or inoperative for the duration of the VSE image. This is a fail-safe action, on the grounds that, if the security manager was active, it might retrieve a profile that does not permit access.

IRC allows the logon or connect request if:

- There is no security manager installed, or
- There is an active security manager, but the FACILITY class is inactive, or there is no profile in the FACILITY class. The logon is allowed in this case because there is no evidence that you want to control access to the CICS APPLID.

Any CICS region without a specific DFHAPPL.*applid* profile, or applicable generic profile, permits all logon and connect requests. No messages are issued to indicate this. To avoid any potential security exposures, you may consider using generic profiles to protect all, or specific groups of, regions before, or in parallel with, security measures for specific regions.

Link security with MRO

Link security restricts the resources that a user can access, depending on the remote system from which they are accessed. The practical effect of link security is to prevent a remote user from attaching a transaction or accessing a resource for which the link userid has no authority.

Each link between systems is given an access authority defined by a link userid. A link userid for MRO is a userid defined on your sessions definition for this connection. Note that for MRO, unlike LU6.2, you can have only one sessions definition per connection, and there can be only one link userid per connection. If there is no preset session userid, the link userid is taken to be the region userid of the TOR region. The SECURITYNAME field on the connection definition is ignored for MRO.

You can never transaction route or function ship to CICS without having at least one security check, but the security checks done are minimized if the two regions involved are **equivalent systems**. This term means that the link userid matches the local region's userid and applies to LU6.1, LU6.2, and MRO.

If you have equivalent systems, you will always only have one security check. This will be made either against the local region's default user (for ATTACHSEC=LOCAL) or against the userid in the received FMH-5 attach request (ATTACHSEC=IDENTIFY).

If you do not have equivalent systems for ATTACHSEC=LOCAL, resource checks are done only against the link userid. For ATTACHSEC=IDENTIFY you will always have two resource checks. One check is against the link userid, and the other is against the userid received from the remote user in the attach request.

If a failure occurs in establishing link security, the link is given the same security authorization as defined for the local region's default user. This would happen, for example, if the preset session userid had been revoked.

Associate the SESSIONS definition with an ESM user profile that has access to any protected resource to which the inbound transaction needs access. See Chapter 2, "Facilities provided by an external security manager" on page 9 for guidance on defining profiles.

If the sign-on fails, a sign-on failure message is sent to the CSCS security destination, and the link is given the security of the DFLTUSER in the receiving system; that is, it is able to access only those resources to which the default user has access.

Obtaining the CICS region userid

For the purposes of MRO logon and connect security checks, DFHIRP needs to know the CICS region userid under which the CICS job or task is running. DFHIRP obtains the CICS region's userid by issuing a RACROUTE REQUEST=EXTRACT macro.

CICS determines whether a security manager is present or not by examining the SAF response codes.

User security with MRO

User security causes CICS to make a second check against a user signed on to a terminal, in addition to the link security check described in "Link security with MRO" on page 145. You should consider whether you want the extra level of security checking that user security provides.

You can specify either LOCAL, in which case the user is not checked, or IDENTIFY, in which case a userid is required, but no password is sent.

You specify the sign-on support for each connection using the ATTACHSEC operand of CONNECTION definition, as described in "User security in link definitions."

User security in link definitions

The level of user security you require for a remote system is specified in the ATTACHSEC operand of the CONNECTION definition. Figure 24 shows an example of defining ATTACHSEC using CEDA.

CICS interprets the parameters of the ATTACHSEC operand as described here. However, special rules apply for CICS transaction routing using CRTE, as described in "CICS routing transaction, CRTE" on page 151.

```
CEDA DEFINE CONNECTION(name)
  GROUP(groupname)
  .
  ATTACHSEC(LOCAL|IDENTIFY)
```

Figure 24. Defining sign-on level for user security

The ATTACHSEC operand specifies the sign-on requirements for incoming requests. It has no effect on requests that are issued by your system to a remote system; these are dealt with by the remote system.

The following ATTACHSEC operands are valid with MRO:

LOCAL
specifies that a user identifier is not required from the remote system, and if one is received, it is ignored. Here, CICS makes the user security profile equivalent to the link security profile. You do not need to specify ESM profiles for the remote users. (LOCAL is the default value.)

Specify ATTACHSEC(LOCAL) if you think that the link security profile alone provides sufficient security for your system.

IDENTIFY
specifies that a user identifier is expected on every attach request. All remote users of a system must be identified to the ESM.

Specify ATTACHSEC(IDENTIFY) when you know that CICS can trust the remote system to verify its users, when, for example, the remote system is another CICS.

The following rules apply to IDENTIFY:

- If a password is included in an attach request with a user identifier on a link with ATTACHSEC(IDENTIFY), CICS rejects the attach request and unbinds the session.
- If a null user identifier or an unknown user identifier is received, CICS rejects the attach request.
- If no user identifier is received, the attach is rejected unless USEDFLTUSER(YES) is specified on the connection. In this case CICS applies the security capabilities of the default user, as specified in the DFLTUSER system initialization parameter. For more information, see “CICS default user” on page 12.

Note: In the case of distributed transaction processing (DTP) transactions, you must issue a BUILD ATTACH request before the MRO SEND or CONVERSE command to include the userid of the terminal user in an attach request.

Sign-on status

With ATTACHSEC(IDENTIFY), the remote user remains signed-on after the conversation associated with the first attach request is complete. CICS then accepts attach requests from the same user without a new sign-on until either of the following occurs:

- The period specified in the system initialization parameter USRDELAY elapses after completion of the last transaction associated with the attach request for this user.

  When you are running remote transactions, over ISC and IRC links, USRDELAY defines the length of time for which entries can remain signed onto the remote CICS region. For information on specifying USRDELAY, see the *CICS System Definition Guide*. For information on tuning, see the *CICS Performance Guide*
- The CICS system is terminated.

If you alter the ESM profile of a signed-on remote user (for example, by revoking the user), CICS continues to use the authorization established at the first attach request until the user is signed off by one of the events just described.

Information about remote users

With MRO links, information about the user can be transmitted with the attach request from the remote system. This means that you can protect your resources not only on the basis of which remote system is making the request, but also on the basis of which actual user at the remote system is making the request.

This section describes some of the concepts associated with remote-user security, and how CICS sends and receives user information.

You will have to define your users to the ESM. If a remote user is not defined to the ESM, any attach requests from that remote user are rejected.

CICS sends userids on ATTACHSEC(IDENTIFY) conversations. Table 29 shows how CICS decides which userid to send.

Table 29. MRO attach-time user identifiers

Characteristics of the local task	**User identifier sent by the TOR to the AOR**
Task with associated terminal—user identifier	Terminal user identifier
Task with associated terminal—no user signed on and no USERID specified in the terminal definition	Default user identifier from the TOR
Task with no associated terminal or USERID, started by interval control START command (if using function shipping or DTP)	User identifier for the task that issued the START command
Task started with USERID option	User identifier specified on the START command
CICS internal system task	CICS region userid
Task with no associated terminal, started by transient data trigger	User identifier specified on the DCT that defines the queue
Task with associated terminal, started by transient data trigger	Terminal user identifier
Task started from PLTPI	User identifier specified by the PLTPIUSR system initialization parameter

New sign-on authorization processes

The sign-on authorization process is affected by the following:

1. When signing on users in the terminal-owning region, CICS passes to the ESM one of the following names as the CICS APPL name:
   - The generic APPLID if one is specified on the APPLID system initialization parameter
   - The specific APPLID if only one is specified on the system initialization parameter

The effect of this change is that you need define only one APPL profile name in the ESM database for all the CICS regions that are members of the same VTAM generic resources name, (for example, in XRF systems).

2. CICS passes the APPL name used in the sign-on process, and the NETNAME, across all MRO links (for example, from TOR to AOR, and from AOR to FOR). When signing-on the user in application-owning region and file-owning regions, where the connection definition specifies ATTACHSEC=IDENTIFY, CICS passes the terminal-owning region's APPLID and NETNAME to ESM.

   There are several benefits from this. It enables the ESM to reuse original terminal-owning region sign-on information, when CICS is signing on the user in the application-owning region. This gives a significant improvement in performance. It also reduces the number of APPL profiles you need to maintain in the ESM database, saving on security administration. Finally, it prevents users signing on directly to an application-owning region, because the terminal-owning region APPLs are the only ones to which they are authorized.

Transaction, resource, and command security with MRO

As in a single-system environment, users must be authorized to:

- Attach a transaction.
- Access all the resources that the transaction is programmed to use. This results in security levels called transaction security, resource security, and command security.

Transaction security

As in a single-system environment, the security requirements of a transaction are specified when the transaction is defined, as described in Chapter 5, “Transaction security” on page 41.

In an MRO environment, two basic security requirements must be met before a transaction can be initiated:

- The link must have sufficient authority to initiate the transaction.
- The “user” who is making the request must have sufficient authority to access the system and to initiate the transaction.

Resource and command security

Resource and command security in an intercommunication environment are handled in much the same way as in a single-system environment.

When resource and command security checking are performed

Resource and command security checking are performed only if the installed transaction definition specifies that they are required; for example, on the CEDA DEFINE TRANSACTION command, as shown in Figure 25 on page 150.

```
CEDA DEFINE TRANSACTION
  .
  RESSEC(YES)
  CMDSEC(YES)
  .
```

Figure 25. Specifying resource and command security for transactions

If a transaction specifies resource security checking, using RESSEC(YES), both the link and the user must also have sufficient authority for the resources that the attached transaction accesses.

If a transaction specifies command security checking, using CMDSEC(YES), both the link and the user must also have sufficient authority for the commands (shown in Table 12 on page 66) that the attached transaction issues.

For further guidance on specifying resource and command security, see Chapter 6, “Resource security” on page 45 and Chapter 8, “CICS command security” on page 65.

NOTAUTH exceptional condition

If a transaction tries to access a resource, but fails the resource security checks, the NOTAUTH condition is raised.

When the transaction is the CICS mirror transaction, the NOTAUTH condition is returned to the requesting transaction, where it can be handled in the usual way.

Transaction routing security with MRO

In transaction routing, the authority of a user to access a transaction can be tested in both the TOR and the AOR.

In the TOR, a normal test is made to ensure that the user has authority to access the transaction defined as remote, just as if it were a local transaction. This test determines whether the user is allowed to run the relay program.

In the AOR, the transaction has as its principal facility a remote terminal (the “surrogate” terminal) that represents the “real” terminal in the TOR. The way in which the remote terminal is defined (see the *CICS Intercommunication Guide*) affects the way in which user security is applied.

- If the definition of the remote terminal does not specify the USERID parameter:
  - For links with ATTACHSEC(IDENTIFY), the transaction security and resource security of the user are established when the remote user is signed on. The userid under which the user is signed on, whether explicitly or implicitly (in the DFLTUSER system initialization parameter), has this security capability assigned in the remote system.
  - For links with ATTACHSEC(LOCAL), transaction security, command security, and resource security are limited by the authority of the link.

In both cases, tests against the link security are made as described in “Link security with MRO” on page 145.

During transaction routing, the 3-character operator identifier from the TOR is transferred to the surrogate terminal entry in the AOR. This identifier is not used for security purposes, but it may be referred to in messages and audit trails.

Preset-security terminals and transaction routing

Preset-security for a terminal is determined by the specification of the USERID parameter.

When considering the security aspects of transaction routing from a preset-security terminal, remember that preset-security is an attribute of the terminal rather than of the user who is performing the transaction routing request.

During transaction routing, CICS creates a surrogate terminal in the AOR to represent the terminal at which the transaction routing request was issued. Whether the surrogate terminal has preset- security or not depends upon a number of factors:

- If a remote terminal definition exists in the AOR for the terminal at the TOR, and specifies the USERID parameter, the surrogate terminal is preset with this userid. If the USERID parameter is not coded, the surrogate terminal does not have preset-security.
- If a remote terminal definition does not exist in the AOR, the preset-security characteristics of the surrogate terminal are determined from the terminal definition shipped from the TOR. If the shipped terminal definition has preset security, the surrogate also has preset security, unless the connection to the AOR is defined with ATTACHSEC=LOCAL, in which case any preset security information shipped to the AOR is ignored.

CICS routing transaction, CRTE

You can use the CICS routing transaction, CRTE, with MRO to run transactions that reside on a connected remote system, instead of defining these transactions as remote in the local system. CRTE is particularly useful for infrequently used transactions, or for transactions such as CEMT that reside on all systems.

Ensure that the terminal through which CRTE is invoked is defined on the remote system (or defined as “shippable” in the local system). The terminal operator needs security authority if the remote system is protected.

Security checking done in the AOR for transactions executed under CRTE does not depend on what is specified on ATTACHSEC, nor on the userid signed on in the TOR. Instead, security checking depends on whether the user signs on while using CRTE:

- If the user does **not** sign on, the surrogate terminal created is associated with the AOR default user. When a transaction is run, the security checks are carried out against this default user. A check is also done against the link userid to see whether the routing application itself has authority to access the resource.
- If the user does **not** sign on, using the CESN transaction while running CRTE, the surrogate points to the userid of the signed-on user. For transactions

attempting to access resources, security checking is done against the signed-on user's userid in the surrogate and the link userid.

For more information on CRTE, see the *CICS-Supplied Transactions* manual and the *CICS Intercommunication Guide*.

Function shipping security with MRO

When CICS receives a function-shipped request, the transaction that is invoked is the **mirror transaction**. The CICS-supplied definitions of the mirror transactions all specify resource security checking, but not command security checking. This means that you are prevented from accessing the remote resources if either the link or your user profile on the other system does not have the necessary authority.

If the CICS-supplied definitions of the mirror transactions are not what your security strategy needs, you can change them by copying the definitions in group DFHISC into your own group, changing them, and then reinstalling them. For more information, see "Category 2 transactions" on page 83.

If you include a remote resource in your resource definitions, you can arrange for security checking to be done locally, just as if the resource were a local one. Also, the system that owns the resource can be made to apply an independent check, if it is able to receive the user identifier. You can therefore choose to apply security restrictions on both sides, on either side, or not at all.

Note: If you specify the SYSID option on a function-shipped request, security checking is done in the remote system but is **bypassed in the local system**. Figure 26 summarizes what happens.

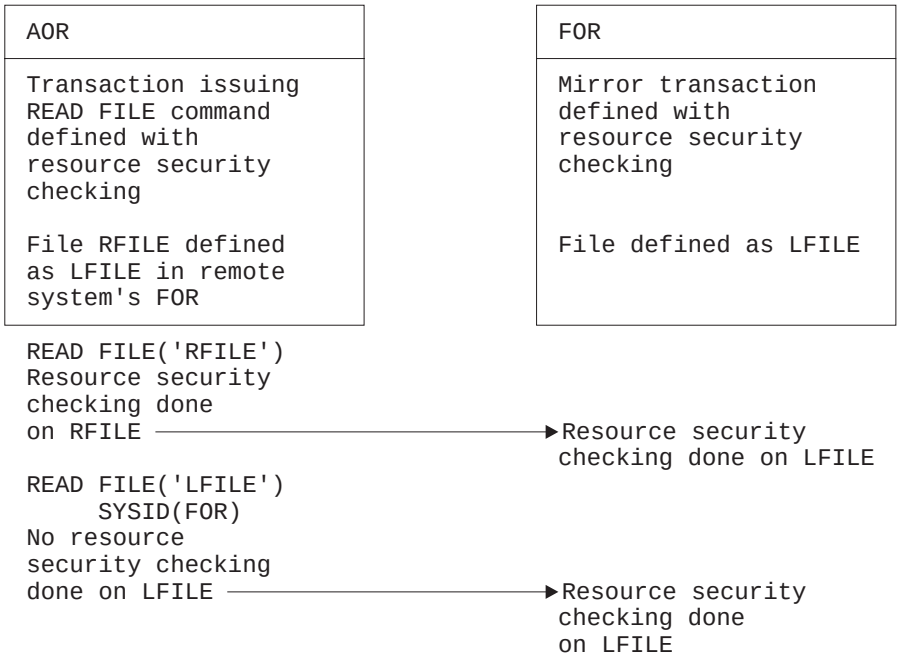


Figure 26. Security checking done with and without SYSID

For programming information on specifying the SYSID option, see the *CICS Application Programming Reference* manual.

Distributed program link security with MRO

The CICS distributed program link (DPL) facility enables a program (the client program) to call a CICS program (the server program) in a remote CICS region. The client program may be a CICS program or a non-CICS program.

A CICS client program uses DPL by specifying the SYSID option on the EXEC CICS LINK PROGRAM command, or omitting the SYSID option if the REMOTESYSTEM option of the program resource definition already specifies a remote CICS region. When the SYSID option on the EXEC CICS LINK command specifies a remote CICS system, the client region does not perform any resource security checking, but leaves the resource check to be performed in the server region.

A non-CICS client program uses calls to DFHXCIS to open a line to the CICS system, and then to link to a CICS program. This is called the external CICS interface (EXCI). One of the parameters of the link call is the transaction identifier under which the server program is to run. Define this transaction to CICS as running program DFHMIRS and as using profile DFHCICSA. Another parameter of the link call is the client's userid, which is validated if the MRO connection has been defined with ATTACHSEC(IDENTIFY).

To use the userid parameter in the DFHXCIS call, the client program must have surrogate-user authority to the specified userid. This is described in more detail in the *CICS External Interfaces Guide*. For information about using the SURROGCHK parameter to specify surrogate user checking on DPL calls, see "Userid passed as parameter on EXCI calls" on page 61.

The client program receives a USER_ERROR error if the external CICS interface command fails the security check. However, this error can have other causes; each reason code value for a USER_ERROR response indicates whether the command can be reissued directly, or whether the pipe being used has to be closed and reopened first.

The server program is executed by a mirror transaction, in a similar way to other function-shipped CICS requests. However, the transaction name associated with the mirror depends on how the program link is invoked in the client region. You must be aware of the transaction name because normal attach security applies to the mirror transaction:

- If a transaction identifier is specified on the link request, the specified transaction name is used for the mirror.
- If the transaction is omitted from the link request, but the TRANSID option is used in the program resource definition in the client region, the name for the mirror is taken from the program's TRANSID specification.
- Otherwise, the default name of CSMI is used for the mirror transaction.

Authorize users to access the transaction name that the mirror runs under. The userids to be authorized depend on whether LOCAL or IDENTIFY attach security is being used, and are described in "Security checking done in AOR with MRO" on page 154. If you define the mirror transaction with RESSEC(YES) in the server region, authorize these userids to access the server program that is being linked to by the mirror. If the server program accesses any CICS resources, authorize the same userids to access them. If the server program invokes any SP-type

commands, and the mirror transaction is defined with CMDSEC(YES) in the server region, authorize the same userids to access the commands.

If the mirror transaction cannot be attached because of security reasons, the NOTAUTH condition is not raised, but the TERMERR condition is returned to the issuing application in the client region. If the mirror transaction is successfully attached, but it is not authorized to link to the distributed program in the server region, the NOTAUTH condition is raised. The NOTAUTH condition is also raised if the server program fails to access any CICS resources for security reasons.

The server program is restricted to a DPL subset of the CICS API commands when running in a server region. The commands that are not supported include some that return security-related information. For programming information about which commands are restricted, see the *CICS Application Programming Reference* manual. For further information about DPL, refer to the *CICS Intercommunication Guide*.

Security checking done in AOR with MRO

This section summarizes how security checking is done in the AOR.

The userid of the front-end CICS region is assigned as the default. However, if a USERID is specified on the SESSIONS definition, and a link check is done, the userid actually used is the one on the SESSIONS definition.

The region userid referred to in Table 30 through Table 31 is the USERID on the SESSIONS definition. The userid referred to in this case is the one under which the job is running. This userid is the one normally returned by the security manager domain.

With ATTACHSEC(LOCAL) specified

Table 30 shows how checking is done in the AOR when ATTACHSEC(LOCAL) has been specified.

Table 30. Security checking done in AOR—ATTACHSEC(LOCAL) specified

Region userid for AOR	Userid in session definition	Region userid for TOR	Checking in AOR
USERIDA	Not specified	USERIDA	Check against AOR DFLTUSER
USERIDA	USERIDA	Anything	Check against AOR DFLTUSER
USERIDA	Not specified	USERIDB	Check against USERIDB
USERIDA	USERIDB	Anything	Check against USERIDB

With ATTACHSEC(IDENTIFY) specified

Table 31 shows how checking is done in the AOR when ATTACHSEC(IDENTIFY) has been specified.

Table 31. Security checking done in AOR—ATTACHSEC(IDENTIFY) specified

Region userid for AOR	Userid in session definition	Region userid for TOR	Checking in AOR
USERIDA	Not specified	USERIDA	FMH-5 ATTACH check only
USERIDA	USERIDA	Anything	FMH-5 ATTACH check only
USERIDA	Not specified	USERIDB	FMH-5 ATTACH check and USERIDB
USERIDA	USERIDB	Anything	FMH-5 ATTACH check and USERIDB

Summary of resource definition options for MRO security

The following is a summary of the resource definition options you need to define for MRO security:

- On the CONNECTION definition:
  - ATTACHSEC, with either of the following options:

    IDENTIFY

    LOCAL
- On the SESSIONS definition:
  - USERID

For guidance on defining CONNECTION and SESSION resources, see the *CICS Resource Definition Guide*.

Chapter 16. Security for shared data tables

This chapter describes how to provide security for CICS shared data tables. It covers the following topics:

- "Overview"
- "Security checking"
- "LOGON security check" on page 158
- "CONNECT security checks" on page 158

Overview

To provide security for a shared data table when **cross-memory services** are used, you must ensure that:

- The file-owning region (FOR) cannot be impersonated. You can prevent this by checking at LOGON time that the FOR is allowed to log on with the specified generic APPLID of the CICS system.
- An application-owning region (AOR) cannot gain access to data that it is not meant to access. You can prevent this by checking at CONNECT time that the AOR is allowed access to the FOR and, if file security is in force, that the AOR is allowed access to the requested file.

These security checks are performed by using the system authorization facility (SAF) to invoke the ESM.

Note: A region is still able to use data tables locally even if it does not have authority to act as a shared data table server.

The CICS shared data tables (SDT) facility reproduces the main characteristics of function-shipping security that operate at the region level, but you should note the following differences:

- SDT does not provide any mechanism for the FOR to perform security checks at the transaction level (there is no equivalent of ATTACHSEC(IDENTIFY) or ATTACHSEC(VERIFY)). Therefore, if you consider that the transaction-level checks performed by the AOR are inadequate for some files, ensure that those files are not associated with data tables in the FOR.
- SDT does not support any equivalent of preset security on SESSIONS, because no sessions are used.
- SDT does not pass any installation parameter list (INSTLN) information to the security user exits.

Security checking

You should consider the implications of the security checks before sharing a file that is associated with a data table.

SDT security makes use of existing CICS file security definitions, but it also relies on treating server application names (generic APPLIDs) as protected resources. An SDT server's generic APPLID is represented by a DFHAPPL.*applid* profile in the FACILITY resource class.

LOGON security check

To minimize the risk that an AOR might accept **counterfeit data records** from an FOR which is in fact an **imposter**, LOGON processing includes a security check to verify that the FOR is authorized to act as a server with the specified application name. This check is never bypassed, even when SEC=NO is specified at system initialization.

To act as a server for a protected APPLID named *applid*, a CICS region's userid must have UPDATE (or higher) access to DFHAPPL.*applid* in the FACILITY class. See Table 12 on page 66.

When a region attempts to logon as a server, SDT calls the system authorization facility (SAF) to check whether its userid has the required access authority.

If SAF neither grants nor refuses an access request: If a security profile for a specified resource is not retrieved, SAF neither grants nor refuses the access request. In this situation:

- SDT rejects the LOGON request if a security manager is installed but is either temporarily inactive or inoperative for the duration of this VSE IPL. This decision is made on the grounds that had the security manager been active it might have retrieved a profile that refuses access.
- SDT allows the LOGON to proceed if:
  - There is no security manager at all.
  - There is an active security manager but the FACILITY class is undefined or inactive.
  - There is no profile covering the APPLID in question.

  The LOGON is allowed in these cases because there is no evidence that you want to control access to this particular APPLID.

CONNECT security checks

The security checks performed at CONNECT time provide two levels of security:

- **Bind security** allows an FOR that runs without CICS file security to be able to restrict shared access to selected AORs. (Running without file security minimizes runtime overheads and the number of security definitions.)
- **File security** can be activated in the FOR if you want SDT to implement those checks that apply to the AOR as a whole.

But note that SDT provides no way of implementing those security checks that an FOR makes at the transaction level when ATTACHSEC(IDENTIFY) or ATTACHSEC(VERIFY) is used with function shipping.

Bind security

To be allowed shared access to any of an FOR's data tables, an AOR's userid needs READ (or higher) access to the FOR's DFHAPPL.*applid* in the FACILITY class. This check is never bypassed, even when SEC=NO is specified at system initialization. Cases when SAF neither grants nor refuses access are resolved in the same way as for server LOGON (see "If SAF neither grants nor refuses an

access request"). If the result is a refusal, CICS does not permit shared access by the AOR to this APPLID.

Note that controlling LOGON and bind by using different (but hierarchical) levels of access to the same resource has the following consequences:

- Any region with the same userid as a server can always bind to that server.
- It is impossible to control which userids can bind to a given APPLID without also controlling which userids can log on as servers for that APPLID.

SDT bind-time security uses different definitions from those employed by ISC and (if using preset sessions) MRO. So, unless you make them consistent, SDT access might be granted when function shipping attempts are rejected, or vice versa. Both MRO and SDT use the same class and so, with ISC only, SDT CONNECT security might react to changes in security definitions either earlier or later than function shipping.

If file security is not in force in the FOR (that is, if SEC=NO or XFCT=NO was specified at system initialization), an AOR that is allowed to bind to an FOR is also allowed to access all that FOR's shared data tables.

If file security is in force, an AOR that is allowed to bind is still allowed free access if the userids of the AOR and FOR are the same (undefined userids are not considered to be the same).

File security

When file security is in force in an FOR, and the userid of the AOR is undefined, a CONNECT request fails unless the FOR's default userid (specified by the DFLTUSER system initialization parameter) is allowed to read the specified file.

When file security is in force in an FOR, and the userid of the AOR is known but is different from that of the FOR, SDT first checks whether the AOR's userid is allowed to sign on to the FOR's application (that is, whether its userid has READ access to entity APPLID in the APPL class). Cases when SAF neither grants nor refuses the request are resolved in the same way as for server LOGON (see "If SAF neither grants nor refuses an access request" on page 158).

If the userid is allowed to sign on to the FOR's application, the CONNECT request succeeds unless the AOR's userid is not allowed to read the specified file. Otherwise, the CONNECT request is treated in the same way as when the AOR's userid is undefined.

Function shipping detects that an AOR's access to a file has been revoked when a rebuild of the file control resource class is completed in the FOR. However, if a valid connection already exists, SDT continues to allow access until something causes the connection to be broken.

Caution: If you use ISC instead of MRO for function shipping, ensure that the value of the SECURITYNAME parameter in the FOR is the same as the userid of the AOR. Otherwise, the SDT CONNECT and function shipping security checks will be inconsistent.

Chapter 17. Security for the Report Controller facility and CICS SPOOL interface

This chapter gives an overview of how security works for the Report Controller facility and the CICS SPOOL interface. It consists of the following topics:

- "ESM requirement"
- "Retention of RSL"
- "Mapping of RSL values to ESM resource names"
- "Report security checking" on page 162
- "Controlling RCF printers." on page 162
- "Report Browse" on page 163

ESM requirement

The lack of any resource security in the Basic Security Manager mandates the use of a full-function ESM if reports (spool files) are to be secured. Without a full-function ESM, reports are **not secured**. This requirement applies to the other security facilities of the RCF, such as the control of which printers can print which reports, and which end users can control RCF printers.

Retention of RSL

For application source compatibility, and to allow reports created on CICS Transaction Server to be processed correctly on earlier CICS/VSE releases (and vice versa), it is necessary to partially retain the RSL concept of CICS/VSE. This means that the RSL keyword on SPOOLOPEN REPORT is still available and the SPOOLPRTRSL attribute on the printer TERMINAL definition continues to be supported. The specified values are used to derive resource names for use with the ESM. All other use of RSL in CICS is dropped and replaced with ESM facilities.

Mapping of RSL values to ESM resource names

To provide resource security level (RSL) compatibility in an environment managed by external security, the report controller requires the following three sets of resource names to be defined to the ESM:

1. **For REPORT security**: DFHRCF.RSL01 through DFHRCF.RSL24 and DFHRCF.RSLPU
2. **For report BROWSE security**: DFHRCF.BRSL01 through DFHRCF.BRSL24 and DFHRCF.BRSLPU
3. **For PRINTER security**: DFHRCF.PRSL01 through DFHRCF.PRSL24 and DFHRCF.PRSLPU

In each case, 01-24 and PU (public) correspond to an RSL specification on either the SPOOLOPEN command or, in the case of printer security, the SPOOLPRTRSL value on the printer TERMINAL definition.

Relationship to an ESM

A report is **not** a CICS resource. It may have been created by a CICS application, but it is just as likely that it was not. That is, it may be a batch-created spoolfile. The data owned and stored by another VSE/ESA component, VSE/POWER. Therefore, it does not properly belong to that set of resource classes supported for CICS by ESMs (files, programs, etc.). For this reason, a report is classed as a **non-CICS resource** from the standpoint of resource class checking.

All the resource names listed, in an ESM context, must be defined in the general purpose FACILITY class. This is an IBM RACF® term; vendor products which are SAF/RACF compliant offer the equivalent of this resource class. Individual users and group profiles can then be permitted access to any required subset of these with the appropriate access scope.

Report security checking

Reports are secured with the DFHRCF.RSL*nn* resource name where *nn* is the RSL value of the report. The report RSL value is specified on the EXEC CICS SPOOLOPEN REPORT commands. For more information about the EXEC CICS SPOOLOPEN commands, see the *CICS Application Programming Reference* manual. You can also specify an RSL value for a report created by a batch program (see the *CICS Report Controller Planning Guide*), and when creating a report from a transient data queue (CEMS option 4).

There are several places where this resource name is used:

- **Appending to or reading an existing report**. Whenever a program attempts to append to a report (log and resumable reports), the ESM is called to ensure that the user has the necessary UPDATE authority to DFHRCF.RSL*nn*. The same check is performed when using a SPOOLOPEN INPUT command from an application program for the purposes of reading a spoolfile.
- **Accessing report characteristics**. When a CEMS user attempts to look at or update the characteristics of a report (using option 2 from the report list panel), the ESM is called to ensure the user has the necessary UPDATE authority to DFHRCF.RSL*nn*.

Printing a report.

When printing a report on a CICS terminal printer, the userid of the terminal printer must have READ authority to DFHRCF.RSL*nn*.

Controlling RCF printers.

A printer intended for RCF use can be given an RSL value for control purposes. This is specified with the SPOOLPRTRSL keyword in the terminal definition for the printer. For more information, see the *CICS Resource Definition Guide*. Users of the CEMS/CEOS transactions must have UPDATE authority to DFHRCF.PRSL*nn* to be able to control the printer where *nn* is the SPOOLPRTRSL value of the printer.

Report Browse

While an end user may have authority to see and manipulate the **characteristics** of a report, it does follow that access to the **content** of the report should be permitted. In previous releases of the report controller, authority to browse the contents of a report was controlled using the RSL of a dummy program, DFHPSBRS, which had to match one of the user's RSL keys. In CICS Transaction Server for VSE/ESA Release 1, users attempting to browse a report (option 8 on the report list panel) must have READ access to DFHRCF.BRSL*nn*, where *nn* is the RSL value of the report.

Part 4. Customization

This section consists of Chapter 18, “Customizing security processing” on page 167 and discusses customizing the CICS-ESM interface, in the following:

- “Installation data parameter list” on page 167
- “Determining the userid of the CICS region” on page 169
- “Specifying user-defined resources to the ESM” on page 170

Chapter 18. Customizing security processing

Product-sensitive Programming Interface information

This chapter introduces you to the CICS-ESM interface, and describes how the VSE/ESA router passes control to the ESM. It describes how ESM exit programs can access CICS-related information. Finally, it lists the control points at which CICS invokes the ESM. The chapter is organized as follows:

- "Installation data parameter list"
- "Determining the userid of the CICS region" on page 169
- "Specifying user-defined resources to the ESM" on page 170

For programming information on customizing the CICS-ESM interface (using a compatible user-written or vendor-supplied ESM), see the *CICS Customization Guide*.

Installation data parameter list

The installation data parameter list gives your ESM exit programs access to the following information:

- CICS security event being processed
- Details of the current CICS environment, as available
  - APPLID of the CICS region
  - Common work area
  - Transaction being invoked
  - Program being executed
  - CICS terminal identifier
  - VTAM LUname
  - Terminal user area
  - An 8-byte communication area, whose usage is described in the *CICS Customization Guide*.

For programming information about user-written ESMs, see the *CICS Customization Guide*.

CICS security control points

This section summarizes the RACROUTE macros used by CICS to invoke the ESM, and the control points at which they are issued.

Some of these calls may not always be issued, because CICS reuses entries for users already signed on.

RACROUTE
This is the "front end" to the macros described below; it invokes the VSE/ESA router.

RACROUTE REQUEST=VERIFY

This macro is issued at operator sign-on (with the parameter ENVIR=CREATE), and at signoff (with the parameter ENVIR=DELETE). It creates or destroys an ACEE (access control environment element). It is issued at the following CICS control points

Each of the following control points relates to ENVIR=CREATE:

- Normal sign-on through EXEC CICS SIGNON
- Sign-on of the default userid DFLTUSER
- Sign-on of preset-security terminal
- Sign-on of MRO session
- Sign-on of LU6.1 session
- Sign-on of LU6.2 session (or APPC session)
- Sign-on for XRF tracking of any of the above
- Sign-on associated with the userid on an attach request (for all operands of ATTACHSEC except LOCAL).

Each of the following control points relates to ENVIR=DELETE:

- Normal sign-off through EXEC CICS SIGNOFF
- Sign-off when deleting a terminal
- Sign-off when TIMEOUT expires
- Sign-off of MRO session
- Sign-off of LU6.1 session
- Sign-off of LU6.2 session (or APPC session)
- Sign-off for XRF tracking of any of the above.
- Sign-off associated with the userid on an attach request (for all operands of ATTACHSEC except LOCAL).

RACROUTE REQUEST=VERIFYX

This macro creates and deletes an ACEE in a single call. It is issued at the following control points:

- Sign-on, as an alternative to VERIFY, when an optimized sign-on is performed for subsequent attach sign-ons across an LU6.2 link with ATTACHSEC(VERIFY).
- When an invalid password or PassTicket is presented, or EXEC CICS VERIFY PASSWORD command is issued.

RACROUTE REQUEST=FASTAUTH

This macro is issued during resource checking, on behalf of a user who is identified by an ACEE. It is the high-performance form of REQUEST=AUTH, using in-storage resource profiles, which does not cause auditing to be performed. It is issued at the following CICS control points:

- When attaching a local transaction
- When checking link security for transaction attach
- Transaction validation for an MRO task
- CICS resource checking
- Link security check for a CICS resource
- Transaction validation for EDF
- Transaction validation for the transaction being tested (by EDF)
- When checking a surrogate user authority
- QUERY SECURITY with the RESTYPE option.

RACROUTE REQUEST=AUTH

This macro provides a form of resource checking with a larger pathlength, and causes auditing to be performed. It is used as follows:

- After a call to FASTAUTH indicates an access failure that requires logging.
- When a QUERY SECURITY request with the RESCLASS option is used. This indicates a request for a resource for which CICS has not built in-storage profiles. If CICS has built in-storage profiles, REQUEST=AUTH uses them.

RACROUTE REQUEST=LIST

This macro is issued to create and delete the in-storage profile lists needed by REQUEST=FASTAUTH. (One REQUEST=LIST macro is required for each resource class.) It is issued at the following CICS control points:

- When CICS security is being initialized
- When an EXEC CICS PERFORM SECURITY REBUILD command is issued
- When XRF tracks either of these events.

RACROUTE REQUEST=EXTRACT

This macro is issued (with the parameters SEGMENT=CICS,CLASS=USER, with the parameters and with the SEGMENT=BASE,CLASS=USER to obtain the national language and user name) at all the following control points:

- Normal sign-on through EXEC CICS SIGNON
- Sign-on of the default userid DFLTUSER
- Sign-on of preset security terminal
- Sign-on of MRO session
- Sign-on of LU6.1 session
- Sign-on of LU6.2 session (or APPC session)
- Sign-on for XRF tracking of any of the above
- Sign-on associated with the userid on an attach request (for all operands of ATTACHSEC except LOCAL).

It is also issued (with the parameters SEGMENT=SESSION,CLASS=APPCLU) during verification of LU6.2 (APPC) bind security, at the CICS control point for bind of an LU6.2 (APPC) session.

Note: Any customization must be done using the VSE/ESA router exit, ICHRTX00. For a detailed description of these macros, see the *RACF External Security Interface (RACROUTE) Macro Reference for VSE*.

Determining the userid of the CICS region

CICS makes use of the userid of the region in which it runs for the following purposes:

- To prefix resource names if SECPRFX=YES is specified. For more information about the SECPRFX system initialization parameter, see “SECPRFX” on page 25.
- As the user to be checked for category 1 transactions. For more information, see “Category 1 transactions” on page 81.

- As the default PLTPI user for PLTPI non-terminal security, if a PLTPIUSR is not specified in the system initialization parameter.
- For SURROGAT checking (for example, authority to use the PLTPI and default userids).
- For MRO bind security. For more information, see Chapter 15, “Implementing MRO security” on page 143.

CICS obtains the region userid by invoking the external security manager, which extracts it from the ESM control blocks relevant for the job. The security domain and MRO-bind security each obtain the region userid by issuing a RACROUTE REQUEST=EXTRACT macro.

Specifying user-defined resources to the ESM

If you want to use the QUERY SECURITY command with the RESCLASS option, you may need to create user-defined resources within user-defined classes to represent the non-CICS resources that you want to query.

Designing applications to use the user-defined resources

This topic gives an example of how you might design applications to make use of the user-defined resources.

Your applications use CICS file control in the normal way to read records from the pay and personal details file. Because you are controlling individual fields within each record, you may not need to apply resource security at the file level, so your transactions can be defined with RESSEC(NO). After reading the file record, but before displaying the results, you use QUERY SECURITY to determine whether the user has the authority to access the particular field within the record. For instance, before displaying the salary amount, you issue:

```
EXEC CICS QUERY SECURITY RESCLASS('$FILEREC')
                         RESID('PAYFILE.SALARY')
                         RESIDLENGTH(14)
                         READ(read_cvda)
```

Then, depending on the value returned in `read_cvda`, your application either displays the salary or a message stating that the user is not authorized to display it. Likewise, as part of a transaction that updates a person’s telephone number, you issue:

```
EXEC CICS QUERY SECURITY RESCLASS('$FILEREC')
                         RESID('PERSONAL.PHONE')
                         RESIDLENGTH(14)
                         UPDATE(update_cvda)
```

If the value returned in `update_cvda` indicates that the user has UPDATE access, the transaction can continue and update the telephone number in the file. Otherwise, it should indicate that the user is not authorized to update the telephone number.

Global user exits in signon and signoff

CICS provides the XSNON global user exit in EXEC CICS SIGNON processing and the XSNOFF global user exit in EXEC CICS SIGNOFF processing. These exits do not allow you to affect the result of the sign-on or sign-off, but notify you when the userid associated with a terminal changes. The exits are further described in the *CICS Customization Guide*.

End of Product-sensitive Programming Interface information

Part 5. Migration and coexistence

This part describes migration implications of certain security-related features introduced in CICS Transaction Server for VSE/ESA Release 1

Various aspects of coexistence from a security viewpoint are discussed.

Chapter 19. Migration considerations

The full implications of migrating to an ESM are discussed in the *CICS Migration Guide*, which introduces the Security Migration Aid (SMA).

The SMA is a utility designed to assist you in the migration of CICS internal security information to an external security manager required for CICS Transaction Server for VSE/ESA Release 1. The CICS security migration aid (SMA) extracts the following security information from your current CICS systems into a VSAM key sequenced data set (KSDS) file:

- Resource security values for transient data queues
- Resource security values for files and data tables
- Resource security values for journals
- Resource and transaction security values for transactions
- Resource security values for programs and mapsets
- User profile information from the CICS sign on table and the VSE/ESA interactive user interface (IUI) control file
- User profile information for terminals defined with preset terminal security
- SPRTRSL for terminal printers
- Resource security values for secured temporary storage queues

The SMA analyzes this data and advises you of areas that need attention. For more information about this process, see the the *CICS/VSE Security Migration Guide*, SC33-1406.

Resource groups and transaction groups

To migrate your CICS internal security to an external security management system you must pay particular attention to your allocation of resource and transaction groupings.

You should analyze your current security setup to ensure that you are not using more resource security levels (RSL) or transaction security levels (TRANSEC) than are really required. A good working base from which to start is to assign security keys by application, and resource security levels by functional area.

CICS manages assets application programs, application data, and application output. To prevent disclosure, corruption, or destruction of these assets, you must safeguard the CICS system components themselves.

CICS provides a variety of security and control mechanisms. They limit the activities of CICS terminal users to only those functions that any particular individual user is authorized to use. They are:

- Terminal signon/preset terminal security
- Transaction security
- Resource security

CICS itself does **not** provide facilities to protect its own assets from external access. You should restrict access to program libraries and CICS partitions, and authorize access only to those programmers responsible for incorporating approved application and system changes. Similarly, the data sets and databases used by CICS and by CICS applications must be accessible only by approved batch processing and operations procedures.

CICS does not protect your system from application programs that use undocumented or unsupported interfaces to bypass CICS security. You are responsible for ensuring that such programs are not installed on your system.

CICS does not protect your application source libraries. You should ensure that procedures are established and followed that prevent the introduction of unauthorized or untested application programs into your “production” application base. You should also protect the integrity of your system by exercising control over libraries that are admitted to the system and any changes made to those libraries.

The security manager is accessed through the operating system supplied system authorisation facility (SAF), using RACROUTE macros placed at strategic points within CICS.

The SMA is a menu-driven online facility that extracts security information from the control blocks of the CICS system from which it is run, and from the IUI control file. You can display the data extracted by the facility. In effect all you need to do is to run the transaction on your CICS Transaction Server for VSE/ESA Release 1 system.

The output from this transaction, a VSAM KSDS file, can be used as input in to a stage two utility provided by the ESM. The stage two utility reads the file and creates the necessary definitions on the ESM database.

You select one of the following options from the main menu:

Replace all data Update all data currently held for that system.

Update all data Run a complete extract of the security data for the current CICS system.

Selective replace Run a partial extract of the security data for the current CICS system, based on resource type, replacing all data of this type for this system.

Selective update Run a partial update of the security data for the current CICS system, based on resource type, updating all data of this type for this system, by additions and amendments.

| The SMA collects your security definitions, which currently reside in numerous
| CICS tables, into a central repository - VSAM dataset DFHXSMA. In the process
| of creating this repository, the data is analyzed to highlight any conflicts which may
| cause problems when the data is used by the stage two facility.

For more information about the SMA, see the *CICS/VSE Security Migration Guide*, SC33-1406.

Part 6. Problem determination

This section consists of the chapter Chapter 20, “Problem determination in a security environment” on page 179, and considers the following aspects of problem determination in a security environment as follows:

- “Resolving problems when access is denied incorrectly” on page 179
- “Resolving problems when access is allowed incorrectly” on page 181
- “CICS initialization failures related to security” on page 182
- “Password expiry management problem determination” on page 185.

Chapter 20. Problem determination in a security environment

This chapter provides information to help you find the causes of access authority problems. It covers the following topics:

- "Resolving problems when access is denied incorrectly"
- "Resolving problems when access is allowed incorrectly" on page 181
- "CICS initialization failures related to security" on page 182
- "Password expiry management problem determination" on page 185

Resolving problems when access is denied incorrectly

When a user requires access to a protected resource (such as a CICS transaction) and the ESM denies the requested access, you will often have to analyze the problem before deciding what action to take.

The basic points to ensure are that:

- CICS is using the ESM for this particular kind of resource.
- You know which definition the ESM is using to check the user's authority.
- You know which userid CICS is using for the authorization check.

For each security violation, CICS issues messages:

- CICS issues an authority message to the terminal user (or returns a "not authorized" return code to an application).
- CICS sends a message DFHXS1111 to the CSCS transient data destination.

Note: You can use the CICS-supplied message domain global user exit, XMEOUT, to reroute CICS-issued authorization messages. For programming information about using XMEOUT, see the *CICS Customization Guide*.

Is CICS using an ESM for this particular kind of resource?

- Is CICS using an ESM?

  Make sure that CICS is using an ESM. If it is not using an ESM, it issues message DFHXS1102.

- Is security checking done for the particular general resource class? Message DFHXS1105 tells you if the class named on an X*name* parameter has been initialized.

  Note: If message DFHXS1105 is not there, ensure that the SEC=YES system initialization parameter is specified for the region.

  Check the appropriate CICS system initialization parameter for the resource. For example: for transactions, this is the XTRAN parameter.

Which profile is the ESM using?

- If CICS prefixing is in effect for the CICS region involved, the prefix specified is used as the first qualifier of ESM resource profiles (or member names).
  - Make sure that you have specified the correct prefix as part of resource profile definition.
  - Make sure the CICS job is running under the correct prefix if SECPRFX=YES is specified.
  - Make sure that an installation-written SAF exit is not changing the effective userid under which the CICS region is running.

Which userid did CICS supply for the authorization check?

Check to see if the user reporting the problem has signed on to CICS. If the user has not signed on to CICS, one of the following could be occurring:

- If you are using preset-terminal security, the authorization could be related to that terminal's userid.
- The user could be trying to operate as the CICS default user (without signing on to CICS).
- If the transaction was initiated by a START command, the userid could be inherited from the transaction issuing the START, or specified on the START command itself.
- If the transaction was initiated as the trigger transaction associated with a transient data queue, the userid could have been specified in the DCT for the queue.
- If the program is running as a PLTPI program, the userid could be specified in the PLTPIUSR system initialization parameter.

For help in identifying the user, see Table 2 on page 10.

Which profile is used to protect the resource?

If you are using generic profiles (and you are **not** using resource group profiles), only the most specific profile is used. For example, if the following profiles exist:

```
**
C*
CE*
CEDA
```

CEDA is the profile that is used to control access to the CEDA transaction. If you delete profile CEDA and refresh the in-storage copies, CE* is used.

Note: This assumes CICS prefixing is not used and generic profile checking **is** used.

If resource group profiles have been defined in the relevant class (for example, profiles in the GCICSTRN class), it is possible that more than one profile is used in determining a user's access. Determine which profiles protect the resource.

Resolving problems when access is allowed incorrectly

There could be many reasons why a user might have access to a protected resource, even when you think that the user should **not** have that access. Here are some checks that you can make to investigate this kind of situation:

- Confirm which userid the user is signed on as. (Make sure the user has actually signed on and is not acting as the CICS default user.) You can ask the user to sign off, then sign on again. You can also ask the user to issue EXEC CICS ASSIGN or EXEC CICS INQUIRE TERMINAL, which can be issued with the CECI transaction (or a user-written transaction).
- Make sure that the SEC system initialization parameter is SEC=YES for the CICS region the user is signed on to.

  If SEC=NO is specified, users can access any resource.
- If the user is running a transaction that communicates with other regions such as application-owning regions (AORs) or file-owning regions (FORs), make sure that the SEC system initialization parameter is SEC=YES for those regions.
- Is prefixing correct?
  - Has the CICS JOB been submitted by the correct USER?
  - Is SECPRFX set correctly?
  - Has an installation-written SAF exit been used to return a different CICS region userid when RACROUTE=EXTRACT has been specified?
- Depending on the resource, make sure that RESSEC(YES) is specified for each transaction that might access that resource.
- Is the appropriate X*name* CICS system initialization parameter correctly set?

  For example, if it is a file control request, is XFCT=YES or XFCT=*value* specified? If the X*name* parameter specifies a value other than YES or NO, does the value show the correct installation-defined class name?
- Is the transaction exempt from transaction security? (For information on transactions that may have been defined in this way, see “Category 3 transactions” on page 85.)
- Does the transaction have the correct RESSEC and CMDSEC options?
- Check that the RESSEC setting on the MIRROR transaction is correct.
- If the resource is temporary storage, are you using the correct TST? Check:
  - The DFHTST TYPE=SECURITY entry in the TST
  - That TST entries are in the correct order
- If intersystem communication is involved, check the following:
  - Is a SECURITY REBUILD required (on this or on the remote system)?
  - If ATTACHSEC=LOCAL is specified, does the SECURITYNAME userid have access to the resource?
  - Is ATTACHSEC=IDENTIFY specified?
  - Are ‘equivalent systems’ causing link security to be bypassed
  - Is the remote system using the same security database?
- Do you have any installation exits?

- Check the profile that you think protects the resource.

CICS initialization failures related to security

If SEC=YES is specified, external security is *required*. If external security cannot be provided, CICS cannot be initialized.

Figure 27 on page 183 shows an example of a failure to initialize.

If security initialization fails:

- Examine the DFHXS1106 message return codes. In the example shown in Figure 27 on page 183, SAF return code X'00000004' and reason code X'00000000' were issued:

  A return code of X'00000004' indicates that an error occurred in the security router (RACROUTE). See the RACROUTE macro reference in "CICS security control points" on page 167.

- Check the CICS startup options, in particular the X*name* system initialization parameters. Make sure that:
  - The class is defined to the ESM and is active.

SAF or ESM installation exits

Check if any SAF or ESM installation exits are causing initialization requests to fail.

CICS default user fails to sign on

Figure 28 on page 184 shows an example of a CICS job log when the DFLTUSER fails to sign on. CICS is started with system initialization parameters, SEC=YES and DFLTUSER=ORMAN. User profile ORMAN has not been defined to the ESM.

This CICS region cannot be initialized because, with SEC=YES specified, external security is required and the default user must be defined to the ESM.

```
DFHPA1927  IYCTZCCA AKPREQ=0
DFHPA1927  IYCTZCCA APPLID=IYCTZCCA
DFHPA1927  IYCTZCCA FCT=NO
DFHPA1927  IYCTZCCA SIT=$$
DFHPA1927  IYCTZCCA START=INITIAL
DFHPA1927  IYCTZCCA GRPLIST=USERLIST
DFHPA1927  IYCTZCCA PLTPI=NO
DFHPA1927  IYCTZCCA SEC=YES
DFHPA1927  IYCTZCCA XCMD=NO
DFHPA1927  IYCTZCCA XDCT=1CVFDCT
DFHPA1927  IYCTZCCA XFCT=1CVFFCT
DFHPA1927  IYCTZCCA XJCT=1CVFJCT
DFHPA1927  IYCTZCCA XPCT=1CVFPCT
DFHPA1927  IYCTZCCA XPPT=UNKNOWN
DFHPA1927  IYCTZCCA XTST=1CVFTST
DFHPA1927  IYCTZCCA XTRAN=1CVFTRN
DFHPA1103  IYCTZCCA END OF FILE ON SYSIPT.
DFHTR0103 TRACE TABLE SIZE IS 64K
DFHSM0122I IYCTZCCA Limit of DSA storage below 16MB is 5,120K.
DFHSM0123I IYCTZCCA Limit of DSA storage above 16MB is 20M.
DFHSM0113I IYCTZCCA Storage protection is not active.
DFHDM0101I IYCTZCCA CICS is initializing.
DFHSI1500 IYCTZCCA CICS startup is in progress for CICS Transaction
Server Version 1.1.0.
DFHXS1100I IYCTZCCA Security initialization has started.
DFHSI1501I IYCTZCCA Loading CICS nucleus.
DFHXS1105 IYCTZCCA Resource profiles for class A1CVFPCT have been built.
DFHXS1105 IYCTZCCA Resource profiles for class D1CVFDCT have been built.
DFHXS1105 IYCTZCCA Resource profiles for class F1CVFFCT have been built.
DFHXS1105 IYCTZCCA Resource profiles for class J1CVFJCT have been built.
+DFHXS1106 IYCTZCCA
Resource profiles could not be built for class MUNKNOWN. CICS is
terminated. SAF codes are (X'00000004',X'00000000'). ESM codes are
(X'00000000',X'00000000').
DFHKE1800 IYCTZCCA ABNORMAL TERMINATION OF CICS IS COMPLETE.
```

Figure 27. Security initialization failure

```
DFHPA1927  IYCTZCCE SEC=YES
DFHPA1927  IYCTZCCE XUSER=YES
DFHPA1927  IYCTZCCE DFLTUSER=ORMAN
DFHPA1927  IYCTZCCE XCMD=NO
DFHPA1927  IYCTZCCE XDCT=1CVFDCT
DFHPA1927  IYCTZCCE XFCT=1CVFFCT
DFHPA1927  IYCTZCCE XJCT=1CVFJCT
DFHPA1927  IYCTZCCE XPCT=1CVFPCT
DFHPA1927  IYCTZCCE XFCT=1CVFPPT
DFHPA1927  IYCTZCCE XTST=1CVFTST
DFHPA1927  IYCTZCCE XTRAN=1CVFTRN
DFHPA1103  IYCTZCCE END OF FILE ON SYSIPT.
DFHTR0103 TRACE TABLE SIZE IS 64K
DFHSM0122I IYCTZCCE Limit of DSA storage below 16MB is 5,120K.
DFHSM0123I IYCTZCCE Limit of DSA storage above 16MB is 20M.
DFHSM0113I IYCTZCCE Storage protection is not active.
DFHDM0101I IYCTZCCE CICS is initializing.
DFHSI1500 IYCTZCCE CICS startup is in progress for CICS Transaction
Server Version 1.1.0.
DFHXS1100I IYCTZCCE Security initialization has started.
DFHDU0304I IYCTZCCE Transaction Dump Data set DFHDMPA opened.
DFHSI1501I IYCTZCCE Loading CICS nucleus.
DFHXS1105 IYCTZCCE Resource profiles for class A1CVFPCT have been built.
DFHXS1105 IYCTZCCE Resource profiles for class D1CVFDCT have been built.
DFHXS1105 IYCTZCCE Resource profiles for class F1CVFFCT have been built.
DFHXS1105 IYCTZCCE Resource profiles for class J1CVFJCT have been built.
DFHXS1105 IYCTZCCE Resource profiles for class M1CVFPPT have been built.
DFHXS1105 IYCTZCCE Resource profiles for class S1CVFTST have been built.
DFHXS1105 IYCTZCCE Resource profiles for class T1CVFTRN have been built.
DFHXS1105 IYCTZCCE Resource profiles for class SURROGAT have been built.
+DFHXS1104 IYCTZCCE
Default security could not be established for userid ORMAN.  The
security domain cannot continue, so CICS is terminated.  SAF codes are
(X'00000004',X'00000000'). ESM codes are (X'00000004',X'00000000') .
DFHKE1800 IYCTZCCE ABNORMAL TERMINATION OF CICS IS COMPLETE.
```

Figure 28. Example of CICS job log if DFLTUSER fails to sign on

Revoked user attempting to sign on

The following example sequence illustrates what happens when a revoked user attempts to sign on:

1. User USR001 attempts to sign on using CESN. However, the user is revoked. The user sees the following on the terminal:

   ```
   DFHCE3546 Your signon userid has been revoked. Signon is terminated.
   ```

2. A CICS message is sent to the CSCS transient data queue:

   ```
   DFHSN1120 26/04/99 12:20:24 CICSSYS1 Signon at netname D2D1
   with userid USR001 failed because the userid has been revoked.
   ```

User has insufficient authority to access a resource

Now let us consider user USR001, who has signed on successfully with current connect group GRP001. User USR001 attempts unsuccessfully to use transaction CEMT, which is protected by profile CAT2 in class GCICSTRN (the resource group class for CICS transactions), because XTRAN=YES is specified in the CICS system initialization parameters.

1. The terminal user received the following CICS message:

```
DFHAC2033 26/04/99 15:18:44 CICSSYS1 You are not authorized to use
transaction CEMT. Check that the transaction name is correct.
```

2. A CICS message is sent to the CSCS transient data queue:

```
DFHXS1111 26/04/99 13:30:41 CICSSYS1 CEMT Security violation
          by user USR001 at netname D2D1 for resource CEMT in class
          TCICSTRN. SAF codes are (X'00000008',X'00000000'). ESM codes
          are (X'00000008',X'00000000').
```

   The following message is also sent to the CSMT transient data queue:

```
DFHAC2003 26/04/99 15:18:44 CICSSYS1 Security violation has been
detected term id = D2D1, trans id = CEMT, userid = USR001.
```

CICS region user ID access problem

CICS security initialization can fail if the CICS region user ID does not have access to the necessary Category 1 transactions. A message similar to the following is shown:

```
DFHXS1103I CICSAPPL Default security for userid CICSUSER has been established
DFHDUO304I CICSAPPL Transaction dump data set DFHDMPA opened
DFHXS1111 CICSAPPL
  04/28/99 16:05:17 CICSAPPL ???? Security violation by user TESTRGN for resource
  CATA in class TCICSTRN.SAF codes are
(X'00000004',X'00000000').ESM codes are (X'00000004',X'00000000').
DFHXS1113 CICSAPPL
The region userid cannot access system transaction CATA.  CICS will terminate.
SAF codes are (X'00000004',X'00000000').ESM codes are (X'00000004',X'00000000').
```

Password expiry management problem determination

If you are running a CICS-APPC PEM environment, and are not receiving the expected responses, check the following possible sources of errors in the sign-on transaction program:

- The function management header (FMH) may be in error; check that:
  - The conversation type being used is **basic**.
  - The XTRANID in the CICS TRANSACTION definition for CLS4 is X'06F3F0F1'. (See “Setting up the PEM client” on page 124.)

  - The CICS PEM server sign-on transaction is running as a **synclevel 0** transaction. (See “Setting up the PEM client” on page 124.)

- The user data may be in error; check that:
  - Valid lengths are being sent. (See Table 23 on page 128, Table 24 on page 129, and “Format of user data” on page 125.)
  - Userids and passwords are sent in uppercase EBCDIC. (See “Setting up the PEM client” on page 124.)
  - GDS variables (required in basic conversations) are being used: (See “Format of user data” on page 125.)

Note: If the CICS PEM server receives an error in the FMH or user data, it sends an ISSUE ERROR to the PEM requester, and terminates without an abend. If this happens, it is likely that there is an error in the flow. For examples of valid flows, see “Examples of PEM client and CICS PEM server user data” on page 132.

Execution diagnostic facility (EDF): The execution diagnostic facility (EDF) **cannot** be used to check DFHCLS4, for security reasons, because user passwords would be displayed on the EDF screens.

Part 7. Appendix

The appendix is:

- Appendix A, “Resource and command check cross reference” on page 189.

Appendix A. Resource and command check cross reference

This appendix provides a complete API command and resource check cross reference.

Table 32 (Page 1 of 6). Resource and command check cross reference

EXEC CICS COMMAND	Resource Check			Check class=XCMD	
	Class	Access	Resource	Access	Resource
ABEND					
ACQUIRE				UPDATE	TERMINAL
ADDRESS					
ALLOCATE					
ASKTIME					
ASSIGN					
BIF DEEDIT					
BUILD ATTACH					
CANCEL (see note 1)	XPCT	READ	transid		
CHANGE PASSWORD					
CHANGE TASK					
COLLECT FILE	XFCT	READ	file	READ	STATISTICS
COLLECT JOURNALNUM	XJCT	READ	journal	READ	STATISTICS
COLLECT PROGRAM	XPPT	READ	program	READ	STATISTICS
COLLECT STATISTICS				READ	STATISTICS
COLLECT TDQUEUE	XDCT	READ	tdqueue	READ	STATISTICS
COLLECT TRANSACTION	XPCT	READ	transid	READ	STATISTICS
CONNECT PROCESS					
CONVERSE					
CREATE CONNECTION (see note 2)				ALTER	CONNECTION
CREATE FILE	XFCT	ALTER	file	ALTER	FILE
CREATE LSRPOOL				ALTER	LSRPOOL
CREATE MAPSET	XPPT	ALTER	mapset	ALTER	MAPSET
CREATE PARTITIONSET	XPPT	ALTER	partitionset	ALTER	PARTITIONSET
CREATE PARTNER				ALTER	PARTNER
CREATE PROFILE				ALTER	PROFILE
CREATE PROGRAM	XPPT	ALTER	program	ALTER	PROGRAM
CREATE SESSIONS (see note 3)				ALTER	SESSIONS
CREATE TERMINAL (see note 3)				ALTER	TERMINAL
CREATE TRANCLASS				ALTER	TCLASS
CREATE TRANSACTION	XPCT	ALTER	transid	ALTER	TRANSACTION

Table 32 (Page 2 of 6). Resource and command check cross reference

EXEC CICS COMMAND	Resource Check			Check class=XCMD	
	Class	**Access**	**Resource**	**Access**	**Resource**
CREATE TYPETERM				ALTER	TYPETERM
DELAY					
DELETE	XFCT	UPDATE	file		
DELETEQ TD	XDCT	UPDATE	tdqueue		
DELETEQ TS (see note 4)	XTST	UPDATE	tsqueue		
DEQ					
DISABLE PROGRAM	XPPT	UPDATE	program	UPDATE	EXITPROGRAM
DISCARD AUTINSTMODEL				ALTER	AUTINSTMODEL
DISCARD CONNECTION					
DISCARD FILE	XFCT	ALTER	file	ALTER	FILE
DISCARD JOURNALNAME	XJCT	ALTER	journal	ALTER	JOURNALNAME
DISCARD PARTNER				ALTER	PARTNER
DISCARD PROFILE				ALTER	PROFILE
DISCARD PROGRAM	XPPT	ALTER	program	ALTER	PROGRAM
DISCARD TERMINAL				ALTER	TERMINAL
DISCARD TRANCLASS				ALTER	TCLASS
DISCARD TRANSACTION	XPCT	ALTER	transid	ALTER	TRANSACTION
DUMP TRANSACTION					
ENABLE PROGRAM	XPPT	UPDATE	program	UPDATE	EXITPROGRAM
ENDBR (see note 5)					
ENQ					
ENTER TRACENUM					
EXTRACT					
EXTRACT EXIT	XPPT	READ	program	UPDATE	EXITPROGRAM
FORMATTIME					
FREE					
FREEMAIN					
GDS					
GETMAIN					
HANDLE ABEND PROGRAM	XPPT	READ	program		
HANDLE AID					
HANDLE CONDITION					
IGNORE CONDITION					
INQUIRE AUTINSTMODEL				READ	AUTINSTMODEL
INQUIRE AUTOINSTALL				READ	AUTOINSTALL
INQUIRE CONNECTION				READ	CONNECTION

Table 32 (Page 3 of 6). Resource and command check cross reference

EXEC CICS COMMAND	Resource Check			Check class=XCMD	
	Class	Access	Resource	Access	Resource
INQUIRE DELETSHIPPED				READ	DELETSHIPPED
INQUIRE DSNAME				READ	DSNAME
INQUIRE DUMPDS				READ	DUMPDS
INQUIRE EXITPROGRAM	XPPT	READ	program	READ	EXITPROGRAM
INQUIRE FILE	XFCT	READ	file	READ	FILE
INQUIRE IRC				READ	IRC
INQUIRE JOURNALNUM	XJCT	READ	journal	READ	JOURNALNUM
INQUIRE MODENAME				READ	MODENAME
INQUIRE MONITOR				READ	MONITOR
INQUIRE NETNAME				READ	TERMINAL
INQUIRE PARTNER				READ	PARTNER
INQUIRE PROFILE				READ	PROFILE
INQUIRE PROGRAM	XPPT	READ	program	READ	PROGRAM
INQUIRE REQID (see note 8)	XPCT	READ	transid	READ	REQID
INQUIRE STATISTICS				READ	STATISTICS
INQUIRE STORAGE				READ	STORAGE
INQUIRE SYSDUMPCODE				READ	SYSDUMPCODE
INQUIRE SYSTEM				READ	SYSTEM
INQUIRE TASK				READ	TASK
INQUIRE TCLASS				READ	TCLASS
INQUIRE TDQUEUE	XDCT	READ	tdqueue	READ	TDQUEUE
INQUIRE TERMINAL				READ	TERMINAL
INQUIRE TRACEDEST				READ	TRACEDEST
INQUIRE TRACEFLAG				READ	TRACEFLAG
INQUIRE TRACETYPE				READ	TRACETYPE
INQUIRE TRANCLASS				READ	TRANCLASS
INQUIRE TRANDUMPCODE				READ	DUMPCODE
INQUIRE TRANSACTION	XPCT	READ	program	READ	TRANSACTION
INQUIRE TSQUEUE (see note 4)	XTST	READ	tsqueue	READ	TSQUEUE
INQUIRE VTAM				READ	VTAM
ISSUE					
LINK	XPPT	READ	program		
LOAD	XPPT	READ	program		
MONITOR					
PERFORM DELETSHIPPED				UPDATE	DELETSHIPPED
PERFORM DUMP				UPDATE	DUMP

Table 32 (Page 4 of 6). Resource and command check cross reference

EXEC CICS COMMAND	Resource Check			Check class=XCMD	
	Class	Access	Resource	Access	Resource
PERFORM RESETTIME				UPDATE	RESETTIME
PERFORM SECURITY				UPDATE	SECURITY
PERFORM SHUTDOWN				UPDATE	SHUTDOWN
PERFORM STATISTICS				UPDATE	STATISTICS
POINT					
POP HANDLE					
POST					
PURGE MESSAGE					
PUSH HANDLE					
QUERY SECURITY (see note 6)					
READ	XFCT	READ	file		
READ PREV (see note 5)					
READ NEXT (see note 5)					
READQ TD	XDCT	UPDATE	tdqueue		
READQ TS (see note 4)	XTST	READ	tsqueue		
RECEIVE					
RELEASE	XPPT	READ	program		
RESETBR (see note 5)					
RESYNC ENTRYNAME				UPDATE	EXITPROGRAM
RETRIEVE					
RETURN					
REWRITE	XFCT	UPDATE	file		
ROUTE					
SEND					
SET AUTOINSTALL				UPDATE	AUTOINSTALL
SET CONNECTION				UPDATE	CONNECTION
SET DELETSHIPPED				UPDATE	DELETSHIPPED
SET DSNAME				UPDATE	DSNAME
SET DUMPDS				UPDATE	DUMPDS
SET FILE	XFCT	UPDATE	file	UPDATE	FILE
SET IRC				UPDATE	IRC
SET JOURNALNUM	XJCT	UPDATE	journal	UPDATE	JOURNALNAME
SET MODENAME				UPDATE	MODENAME
SET MONITOR				UPDATE	MONITOR
SET NETNAME				UPDATE	TERMINAL
SET PROGRAM	XPPT	UPDATE	program	UPDATE	PROGRAM
SET STATISTICS				UPDATE	STATISTICS
SET SYSDUMPCODE				UPDATE	SYSDUMPCODE
SET SYSTEM				UPDATE	SYSTEM

Table 32 (Page 5 of 6). Resource and command check cross reference

EXEC CICS COMMAND	Resource Check			Check class=XCMD	
	Class	Access	Resource	Access	Resource
SET TASK				UPDATE	TASK
SET TCLASS				UPDATE	TCLASS
SET TDQUEUE (see note 3)	XDCT	UPDATE	tdqueue	UPDATE	TDQUEUE
SET TERMINAL				UPDATE	TERMINAL
SET TRACEDEST				UPDATE	TRACEDEST
SET TRACEFLAG				UPDATE	TRACEFLAG
SET TRACETYPE				UPDATE	TRACETYPE
SET TRANCLASS				UPDATE	TCLASS
SET TRANDUMPCODE				UPDATE	TRANDUMPCODE
SET TRANSACTION	XPCT	UPDATE	transid	UPDATE	TRANSACTION
SIGNOFF					
SIGNON					
SPOOLCLOSE REPORT					
SPOOLOPEN ESCAPE					
SPOOLOPEN INPUT	FACILITY	UPDATE	DFHRCF		
SPOOLOPEN MAPNAME					
SPOOLOPEN OUTPUT					
SPOOLOPEN REPORT	FACILITY	UPDATE	DFHRCF		
SPOOLOPEN RESUME (see note 9)	FACILITY	UPDATE	DFHRCF		
SPOOLREAD					
SPOOLWRITE					
SPOOLWRITE MAPNAME					
SPOOLWRITE REPORT (see note 9)	FACILITY	UPDATE	DFHRCF		
START (see note 7)	XPCT	READ	transid		
STARTBR	XFCT	READ	file		
SUSPEND					
SYNCPOINT					
UNLOCK					
VERIFY PASSWORD					
WAIT					
WAIT JOURNALNAME	XJCT	READ	journal		
WAIT JOURNALNUM	XJCT	READ	journal		
WAITCICS					
WRITE	XFCT	UPDATE	file		
WRITE JOURNALNUM	XJCT	UPDATE	DFHJnn		
WRITE OPERATOR					
WRITEQ TD	XDCT	UPDATE	tdqueue		

Table 32 (Page 6 of 6). Resource and command check cross reference					
EXEC CICS COMMAND	**Resource Check**			**Check class=XCMD**	
	Class	**Access**	**Resource**	**Access**	**Resource**
WRITEQ TS (see note 4)	XTST	UPDATE	tsqueue		
XCTL	XPPT	READ	program		

Notes:

1. CANCEL does two checks. One is done against the transaction specified on the CANCEL command, and the other is done against the transaction associated with the reqid you are canceling (where applicable).
2. The CREATE CONNECTION command is subject to command security checking when you define a connection, for example; CREATE CONNECTION(con1) *Attribute*(...). However, when you use the CREATE CONNECTION COMPLETE or CREATE CONNECTION DISCARD command, no command security checking is performed unless you have been authorized to use COMPLETE and DISCARD. COMPLETE and DISCARD can only be used by those authorized to perform CREATE CONNECTION(con1) and CREATE SESSIONS(ses1) commands. Otherwise, ILLOGIC is returned.
3. An install surrogate user check can also occur.
4. A security check is performed only if a DFHTST TYPE=SECURITY macro has been coded in the TST with a name that matches the tsname.
5. No security check is performed, because the STARTBR command must be issued before this command and a security check is issued on the STARTBR command.
6. The QUERY SECURITY command is not controlled by resource or command checks, but it can cause them to be issued.
7. A start surrogate user check can also occur.
8. The resource check for the transid is only done if the reqid is associated with a transaction.
9. Security checked only for log/resumable reports.

Notices

This information was developed for products and services offered in the U.S.A. IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2-31 Roppongi 3-chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply in the United Kingdom or any other country where such provisions are inconsistent with local law:
INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

This publication could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact IBM United Kingdom Laboratories, MP151, Hursley Park, Winchester, Hampshire, England, SO21 2JN. Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

Trademarks and service marks

The following terms, used in this publication, are trademarks or service marks of IBM Corporation in the United States or other countries:

CICS, CICS/VSE, IBM, VSE/ESA, POWER, CICS Transaction server for VSE/ESA,

Other company, product, and service names may be trademarks or service marks of others.

Bibliography

CICS Transaction Server for VSE/ESA Release 1 library

Evaluation and planning	
Release Guide	GC33-1645
Migration Guide	GC33-1646
Report Controller Planning Guide	SC33-1941
General	
Master Index	SC33-1648
Trace Entries	SX33-6108
User's Handbook	SX33-6101
Glossary (softcopy only)	GC33-1649
Administration	
System Definition Guide	SC33-1651
Customization Guide	SC33-1652
Resource Definition Guide	SC33-1653
Operations and Utilities Guide	SC33-1654
CICS-Supplied Transactions	SC33-1655
Programming	
Application Programming Guide	SC33-1657
Application Programming Reference	SC33-1658
Sample Applications Guide	SC33-1713
Application Migration Aid Guide	SC33-1943
System Programming Reference	SC33-1659
Distributed Transaction Programming Guide	SC33-1661
Front End Programming Interface User's Guide	SC33-1662
Diagnosis	
Problem Determination Guide	GC33-1663
Messages and Codes Vol 3 (softcopy only)	SC33-6799
Diagnosis Reference	LY33-6085
Data Areas	LY33-6086
Supplementary Data Areas	LY33-6087
Communication	
Intercommunication Guide	SC33-1665
CICS Family: Interproduct Communication	SC33-0824
CICS Family: Communicating from CICS on System/390	SC33-1697
Special topics	
Recovery and Restart Guide	SC33-1666
Performance Guide	SC33-1667
Shared Data Tables Guide	SC33-1668
Security Guide	SC33-1942
External CICS Interface	SC33-1669
XRF Guide	SC33-1671
Report Controller User's Guide	SC33-1940
CICS Clients	
CICS Clients: Administration	SC33-1792
CICS Universal Clients Version 3 for OS/2: Administration	SC34-5450
CICS Universal Clients Version 3 for Windows: Administration	SC34-5449
CICS Universal Clients Version 3 for AIX: Administration	SC34-5348
CICS Universal Clients Version 3 for Solaris: Administration	SC34-5451
CICS Family: OO programming in C++ for CICS Clients	SC33-1923
CICS Family: OO programming in BASIC for CICS Clients	SC33-1671
CICS Family: Client/Server Programming	SC33-1435
CICS Transaction Gateway Version 3: Administration	SC34-5448

Books from VSE/ESA 2.4 base program libraries

VSE/ESA Version 2 Release 4

Book title	Order number
Administration	SC33-6705
Diagnosis Tools	SC33-6614
Extended Addressability	SC33-6621
Guide for Solving Problems	SC33-6710
Guide to System Functions	SC33-6711
Installation	SC33-6704
Licensed Program Specification	GC33-6700
Messages and Codes Volume 1	SC33-6796
Messages and Codes Volume 2	SC33-6798
Messages and Codes Volume 3	SC33-6799
Networking Support	SC33-6708
Operation	SC33-6706
Planning	SC33-6703
Programming and Workstation Guide	SC33-6709
System Control Statements	SC33-6713
System Macro Reference	SC33-6716
System Macro User's Guide	SC33-6715
System Upgrade and Service	SC33-6702
System Utilities	SC33-6717
TCP/IP User's Guide	SC33-6601
Turbo Dispatcher Guide and Reference	SC33-6797
Unattended Node Support	SC33-6712

High-Level Assembler Language (HLASM)

Book title	Order number
General Information	GC26-8261
Installation and Customization Guide	SC26-8263
Language Reference	SC26-8265
Programmer's Guide	SC26-8264

Language Environment for VSE/ESA (LE/VSE)

Book title	Order number
C Run-Time Library Reference	SC33-6689
C Run-Time Programming Guide	SC33-6688
Concepts Guide	GC33-6680
Debug Tool for VSE/ESA Fact Sheet	GC26-8925
Debug Tool for VSE/ESA Installation and Customization Guide	SC26-8798
Debug Tool for VSE/ESA User's Guide and Reference	SC26-8797
Debugging Guide and Run-Time Messages	SC33-6681
Diagnosis Guide	SC26-8060
Fact Sheet	GC33-6679
Installation and Customization Guide	SC33-6682
LE/VSE Enhancements	SC33-6778
Licensed Program Specification	GC33-6683
Programming Guide	SC33-6684
Programming Reference	SC33-6685
Run-Time Migration Guide	SC33-6687
Writing Interlanguage Communication Applications	SC33-6686

VSE/ICCF

Book title	Order number
Adminstration and Operations	SC33-6738
User's Guide	SC33-6739

VSE/POWER

Book title	Order number
Administration and Operation	SC33-6733
Application Programming	SC33-6736
Networking Guide	SC33-6735
Remote Job Entry User's Guide	SC33-6734

VSE/VSAM

Book title	Order number
Commands	SC33-6731
User's Guide and Application Programming	SC33-6732

VTAM for VSE/ESA

Book title	Order number
Customization	LY43-0063
Diagnosis	LY43-0065
Data Areas	LY43-0104
Messages and Codes	SC31-6493
Migration Guide	GC31-8072
Network Implementation Guide	SC31-6494
Operation	SC31-6495
Overview	GC31-8114
Programming	SC31-6496
Programming for LU6.2	SC31-6497
Release Guide	GC31-8090
Resource Definition Reference	SC31-6498

Books from VSE/ESA 2.4 optional program libraries

C for VSE/ESA (C/VSE)

Book title	Order number
C Run-Time Library Reference	SC33-6689
C Run-Time Programming Guide	SC33-6688
Diagnosis Guide	GC09-2426
Installation and Customization Guide	GC09-2422
Language Reference	SC09-2425
Licensed Program Specification	GC09-2421
Migration Guide	SC09-2423
User's Guide	SC09-2424

COBOL for VSE/ESA (COBOL/VSE)

Book title	Order number
Debug Tool for VSE/ESA Fact Sheet	GC26-8925
Debug Tool for VSE/ESA Installation and Customization Guide	SC26-8798
Debug Tool for VSE/ESA User's Guide and Reference	SC26-8797
Diagnosis Guide	SC26-8528
General Information	GC26-8068
Installation and Customization Guide	SC26-8071
Language Reference	SC26-8073
Licensed Program Specifications	GC26-8069
Migration Guide	GC26-8070
Migrating VSE Applications To Advanced COBOL	GC26-8349
Programming Guide	SC26-8072

DB2 Server for VSE

Book title	Order number
Application Programming	SC09-2393
Database Administration	GC09-2389
Installation	GC09-2391
Interactive SQL Guide and Reference	SC09-2410
Operation	SC09-2401
Overview	GC08-2386
System Administration	GC09-2406

DL/I VSE

Book title	Order number
Application and Database Design	SH24-5022
Application Programming: CALL and RQDLI Interface	SH12-5411
Application Programming: High-Level Programming Interface	SH24-5009
Database Administration	SH24-5011
Diagnostic Guide	SH24-5002
General Information	GH20-1246
Guide for New Users	SH24-5001
Interactive Resource Definition and Utilities	SH24-5029
Library Guide and Master Index	GH24-5008
Licensed Program Specifications	GH24-5031
Low-level Code and Continuity Check Feature	SH20-9046
Library Guide and Master Index	GH24-5008
Messages and Codes	SH12-5414
Recovery and Restart Guide	SH24-5030
Reference Summary: CALL Program Interface	SX24-5103
Reference Summary: System Programming	SX24-5104
Reference Summary: HLPI Interface	SX24-5120
Release Guide	SC33-6211

PL/I for VSE/ESA (PL/I VSE)

Book title	Order number
Compile Time Messages and Codes	SC26-8059
Debug Tool For VSE/ESA User's Guide and Reference	SC26-8797
Diagnosis Guide	SC26-8058
Installation and Customization Guide	SC26-8057
Language Reference	SC26-8054
Licensed Program Specifications	GC26-8055
Migration Guide	SC26-8056
Programming Guide	SC26-8053
Reference Summary	SX26-3836

Screen Definition Facility II (SDF II)

Book title	Order number
VSE Administrator's Guide	SH12-6311
VSE General Introduction	SH12-6315
VSE Primer for CICS/BMS Programs	SH12-6313
VSE Run-Time Services	SH12-6312

Index

A

access authorization levels 51
access to CICS from ports of entry 35
ACICSPCT general resource class 52
ATTACHSEC operand 102, 138, 146
  IDENTIFY parameter 102
  LOCAL parameter 102
  MIXIDPE parameter 103
  PERSISTENT parameter 103
  USEDFLTUSER option 106
  VERIFY parameter 102
auditing
  bind security failure 97
authorization levels for journal access 51
authorizing CICS region userid as surrogate user 24
autoinstall models 37

B

batch access to CSD, restricting 37
batch call interface 153
BCICSPCT general resource class 52
bind-time security 95, 137, 143
BINDSECURITY option 97, 98
BMS commands 46
BUILD ATTACH command 147

C

categories of CICS-supplied transactions 81
CCICSCMD general resource class 68
CEBT transaction 43
CEDA LOCK command 37
CEDA transaction 36
CEDF transaction 56, 69
CEMT SET PROG(xxx) NEWCOPY command 70
CESN CICS-supplied sign-on transaction 31
CICS JOB statement, PASSWORD parameter 19
CICS JOB statement, USER parameter 19
CICS region userid 19, 81
CICS security does not protect 176
CICS security protection 175
CICS segment 10
CICS system definition file (CSD), restricting batch access to 37
CICS user restart program, PLTPI 43
CICS-supplied transactions, categories 81
CICS-value data area (CVDA) 73
CICS/VSE and the ESM 175
CMDSEC, command security parameter 68
command security 6
controlling access to CICS 35
controlling RCF printers 162
CRTE, routing transaction 109, 151
CSCS transient data destination 34
CSD (CICS system definition file), restricting batch access to 37
CSD definitions, locking 36
CVDA (CICS-value data area) 73

D

data tables
  bind security 158
  CONNECT security checks 158
  file security 159
  LOGON security check 158
  security checking 157
date subfields, format 130
DCICSDCT general resource class 48
DFHEXCI surrogate profile 62
DFHINSTL surrogate profile 62
DFHSNxxxx messages 34
DFHSTART surrogate profile 62
DFHXCIS 153
DFHXCOPT, EXCI options table 61
DFLTUSER, system initialization parameter 26
discrete profiles 15
DPL
  See ?

E

EBCDIC, for PEM userids and passwords 125
ECICSDCT general resource class 48
ESM requirements for Report Controller security 161
ESMEXITS, system initialization parameter 26
EXCI security 153
external call interface 153
External CICS interface (EXCI) and surrogate checking 61
external security manager (ESM) 175

F

FACILITY general resource class 14
FCICSFCT general resource class 51
FEPI security 8
FEPIRESOURCE resource name 65
flows, examples 121
Front End Programming Interface security 8

G
GCICSTRN general resource class 41, 53
generating and using PassTickets 8
group identifier 31

H
HCICSFCT general resource class 51

I
ICHRTX00, router exit 169
id.Report Controller security 8
IDENTIFY parameter, ATTACHSEC operand 102
identifying remote users 103, 147
internal bind time security 98

J
JCICSJCT general resource class 51

K
KCICSJCT general resource class 51

L
languages,primary and secondary 13
link security 99, 137
LOCAL parameter, ATTACHSEC operand 102
LOCK command, CEDA 37
LU6.1 links 137
LU6.1 security 137

M
MCICSPPT general resource class 54
MIXIDPE parameter, ATTACHSEC operand 103

N
National Language Support 38
NATLANG and non-terminal transactions 39
NCICSPPT general resource class 54

O
OIDCARD (operator identification card) 4
OPCLASS 10
OPIDENT 11
OPPRTY 11

P
PassTickets 8
PERSISTENT parameter, ATTACHSEC operand 103
PIP (program initialization parameter) data 125
PLT programs 43
PLTPI 43
PLTSD 43
POWER JECL statement, SEC parameter 20
preset terminal NATLANG 39
preset terminal security 5, 35
primary and secondary languages 13
problem determination 185
program initialization parameter (PIP) data 125
PVDELAY system initialization parameter 104

Q
QUERY SECURITY command 7

R
RACROUTE macros 167
remote operators 100, 146
remote user sign-off 103, 147
remote users 100, 146
report controller relationship to ESM 162
Report Controller security 161
report security checking 162
RESID values for SPCOMMAND 75
resource definition online (RDO) 85
resource security 6, 108, 139, 150
RESSEC operand of DEFINE TRANSACTION 108, 139, 150
RESSEC, resource security parameter 47
restructured CICS
  signon subcomponent 148
retention of RSL 161
routing transaction, CRTE 109, 151

S
SCICSTST general resource class 55
scoping sign-on definition 32
SEC parameter on POWER JECL statement 20
SEC, system initialization parameter 24
SECPRFX, system initialization parameter 25
securing transactions and resources 91
security 43
  non-terminal 43
Security Migration Aid (SMA) 175
security rebuild 97
session key 95
session security 95
session segment 96
shared data tables
  bind security 158
  CONNECT security checks 158
  file security 159
  LOGON security check 158

shared data tables *(continued)*
security checking 157
signon 148
SMA 176
SMF (System Management Facility) 34
SNA service transaction program name for sign-on transaction program 127
SNSCOPE sign-on operand 26
SPCOMMAND, RESID values 75
specifying RSL value 162
stage two utility 176
started transaction security 52
SURROGAT general resource class 15, 36, 62
SURROGAT transaction 36
surrogate authority, querying a user's 77
surrogate terminal 109, 151
surrogate user
authorizing CICS region userid as 24
surrogate user security 6
SURROGCHK parameter 61
System Management Facility (SMF) 34
systems network architecture (SNA) session security 90

T

TCICSTRN general resource class 41, 53
temporary storage 46, 55
terminal user security 4
time subfields, format 130
TIMEOUT 11
transaction initiation 107, 138, 149
transaction routing and QUERY SECURITY 74
transaction security 6, 91
triggered transactions 50

U

UCICSTST general resource class 55
USEDFLTUSER option 106
USER parameter on CICS JOB statement 19
user profile 130
user security 100, 146
user-defined classes 170
userid passed as parameter on EXCI calls 61
USRDELAY, system initialization parameter 147

V

VERIFY parameter, ATTACHSEC operand 102
verifying remote users 104
VSAM KSDS 175, 176

X

XAPPC, system initialization parameter 27, 28, 46, 96
XCMD, system initialization parameter 27, 46, 68
XDCT, system initialization parameter 27, 46, 49
XFCT, system initialization parameter 27, 46, 51
XJCT, system initialization parameter 27, 46, 52
Xname, system initialization parameters 181, 182
XPCT-checked transaction security 52
XPCT, system initialization parameter 27, 46, 52
XPPT, system initialization parameter 27, 46, 55
XSNOFF global user exit 171
XSNON global user exit 171
XTRAN, system initialization parameter 27
XTST, system initialization parameter 27, 46, 55
XUSER, system initialization parameter 27, 29, 36, 46

Sending your comments to IBM

CICS® Transaction Server for VSE/ESA™

Security Guide

SC33-1942-00

If you want to send to IBM any comments you have about this book, please use one of the methods listed below. Feel free to comment on anything you regard as a specific error or omission in the subject matter, and on the clarity, organization or completeness of the book itself.

To request additional publications, or to ask questions or make comments about the functions of IBM products or systems, you should talk to your IBM representative or to your IBM authorized remarketer.

When you send comments to IBM, you grant IBM a nonexclusive right to use or distribute your comments in any way it believes appropriate, without incurring any obligation to you.

You can send your comments to IBM in any of the following ways:

- By mail:

  IBM UK Laboratories
  Information Development
  Mail Point 095
  Hursley Park
  Winchester, SO21 2JN
  England

- By fax:
  - From outside the U.K., after your international access code use 44 1962 870229
  - From within the U.K., use 01962 870229
- Electronically, use the appropriate network ID:
  - IBM Mail Exchange: GBIBM2Q9 at IBMMAIL
  - IBMLink: HURSLEY(IDRCF)
  - Email: idrcf@hursley.ibm.com

Whichever method you use, ensure that you include:

- The publication number and title
- The page number or topic to which your comment applies
- Your name and address/telephone number/fax number/network ID.

Program Number: 5648-054

Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

SC33-1942-00

Spine information:

CICS TS for VSE/ESA Security Guide *Release 1*