

CICS® Transaction Server for VSE/ESA™



# Release Guide

*Release 1*



CICS® Transaction Server for VSE/ESA™



# Release Guide

*Release 1*

**Note!**

Before using this information and the product it supports, be sure to read the general information under "Notices" on page 177.

**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.

The CICS for VSE/ESA Version 2.3 edition remains applicable and current for users of CICS for VSE/ESA Version 2.3.

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.

© Copyright International Business Machines Corporation 1999. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

---

# Contents

<b>Preface</b> . . . . .	vii
Notes on terminology . . . . .	viii
Determining if a publication is current . . . . .	x
<b>Chapter 1. Summary of new and changed function</b> . . . . .	1
<b>Chapter 2. Hardware and software requirements</b> . . . . .	5
VSE/ESA Version 2 Release 4 requirements . . . . .	5
Hardware prerequisites . . . . .	5
Software requirements . . . . .	8
<hr/>	
<b>Part 1. Improved availability</b> . . . . .	11
<b>Chapter 3. Restructure of CICS internal code</b> . . . . .	13
Virtual storage constraint relief . . . . .	17
<b>Chapter 4. Storage enhancements</b> . . . . .	21
Dynamic storage areas . . . . .	21
CICS storage keys . . . . .	23
Using storage protection . . . . .	24
Changes to CICS externals . . . . .	26
<b>Chapter 5. VTAM single-node persistent sessions</b> . . . . .	29
Using VTAM persistent sessions . . . . .	32
Changes to CICS externals . . . . .	32
<b>Chapter 6. Intersystem session queue management</b> . . . . .	35
Connection parameters . . . . .	35
The XZIQUE global user exit . . . . .	35
Using intersystem session queue management . . . . .	36
Changes to CICS externals . . . . .	36
<hr/>	
<b>Part 2. Security</b> . . . . .	37
<b>Chapter 7. CICS security</b> . . . . .	39
The security domains . . . . .	39
Changes to resource definition . . . . .	40
Vendor and in-house products . . . . .	40
Changes to transaction security . . . . .	40
Terminal user security . . . . .	43
Preset terminal security . . . . .	44
Changes to sign-on procedures . . . . .	44
The CICS default user . . . . .	45
APPC (LU6.2) session security . . . . .	46
Multiregion (MRO) security (not supported by the BSM) . . . . .	47
Resource security checking (not supported by the BSM) . . . . .	48
Command security checking (not supported by the BSM) . . . . .	53
Changes affecting console operators (not supported by the BSM) . . . . .	56
Changes to system initialization parameters . . . . .	56

<b>Chapter 8. The report controller</b>	57
Enhancements to the report controller	57
Report security	57
Printer security	58
Browse security	58
<b>Chapter 9. Enhancements to XRF</b>	59
XRF sign-on retention	59
XRFTRSZ SIT parameter removed	59
Change to the APPLID system initialization parameter	60
Overseer program compatibility	60
<b>Chapter 10. APPC (LU6.2) persistent verification</b>	61
Using persistent verification	61
Changes to CICS externals	62

---

### Part 3. Improvements to application programming . . . . . 63

<b>Chapter 11. Changes for application programming</b>	65
Obsoleted function	65
CICS internal macros	67
<b>Chapter 12. CPI-C</b>	69
Using CPI-C	69
Changes to CICS externals	70
<b>Chapter 13. FEPI</b>	71
Overview of using FEPI	71
Changes to CICS externals	73
<b>Chapter 14. EXCI</b>	77
Using EXCI	78
Changes to CICS externals	79
<b>Chapter 15. Cancel start requests</b>	83
Using the cancel function	83
Changes to CICS externals	83
<b>Chapter 16. Shared data tables</b>	85
Using shared data tables	86
Changes to CICS externals	87

---

### Part 4. Improvements to system programming . . . . . 89

<b>Chapter 17. Changes to global user exits in the AP domain</b>	91
Task-related user exits	97
Changes to user-replaceable programs	98
<b>Chapter 18. XPI</b>	101
The exit programming interface (XPI)	101
<b>Chapter 19. Access to CICS state data</b>	105

Using CICS state data . . . . .	105
Changes to CICS externals . . . . .	105

---

**Part 5. Improved system management . . . . . 109**

<b>Chapter 20. Improved dynamic transaction routing . . . . .</b>	<b>111</b>
Using DTR enhancements . . . . .	111
Changes to CICS externals . . . . .	112

<b>Chapter 21. Autoinstall for programs, mapsets, and partitionsets . . . . .</b>	<b>113</b>
Using autoinstall for programs, mapsets, and partitionsets . . . . .	115
Changes to CICS externals . . . . .	115

<b>Chapter 22. Automatic journal archiving . . . . .</b>	<b>117</b>
Using automatic journal archiving . . . . .	119
Changes to CICS externals . . . . .	119

<b>Chapter 23. Improved monitoring facilities . . . . .</b>	<b>121</b>
Obsolete function . . . . .	121
Differences in data collection . . . . .	121
Trace . . . . .	124
Dump . . . . .	124
Using new CICS monitoring functions . . . . .	125
Changes to CICS externals . . . . .	125

<b>Chapter 24. Improved collection of statistics . . . . .</b>	<b>129</b>
Obsolete function . . . . .	129
Differences in data collection . . . . .	130
Changes to controlling statistics . . . . .	131
Using the new CICS statistics function . . . . .	132
Changes to CICS externals . . . . .	133

<b>Chapter 25. Improved RDO . . . . .</b>	<b>135</b>
RDO for files . . . . .	135
Specifying read-only access to the CSD . . . . .	136
Associating descriptive comments with resource definitions in the CSD . . . . .	136
Invoking DFHCSDUP from user programs . . . . .	136
Using the DFHCSDUP ALTER command . . . . .	137
Sharing the CSD . . . . .	137
Dual-purpose resource definition . . . . .	137
Removing installed resource definitions . . . . .	138
Using the programmable interface to CEDA . . . . .	139
Installing single resources . . . . .	139
Immediate commitment of resources . . . . .	139

<b>Chapter 26. Resolving in-doubt resources . . . . .</b>	<b>141</b>
DFH\$IWUP utility program . . . . .	141
Using the DFH\$IWUP utility program . . . . .	142

<b>Chapter 27. Miscellaneous changes to CICS . . . . .</b>	<b>143</b>
Recovery of VSAM files . . . . .	143
Recovery from DTB failures . . . . .	143
Support for forward recovery utilities . . . . .	144

Improvements to MRO . . . . .	144
Enhancements to transaction management . . . . .	145
<b>Chapter 28. Changes for system and resource definition . . . . .</b>	<b>147</b>
Changes to system definition . . . . .	147
Changes to resource definition . . . . .	148
Using the changed system and resource definitions . . . . .	150
<b>Chapter 29. Creating and installing resources using EXEC CICS</b>	
<b>CREATE . . . . .</b>	<b>153</b>
Overview of the EXEC CICS CREATE command . . . . .	153
Benefits of the EXEC CICS CREATE command . . . . .	154
Changes to CICS externals . . . . .	154
Security . . . . .	156
Coexistence . . . . .	156
Examples of the CREATE command . . . . .	156
<b>Appendix A. System initialization parameters . . . . .</b>	<b>159</b>
New system initialization parameters . . . . .	159
Changed system initialization parameters . . . . .	167
Obsolete system initialization parameters . . . . .	169
<b>Bibliography . . . . .</b>	<b>171</b>
Books from VSE/ESA 2.4 base program libraries . . . . .	172
Books from VSE/ESA 2.4 optional program libraries . . . . .	174
<b>Notices . . . . .</b>	<b>177</b>
Trademarks and service marks . . . . .	178
<b>Index . . . . .</b>	<b>179</b>



---

# Preface

CICS® TS Release 1 is the latest release of CICS for VSE. This release contains many new functions. As well as protecting and extending your current investment in CICS applications, CICS TS restructures the CICS code base. This enables CICS to meet ever increasing demands on workload and positions it for further extensions in the future. CICS TS standardizes customization practices to smooth migration in future releases.

CICS TS provides functions to improve application programming, system programming, system management and the reliability and integrity of your CICS system today, but also consolidates CICS for VSE into a strong position for the future.

## What this book is about

This book describes the new and changed features of this version of CICS, and the benefits that they provide.

## Who this book is for

You should read this book if you are responsible for:

- Evaluation and planning
- System administration
- Programming
- Customization

## What you need to know to understand this book

The book assumes that you are already familiar with CICS, either as a systems administrator, or as a systems or application programmer.

## How to use this book

Each chapter in this book:

- Describes a major new function
- Outlines what you have to do to use the new function
- Summarizes the changes to CICS externals—programming interfaces, resource definitions and so on
- Refers you to other books in the library for further, detailed information

Tables, called *roadmaps*, are used to help you find the information you are looking for in a part or chapter of this book.

## 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.

<b>Term</b>	<b>Definition (and abbreviation if appropriate)</b>
\$(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: <ul style="list-style-type: none"><li>• Signon security</li><li>• Transaction attach security</li></ul>
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	<p>ESM is used to indicate a RACROUTE-compliant external security manager that supports some or all of the following functions:</p> <ul style="list-style-type: none"> <li>• Signon security</li> <li>• Transaction attach security</li> <li>• Resource security</li> <li>• Command security</li> <li>• Non-terminal security</li> <li>• Surrogate user security</li> <li>• MRO/ISC security (MRO, LU6.1 or LU6.2)</li> <li>• FEPI security.</li> </ul>
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”

## Determining if a publication is current

IBM regularly updates its publications with new and changed information. When first published, both the printed hardcopy and the BookManager softcopy versions of a publication are in step, but subsequent updates are normally made available in softcopy before they appear in hardcopy.

For CICS Transaction Server for VSE/ESA Release 1 books, softcopy updates appear regularly on the *Transaction Processing and Data Collection Kit* CD-ROM, SK2T-0730-xx and on the *VSE/ESA Collection Kit* CD-ROM, SK2T-0060-xx. Each reissue of the collection kit is indicated by an updated order number suffix (the -xx part). For example, collection kit SK2T-0730-20 is more up-to-date than SK2T-0730-19. The collection kit is also clearly dated on the front cover.

For individual books, the suffix number is incremented each time it is updated, so a publication with order number SC33-0667-02 is more recent than one with order number SC33-0667-01. Updates in the softcopy are clearly marked by revision codes (usually a “#” character) to the left of the changes.

Note that book suffix numbers are updated as a product moves from release to release, as well as for updates within a given release. Also, the date in the edition notice is not changed until the hardcopy is reissued.

# Chapter 1. Summary of new and changed function

This chapter summarizes what is new and changed in CICS®.

Function	What it does	Where to find more information
<b>Availability improvements</b>		
CICS internal restructure	Improves reliability and availability by restructuring CICS code and control blocks into domains.	Chapter 3, "Restructure of CICS internal code" on page 13
CICS storage protection	Prevents CICS application programs accidentally overwriting CICS code and control blocks.	Chapter 4, "Storage enhancements" on page 21
VTAM® single-node persistent session support	Improves the availability of CICS by using the VTAM 4.2 persistent LU-LU session enhancements to enable a fast restart in place in the event of a CICS failure. CICS uses VTAM-retained state data to rebind the CICS sessions.	Chapter 5, "VTAM single-node persistent sessions" on page 29
Intersystem session queue management	Detects that a partner CICS region is under stress by checking the queued ALLOCATE requests. The information passed enables an exit program to determine whether to queue or reject the request. This reduces queueing and maintains performance in interconnected regions.	Chapter 6, "Intersystem session queue management" on page 35
<b>Improved security</b>		
External security management	<p>CICS internal security is no longer provided. Security services are provided by an external security manager (ESM) that conforms to the system authorization facility (SAF) interface. Sign-on and transaction security are supported by a limited-function ESM—the Basic Security Manager (BSM), which is provided in VSE/ESA Version 2 Release 4.</p> <p>To provide the necessary security for CICS regions, CICS issues RACROUTE calls through the new VSE/ESA Version 2 Release 4 SAF interface to route authorization requests to an ESM at appropriate points within CICS transaction processing.</p>	Chapter 7, "CICS security" on page 39
Non-terminal resource security checking	Supports CICS resource security checking for transactions started without a terminal.	"Changes to transaction security" on page 40
Command security checking	Enables you to limit, to properly authorized users, the use of system programming interface (SPI) -type commands issued from CICS application programs, and the equivalent commands that can be issued from the CICS master terminal transaction (CEMT). Command security operates in addition to any transaction or resource security that may already have been defined for a transaction.	"Command security checking (not supported by the BSM)" on page 53
APPC (LU6.2) persistent verification	Enables you to sign-on to a remote host and have that signon remain valid over multiple conversations, until it is no longer needed.	Chapter 10, "APPC (LU6.2) persistent verification" on page 61

Function	What it does	Where to find more information
<b>Application programming improvements</b>		
Common programming interface communications (CPI-C)	<p>Provides a call-based application programming interface for APPC distributed transaction processing on LU6.2 sessions that is common across programming languages, and across hardware systems. For example, a COBOL program on an IBM® AS/400® system and a PL/I program on a CICS system are able to use the same set of commands to communicate.</p> <p>This interface is in addition to existing support for distributed transaction processing provided through CICS APPC commands.</p> <p><b>Note:</b> The existing EXEC CICS GDS commands are <i>not</i> removed.</p>	Chapter 12, "CPI-C" on page 69
Front End Programming Interface (FEPI)	<p>Enables you to write CICS application programs that access other CICS programs. In other words, it provides a front end to those programs: the interface simulates the terminals that the other programs use.</p> <p>FEPI allows existing CICS application programs to be used in different ways, in different combinations, in different environments, and on different systems, without having to change them, because it provides a simple integrated interface to these programs.</p> <p>FEPI also allows you to write new programs that add function to old programs.</p>	Chapter 13, "FEPI" on page 71
External CICS interface (EXCI)	<p>Allows you to write client programs that can call CICS programs running in a CICS system (server programs). Using this interface, you can access and control CICS resources with full integrity. A client program that uses the EXCI can operate multiple sessions for different users all coexisting in the same VSE partition without knowledge of (or interference from) each other.</p>	Chapter 14, "EXCI" on page 77
Cancel start requests	<p>Enables an application program to purge all the pending transaction start requests associated with a defined terminal or connection. It is provided by a CANCEL option on the EXEC CICS SET TERMINAL and EXEC CICS SET CONNECTION commands.</p>	Chapter 15, "Cancel start requests" on page 83
Shared data tables	<p>Enhances the data tables facility and has important advantages over the support provided in previous releases, in terms of performance and function.</p>	Chapter 16, "Shared data tables" on page 85
<b>Improvements to system programming</b>		
New global user exits in the application programming (AP) domain	<p>Examples include modification of transient data control, interval control, and temporary storage control commands.</p>	Chapter 17, "Changes to global user exits in the AP domain" on page 91
Exit programming interface (XPI)	<p>Provides a new exit programming interface (XPI) for you to use if you need to extend the functions of your CICS systems in ways not provided as part of a standard system. The XPI gives you access to a set of CICS services for global user exit programs written in assembler language.</p>	Chapter 18, "XPI" on page 101

<b>Function</b>	<b>What it does</b>	<b>Where to find more information</b>
Access to CICS state data	Provides command-level and exit programming interface (XPI) access to additional CICS state data. It also provides some new exit points, and a programmable interface for purging unwanted started tasks; that is, to purge interval control elements (ICEs).	Chapter 19, "Access to CICS state data" on page 105
<b>System management improvements</b>		
Dynamic transaction routing enhancements	Allow a dynamic transaction routing program to keep track of CICS-initiated transactions, and transactions that abend in an application-owning region (AOR).	Chapter 20, "Improved dynamic transaction routing" on page 111
Autoinstall for programs, mapsets, and partitionsets	Extends the CICS autoinstall facility to enable CICS to install programs, mapsets, and partitionsets without prior definition of each resource. This autoinstall facility is similar to that already supported for terminals.	Chapter 21, "Autoinstall for programs, mapsets, and partitionsets" on page 113
Automatic journal archiving	Allows you to automate the process of backing-up CICS journals when a journal switch has occurred.	Chapter 22, "Automatic journal archiving" on page 117
Monitoring and statistics enhancements	Significantly extends the range and scope of monitoring data. This data is written to data sets of a new CICS facility, the Data Management Facility (DMF) Data Handler.	Chapter 23, "Improved monitoring facilities" on page 121
RDO for files	Supports the online definition of files. The following resources associated with CICS files can be managed using RDO: <ul style="list-style-type: none"> <li>• VSAM files</li> <li>• Remote VSAM files</li> <li>• Remote DAM files</li> <li>• VSAM local shared resource (LSR) pools</li> <li>• Shared data tables</li> </ul>	Chapter 25, "Improved RDO" on page 135
DFH\$IWUP utility program	Helps you to resolve in-doubt resources.	Chapter 26, "Resolving in-doubt resources" on page 141
<b>General enhancements</b>		
Dynamic DSA management	Increases the number of CICS-managed DSAs from one to eight: four above and four below 16MB.	Chapter 4, "Storage enhancements" on page 21
Recovery of VSAM files	Simplifies CICS restart procedures and allows you to: <ul style="list-style-type: none"> <li>• Automatically archive CICS journal data sets</li> <li>• Ensure data set integrity in the event of a dynamic transaction backout (DTB) failure, without having to abend CICS.</li> </ul>	"Recovery of VSAM files" on page 143 for details.
Recovery from DTB failures	Manages backout failures that are detected during DTB (or during emergency restart).	"Recovery from DTB failures" on page 143
Support for forward recovery utilities	Introduces new options to control the logging of REDO records, and to control the journal to which the REDO records are written. These options are available only with resource definition online (RDO).	"Support for forward recovery utilities" on page 144

Function	What it does	Where to find more information
MRO and transaction management	<p>Enhances MRO support:</p> <ul style="list-style-type: none"> <li>• Improved queueing performance</li> <li>• Duplicate logon names for EXCI support</li> <li>• ESM support for MRO link security</li> </ul> <p>Transaction management improvements include:</p> <ul style="list-style-type: none"> <li>• Unlimited number of transaction classes</li> <li>• Simplified system definition</li> <li>• New transaction manager with greater flexibility</li> <li>• New definition option, STORAGECLEAR</li> </ul>	<p>“Improvements to MRO” on page 144 and “Enhancements to transaction management” on page 145</p>
Application programming	<p>Obsoletes some functions:</p> <ul style="list-style-type: none"> <li>• DFHEICAL macro</li> <li>• Access to CICS control blocks such as the CSA and TCA</li> <li>• Support for RPG, DOS PL/I and C/370™ compilers</li> <li>• Macro-level application programs</li> </ul> <p>There are several other minor changes.</p>	<p>Chapter 11, “Changes for application programming” on page 65</p>
System and resource definition	<p>Obsoletes several system definition functions:</p> <ul style="list-style-type: none"> <li>• System initialization overlays (SIMODS)</li> <li>• SYSGEN (DFHSG) macros</li> <li>• Documentation about programmable interface to CEMT</li> </ul> <p>Some resource definition functions are also withdrawn, including:</p> <ul style="list-style-type: none"> <li>• Application load table (ALT)</li> <li>• Nucleus load table (NLT)</li> <li>• Sign-on table (SNT)</li> <li>• System recovery table (SRT) program and routine options</li> </ul>	<p>Chapter 28, “Changes for system and resource definition” on page 147</p>
Changed and new utility programs	<p>Provides a number of additions and enhancements, including:</p> <ul style="list-style-type: none"> <li>• New automatic journaling utility, DFHJACDU</li> <li>• New monitoring utility, DFHMNDUP</li> <li>• Enhancements to statistics utility, DFHSTUP</li> </ul>	<p>Chapter 22, “Automatic journal archiving” on page 117 and Chapter 23, “Improved monitoring facilities” on page 121</p>
System initialization parameters	<p>Introduces new, changed, and obsolete system initialization parameters.</p>	<p>Appendix A, “System initialization parameters” on page 159.</p>



---

## Chapter 2. Hardware and software requirements

This chapter describes hardware and software requirements.

---

### VSE/ESA Version 2 Release 4 requirements

CICS Transaction Server for VSE/ESA Release 1 is available only as part of VSE/ESA™ Version 2 Release 4. The software required by CICS is supplied with VSE/ESA V2.4.

### Other VSE/ESA components or subsystems

Table 2 gives release information for some of the main VSE/ESA components and subsystems.

<i>Table 2. Release information for VSE/ESA components</i>		
<b>Product</b>	<b>Release number</b>	<b>Product number</b>
VSE/ESA Central Functions component	Version 6 Release 4	5686-066
VSE/VSAM <b>1</b>		5686-037
VSE/POWER <b>1</b>		5686-033
VSE/ICCF <b>1</b>		5686-036
VTAM <b>2</b> <b>3</b>	Version 4 Release 2	5686-065
DITTO/ESA for VSE <b>2</b>	Version 1 Release 3	5648-099
LE/VSE <b>2</b>	Version 1 Release 4	5686-094
<b>Notes:</b>		
<b>1</b> Included in VSE/ESA Version 2 Release 4 as part of the Central Functions component.		
<b>2</b> Included in VSE/ESA Version 2 Release 4 as a base program.		
<b>3</b> For persistent session support, the inter-enterprise option is also required—this is not supplied as part of the base program.		

For detailed information about the base and optional programs provided, see the *VSE/ESA Version 2 Release 4 Planning* manual.

---

### Hardware prerequisites

Use the following sections to verify that you have the required hardware:

- Processors, including hardware for storage protection
- Device support, covering terminal devices, DASD, tapes, and printers.

## Processors

CICS runs on processors capable of supporting VSE/ESA Version 2 Release 4 (or upward compatible release) in **ESA mode only**. These processors include Enterprise Systems Architecture/390® (ESA/390™) processors that meet the requirements of the host operating system, CICS, access methods, and user application programs. 370 and XA processors are not supported.

### Hardware requirements for storage protection support

To use storage protection facilities, you need an ESA/390 processor that supports the subsystem storage protection facility, and which meets the requirements of the host operating system, CICS, the access methods, and your own application programs. Any of the following IBM ES/9000® machines are suitable:

- ES/9000 9021 Model 520, 640, 660, 711, 740, 820, 860, 831, 832, 860, 900, 941 or higher
- ES/9000 9121 Model 180, 190, 210, 260, 311, 320, 411, 440, 480 490, 511, 521, 522, 570, 610, 621, 622, 732 or 742
- ES/9000 9221 Model 170 or 200.

Other models may require an engineering change to be able to provide the function.

## Device support

The following sections outline the changes to support for various devices.

### Devices connected using VTAM

All previously supported VTAM terminals are able to communicate with CICS TS Release 1. New or current terminals are directly supported by CICS if they conform to the VTAM interface.

The list of device and system types supported, with modes of connection, is shown in Table 3 on page 7.

### Devices connected using BTAM

CICS no longer supports devices and controllers accessed using BTAM unless they are defined as remote. If they are defined as remote devices, they can communicate with CICS using interregion communication (IRC) from a CICS for VSE/ESA Version 2.3 region to which they are locally defined. The list of device and system types supported, with modes of connection, is shown in Table 3 on page 7.

### DASD

Support for data sets on any new DASD device is transparent to CICS because it uses standard access methods.

### Tapes

Support for data sets on any new tape device is transparent to CICS because it uses standard access methods.

CICS continues to support current tape devices, including early 3480 tape devices, although these are not recommended for tape logging, and CICS does not support the use of 3480 follow-on tape devices for tape logging.

There are no restrictions on the use of tape devices for extrapartition data.

## Printers

Support for any new VTAM-attached printers is transparent to CICS and should not affect CICS programs.

CICS does not support native Advanced Function Printer (AFP™) devices. The EXEC CICS SPOOL API was enhanced in CICS/VSE® Version 2 so that you can specify attributes that are meaningful to the appropriate device driver products (for example, Print Service Facility (PSF)).

## Sequential terminal support

CICS continues to support the use of sequential terminals and printers.

## Supported device and system types

The following are supported:

<i>Table 3 (Page 1 of 2). Supported system and device types</i>	
<b>Direct connection to CICS using VTAM</b>	<b>Connection using transaction routing through CICS for VSE/ESA Version 2.3 (BTAM terminals)</b>
3101 Display Terminal (see note)	1050 Data Communication System
3230 Printer (see note)	2740 Communication Terminal
3268 Printer	2741 Terminal controller
3270 Information Display System (see note)	2770 Data Communication System
3270 PC (see note)	2780 Data Transmission Terminal
3270 PC/G (see note)	2980 General Banking Terminal System
3270 PC/GX (see note)	3660 Supermarket System
3287 Printer	3735 Programmable Buffered Terminal
3600 Finance Communication System (see note)	3740 Data Entry System
3630 Plant Communication System (see note)	<ul style="list-style-type: none"> <li>• 3780 Data Communications Terminal</li> <li>• 5100 Portable Computer</li> </ul>
3650 Retail Store System	5110 Portable Computer
3680 Programmable Store System	5230 Data Collection System
3730 Distributed Office Communication System	5260 Retail System
3767 Communication Terminal (see note)	Communicating Magnetic Card
3770 Data Communication System (see note)	Selectric Typewriter (CMCST)
3790 Communication System	Office System/6
4300 Processors (see note)	Series/1
4700 Finance Communication System (see note)	System/3
5280 Distributed Data System (see note)	System/7
5520 Administrative System (see note)	System/23
5550 Administrative System (see note)	---
5937 Rugged Terminal (see note)	---
6670 Information Distributor (see note)	---
8100 Information System (see note)	---

<i>Table 3 (Page 2 of 2). Supported system and device types</i>	
<b>Direct connection to CICS using VTAM</b>	<b>Connection using transaction routing through CICS for VSE/ESA Version 2.3 (BTAM terminals)</b>
8775 Display Terminal	---
8815 Scanmaster	---
Displaywriter (see note)	---
Personal Computer, PS/2, PS/55 (see note)	---
System/32 (see note)	---
System/34 (see note)	---
System/36™ (see note)	---
System/38™ (see note)	---
AS/400	---
System/370™ (including 303x, 308x, and 3090™ processors) (see note)	---
Teletypewriter Exchange Service (TWX 33/35) (see note)	---
World Trade Typewriter Terminal (WTTY) (see note)	---
<b>Note:</b> These terminals are also supported by BTAM on CICS for VSE/ESA Version 2.3.	

## Software requirements

Use the following sections to verify that you have the appropriate software to run CICS:

- Access methods
- Database products
- Programming languages
- Intersystem communication (ISC) and multiregion operation (MRO)

## Access methods

CICS requires the access methods shown in Table 4.

<i>Table 4. Release information for access methods supported in CICS TS Release 1</i>		
<b>Product</b>	<b>Release number</b>	<b>Program number</b>
VSE/VSAM	Mandatory. Supplied in VSE/ESA V2.4 as part of the Central Functions component.	5686-037
VTAM	Version 4 Release 2. Supplied as a VSE/ESA V2.4 base program.	5686-065

CICS does not support BTAM networks. Consider adopting a coexistence strategy if you cannot migrate your BTAM networks to VTAM. BTAM/ES Version 1 Release 1 is available as a VSE/ESA Version 2 Release 4 optional program. See the *CICS Migration Guide* for more information about coexistence strategies.

## Database products

CICS runs with the optional IBM database products listed in Table 5.

Product	Release number	Program number
DL/I VSE <b>1</b>	1.11	5746-XX1
DB2 for VSE/ESA <b>2</b>	6.0	5648-158

**Notes:**

**1** DL/I VSE is the new name for DL/I DOS VS. DL/I DOS VS 1.10 is still available on the optional product tape, but works only with CICS for VSE/ESA Version 2.3 . DL/I VSE 1.11 does not work with CICS for VSE/ESA Version 2.3.

**2** DB2 for VSE/ESA was previously known as SQL/DS™.

## Programming languages

CICS supports only those application programs that use the CICS command-level application programming interface (API), written in the following languages:

- Assembler
- C
- COBOL
- PL/I

Macro-level application programs must be converted to command-level using the tools and services provided, such as DFHMSCAN and the CICS/AMA utility. Consider a coexistence strategy if full conversion is not possible.

CICS does not support RPG II application programs.

### Assemblers and compilers

You can assemble or compile command-level applications programs using the assembler and compilers shown in Table 6.

Language	Compiler	Release number	Product number
Assembler	High-level assembler	1.10	5696-234
C	IBM C for VSE/ESA <b>1</b> <b>2</b>	1.1.0	5686-A01
COBOL	IBM COBOL for VSE/ESA <b>1</b> <b>3</b>	1.1.0	5686-068

Table 6 (Page 2 of 2). Languages and compilers supported by CICS			
Language	Compiler	Release number	Product number
PL/I	IBM PL/I for VSE/ESA <b>1</b> <b>4</b>	1.1.0	5686-069
<p><b>Notes:</b></p> <p><b>1</b> Language Environment® for VSE/ESA 1.4, program number 5686-094, is the required runtime environment for the supported high-level languages.</p> <p><b>2</b> CICS does not support C application programs compiled with the C/370 compiler. C/370 application programs must be recompiled using the IBM C for VSE/ESA compiler.</p> <p><b>3</b> CICS supports COBOL programs compiled with the DOS/VS COBOL and VS COBOL II compilers provided the programs are relinked using the LE libraries.</p> <p><b>4</b> CICS does not support PL/I application programs compiled with the DOS PL/I compiler. DOS PL/I application programs must be recompiled using the IBM PL/I for VSE/ESA compiler.</p>			

## ISC and MRO communication

CICS TS Release 1 supports intersystem communication (ISC) links with:

- Other CICS TS Release 1 systems
- CICS for VSE/ESA Version 2.3
- CICS Transaction Server for OS/390® Release 1
- CICS Transaction Server for OS/390 Release 2
- CICS Transaction Server for OS/390 Release 3
- CICS/ESA® Version 4 Release 1
- CICS on Open Systems
- CICS for OS/2®
- Transaction Server for Windows NT Version 4
- CICS/400®
- CICS/VM™
- IMS/ESA® Version 5 Release 1
- IMS/ESA Version 6 Release 1
- Any system that supports **Advanced Program-to-Program Communication** (APPC) protocols (LU6.2).

CICS TS Release 1 supports multiregion operation (MRO) for communication with:

- Other CICS TS Release 1 systems
- CICS for VSE/ESA 2.3 systems

---

## Part 1. Improved availability

This part describes availability improvements provided in CICS TS.

<b>If you want to...</b>	<b>Refer to...</b>
Find out how CICS code is reorganized into domains and which CICS control blocks are moved above 16MB.	Chapter 3, "Restructure of CICS internal code" on page 13
Find out how storage keys and a new read-only DSA help you protect CICS code and control blocks.	Chapter 4, "Storage enhancements" on page 21
Restart a failed CICS in place without having to rebind sessions.	Chapter 5, "VTAM single-node persistent sessions" on page 29
Manage the number of queued requests between interconnected CICS regions.	Chapter 6, "Intersystem session queue management" on page 35
Sign on to a remote system and have that signon remain valid over multiple conversations.	Chapter 10, "APPC (LU6.2) persistent verification" on page 61





---

## Chapter 3. Restructure of CICS internal code

CICS Transaction Server for VSE/ESA Release 1 introduces a new internal CICS architecture achieved through the use of well-established software engineering techniques. This new CICS structure improves the quality of CICS code and provides full 31-bit support.

CICS functions are partitioned into *domains*. Each domain encapsulates all functions (including domain initialization, termination, and so on), and associated state data that is accessible only from within the domain.

### Accessing domains

Domain functions can be accessed from other parts of CICS only by using a standard high-level *domain call interface*. Other parts of CICS can access domain state data only through a domain call.

Domains call one another by specifying the required *domain gate*, where previously modules specified the entry point of a target program.

A domain gate is an entry point into a domain. When a domain call is made, the domain identifies the control blocks to be used. The domain and gate together identify the program to be invoked. Gates allow requests that are common over the whole of CICS (such as SYNCPOINT, PREPARE, and COMMIT requests) to be integrated into the domain structure without having to create new linkage mechanisms on an ad hoc basis.

### The domains

The new architecture is shown in Figure 1 on page 14, and is followed by a brief description of each domain.

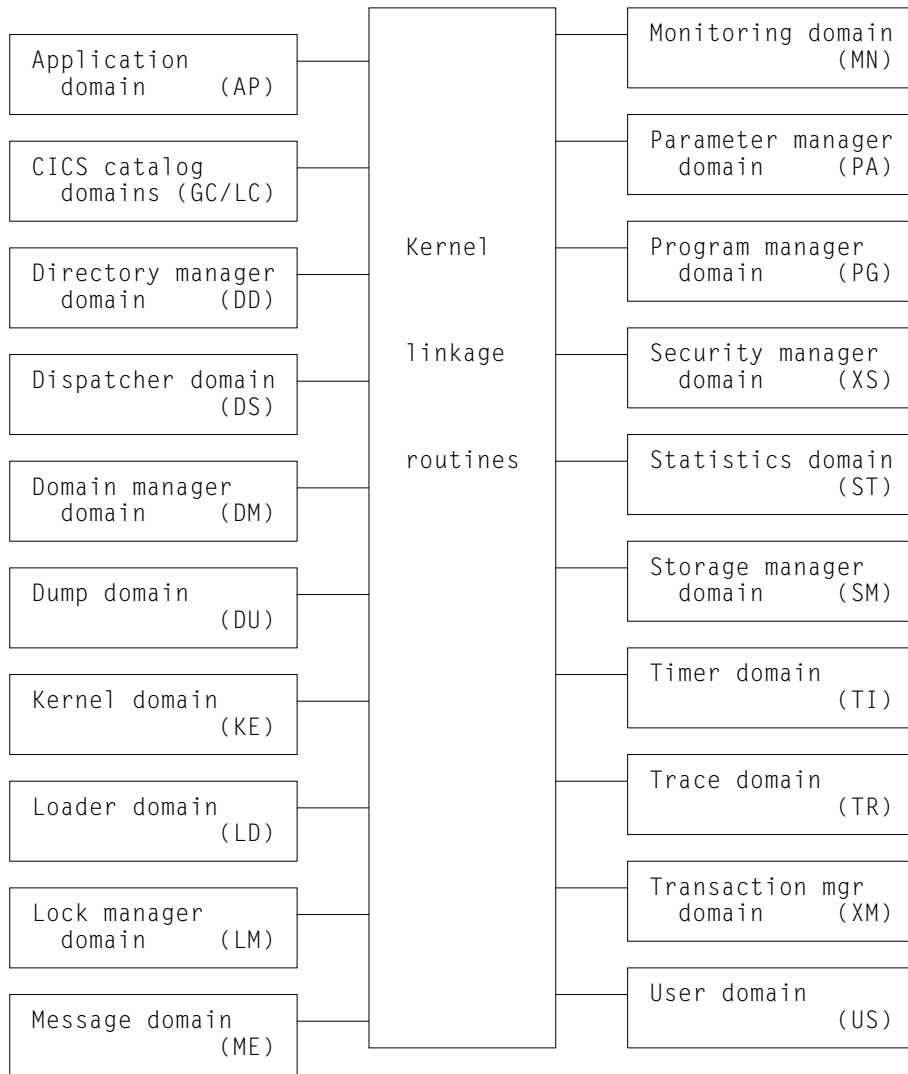


Figure 1. CICS domains. The domain identifiers are shown in parentheses

Table 8 (Page 1 of 4). CICS domains	
Domain	Function
Kernel (KE)	The main CICS control structure. The kernel tracks the existence of domains and is involved in every call from one domain to another, thereby providing a consistent linkage and recovery environment for CICS. Serious system errors can result in the termination of the kernel domain with a system dump containing diagnostic and problem determination information.
Application (AP)	Application programs are run in the application domain. Figure 1 on page 14 shows the major components of the AP domain.

Table 8 (Page 2 of 4). CICS domains

Domain	Function
CICS catalog domains (local and global)	<p>Used by the other domains to hold information to allow an orderly restart. They allow CICS code to read, write and purge records on the local and global catalog data sets so that a record of the CICS state can be maintained when CICS is not running. The catalog domains use a common set of programs to provide a domain interface to VSAM KSDS data sets, and they conceal the underlying VSAM operations from the calling domain.</p> <p>The local catalog contains information relevant to a particular CICS system, including a list of domains.</p> <p>The global catalog is used to hold information applicable to the whole CICS system. Therefore, in an XRF system consisting of one active and one alternate CICS system, there are two local catalogs and one global catalog. Conversely, in a non-XRF system, there is one local catalog and one global catalog.</p>
Directory manager (DD)	<p>A service component, providing resource table lookup services for the other domains (except the application (AP) domain, which uses the table manager program). The resources include:</p> <ul style="list-style-type: none"> <li>• Transaction definitions</li> <li>• Remote transaction definitions</li> <li>• Transaction classes</li> <li>• TPN names</li> <li>• User attributes</li> <li>• Programs</li> <li>• BMS mapsets and partitionsets</li> </ul> <p>The table manager program handles:</p> <ul style="list-style-type: none"> <li>• Terminals</li> <li>• Connections</li> <li>• Sessions</li> <li>• Modegroups</li> <li>• Files</li> <li>• Profiles</li> <li>• Autoinstall terminal models.</li> </ul>
Dispatcher (DS)	Controls the attaching, running and detaching of tasks, and controls the scheduling of VSE tasks.
Domain manager (DM)	Maintains (through the use of catalog services) permanent information about the status of individual domains. The domain manager domain also attaches initialization and termination tasks for the other domains.
Dump (DU)	Produces storage dumps and handles the associated data sets (transaction and system dump data sets) and status in the CICS system.
Loader (LD)	Used by the other domains to obtain access to storage-resident copies of nucleus and application programs, maps and tables.
Lock manager (LM)	Provides both locking and associated queueing facilities for CICS resources. Each resource is associated with a unique lock name that is used to access locking facilities.
Message (ME)	Acts as a centralized repository for storing and issuing CICS messages for all parts of restructured CICS. This ensures consistency of messages issued by the central controlling structures of CICS.

Table 8 (Page 3 of 4). CICS domains

Domain	Function
Monitoring (MN)	<p>Responsible for all monitoring functions in CICS.</p> <p>Monitoring data is written to DMF data sets, and can be used for subsequent processing by the monitoring utility program (DFHMNDUP).</p>
Parameter manager (PA)	<p>Controls the process of applying system initialization parameters to CICS. Initialization takes places using standardized interfaces that provide an improved method of communicating between functional areas of CICS.</p>
Program manager (PG)	<p>Supports the following areas of CICS:</p> <ul style="list-style-type: none"> <li>• Program control functions such as EXEC CICS LINK, XCTL, LOAD, RELEASE, and RETURN</li> <li>• Transaction abend and condition handling functions, such as EXEC CICS ABEND, HANDLE ABEND, HANDLE CONDITION and HANDLE AID</li> <li>• Related functions, such as invoking user-replaceable modules, global user exits, and task-related user exits</li> <li>• Autoinstall for programs, mapsets, and partitionsets.</li> </ul>
Security manager (XS)	<p>The security function (previously provided by the security identification program DFHACEE) is restructured to incorporate several key user requirements including:</p> <ul style="list-style-type: none"> <li>• Control of more than one signon by the same user</li> <li>• Improvements to security messages</li> <li>• Warnings when a password is about to expire</li> <li>• Increasing idle terminal signoff timeout.</li> </ul> <p>The restructure also provides an optional facility for checking user authority to run transactions and access resources.</p> <p>The security function is split into two domains; the <b>user</b> domain and the <b>security</b> domain. The security domain manages the security capabilities of users and handles all the interfaces to the external security manager (ESM).</p> <p>The end-user sees a new CICS API that:</p> <ul style="list-style-type: none"> <li>• Checks that a userid or a password is valid without actually performing a signon function</li> <li>• Extracts information relating to the expiry date of a valid password</li> <li>• Enables non-terminal security.</li> </ul>
Statistics (ST)	<p>Controls the collection of resource statistics for a CICS system. The statistics domain collects statistics data at a user-specified interval, at a system quiesce or at a logical end-of-day, and when requested to by the user. Statistics data is written to DMF data sets and can subsequently be used by the statistics offline utility (DFHSTUP) to produce formatted reports.</p>
Storage manager (SM)	<p>Manages virtual storage requests for CICS.</p>

<i>Table 8 (Page 4 of 4). CICS domains</i>	
<b>Domain</b>	<b>Function</b>
Timer (TI)	Provides internal timing and alarm clock services for the other domains. The timer domain also provides date and time provision and conversion functions. This includes the ability to synchronize the CICS local time with the operating system clock when the system operator has adjusted the time zone.
Trace (TR)	Used by CICS system code and user application programs to record details of the sequence of events occurring in the CICS system. The basic unit of information created for this purpose is the trace entry. The trace domain can write trace entries to: <ul style="list-style-type: none"> <li>• Internal trace, which is a wraparound table in main storage in the CICS address space</li> <li>• Auxiliary trace, which is a pair of CICS-controlled SAM data sets used alternately.</li> </ul>
Transaction manager (XM)	Provides transaction-related services to: <ul style="list-style-type: none"> <li>• Create tables</li> <li>• Terminate, purge and inquire on tasks</li> <li>• Manage transaction definitions and classes</li> </ul> <p>The transaction manager domain also provides a transaction environment that allows other CICS components to implement transaction-related services.</p>
User (US)	Restructures the function previously provided by the CICS sign-on facility.

## Virtual storage constraint relief

There are some changes to CICS structure that free storage below 16MB:

- The following are relocated above 16MB:
  - VTAM terminal control modules and control blocks
  - Most of the CICS nucleus and control blocks.
- File control enhancements eliminate the file work area (FWA) and use storage above 16MB whenever possible.
- The signon table (DFHSNT) is removed.

## VTAM terminal control modules and control blocks

The control blocks relocated above 16MB are:

- Those that are very large.
- Those that have many occurrences (more than one per terminal)
- Those that reside in storage for a long time (that is, are required for the lifetime of the CICS system or the terminal rather than for the lifetime of a task).
- Static terminal control blocks. These include all control blocks that are GETMAINed at initialization but are never FREEMAINed, and those that are attached to the TCTTE and are not FREEMAINed until the TCTTE is deleted.

The control blocks moved above 16MB include:

AWE	BIMG	CCE
CNS	LUC	LUM
LUS	LUCX	NIBD
NIB	RAIA	RPL
TIOA	TCTME	TCTSE
TCTTE	TCTUA	TPE
TTE	VTWA	

**Note:** The TCTTEs for SAM terminals are *not* moved above 16MB.

End-users benefit from this relocation because:

- A TOR can have more terminals connected.
- A TOR-AOR link can have more sessions connected.
- A AOR-FOR environment may show improved performance.

## CICS nucleus and control blocks

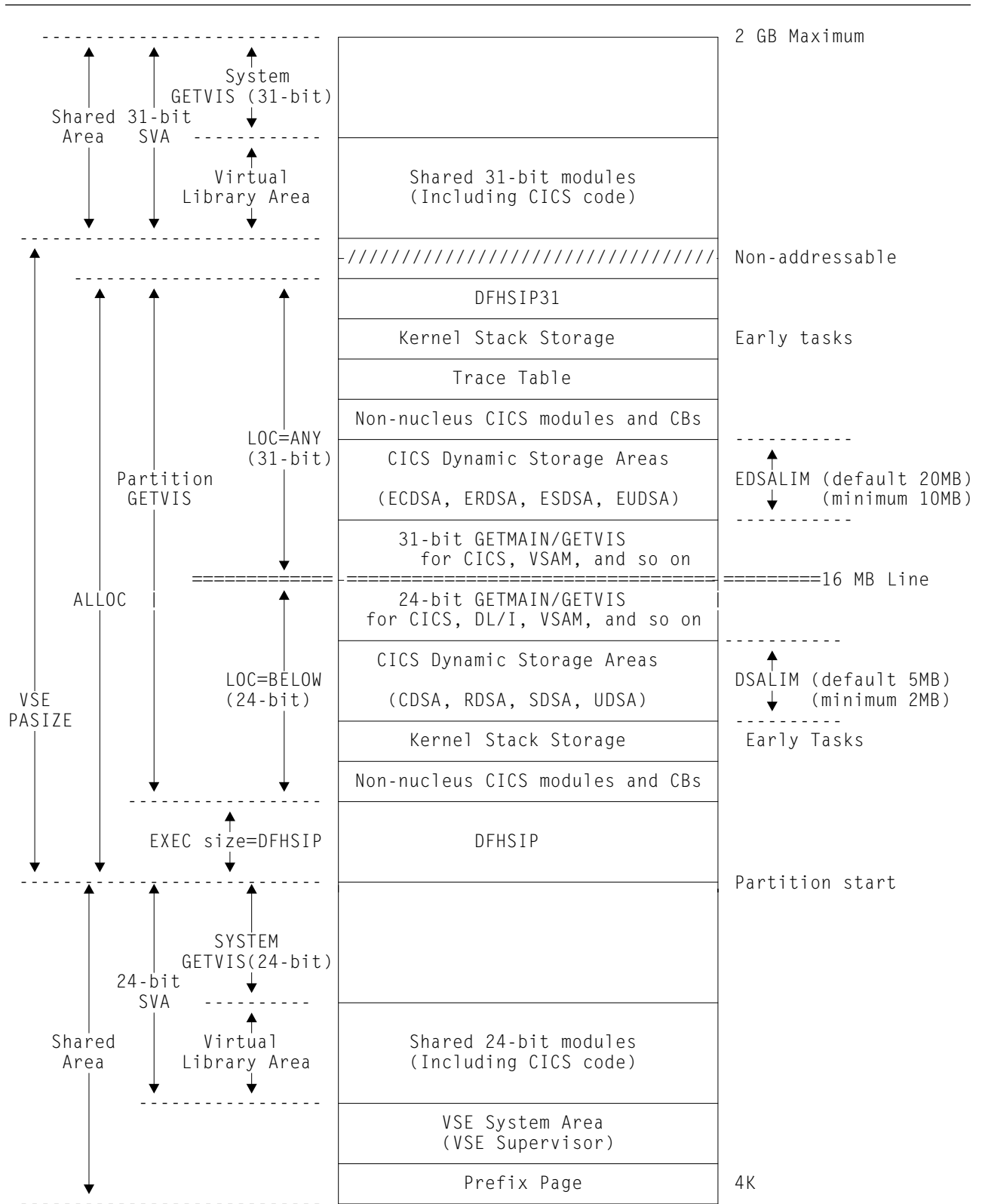
Most of the CICS nucleus is relocated above 16MB, except those parts that use services that do not support 31-bit addressing, including some code associated with transient data, journaling, transaction dumps, and auxiliary trace.

The following CICS control blocks are moved above 16MB:

AID	BMS mapsets	BMS control blocks
DCT intrapartition entry	DCT extrapartition entry	DSNB
FCTE (VSAM)	FCTE (DAM)	FFLE
FRTE	ICE	JCA
KCP	MRO control blocks	PCT entry
PPT entry	READ SET buffers	SHRCTL
SIT	TCA, EIB, EIS and TWA	TS unit table (TSUT)
TS table	TS control blocks	VSWA

## CICS Transaction Server for VSE/ESA Release 1 partition layout

Figure 2 on page 19 shows the CICS Transaction Server for VSE/ESA Release 1 partition layout.



# Figure 2. CICS Transaction Server for VSE/ESA Release 1 partition layout

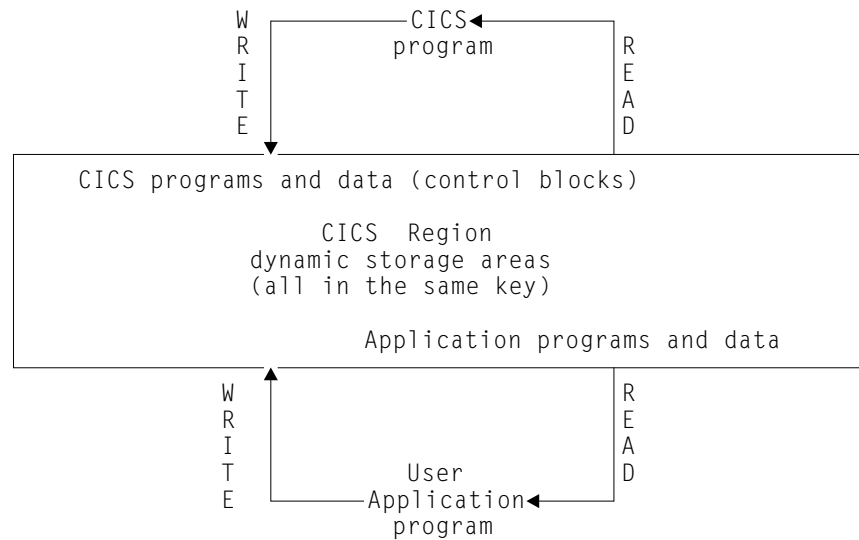




---

## Chapter 4. Storage enhancements

Earlier releases of CICS for VSE held user application programs, CICS programs, control blocks and application data together in the same storage key (the key of the partition). Except for the shared virtual area (SVA), there was nothing to prevent application programs overwriting CICS code and control blocks. Problems caused by overwritten storage are difficult to debug and can cause CICS failures and data corruption. This earlier situation is illustrated in Figure 3.



*Figure 3. Conceptual view of CICS and user programs in earlier releases.. This shows CICS data and control blocks, and user application programs and data, all residing in storage of the same VSE storage key.*

Storage protection in CICS enhanced with:

- An increase in the number of CICS-managed dynamic storage areas (DSAs) from one to eight—four above and four below 16MB. The individual sizes of these DSAs are determined and varied dynamically by CICS, and this enables continuous operation and simplified system management.
- The use of storage keys. Their use is optional but is dependent on your having the appropriate hardware and software (see Chapter 2, “Hardware and software requirements” on page 5 for details).

---

### Dynamic storage areas

CICS dynamic storage is divided into eight separate dynamic storage areas (DSAs)—four above and four below 16MB. The process of moving as much CICS code and as many control blocks as possible above 16MB continues, leaving more storage available below 16MB for user applications that require it. Only CICS code and data areas that have a dependency on certain VSE operating system interfaces are left below 16MB.

This in turn provides considerable virtual storage constraint relief (VSCR).

A consequence of the change is the provision of a number of new system initialization parameters, plus some new resource definition attributes that allow you

to specify whether CICS should acquire task-related storage above or below 16MB. An example is the new TASKDATALOC keyword on RDO TRANSACTION definitions. However, all of these new options default to below 16MB, so that existing applications can run unchanged.

For more detailed information, see the *CICS Resource Definition Guide*.

## Specifying storage requirements for DSAs

CICS allocates the total storage for DSAs according to the values you specify on the new DSALIM and EDSALIM system initialization parameters. These values represent the **upper** limit of the total storage that CICS can allocate to the DSAs.

The following DSAs are below 16MB:

- **The user DSA (UDSA)**  
The user-key storage area for all user-key task-lifetime storage below 16MB.
- **The read-only DSA (RDSA)**  
The key-0 storage area for all reentrant programs and tables below 16MB.
- **The shared DSA (SDSA)**  
The user-key storage area for any non-reentrant user-key RMODE(24) programs, and also for any storage obtained by programs issuing EXEC CICS GETMAIN commands for storage below 16MB with the SHARED option.
- **The CICS DSA (CDSA)**  
The CICS-key storage area for all non-reentrant CICS-key RMODE(24) programs, all CICS-key task-lifetime storage below 16MB, and for all CICS control blocks that reside below 16MB.

To specify a fixed size for these four DSAs, use the UDSASZE, RDSASZE, SDSASZE and CDSASZE system initialization parameters respectively. The default size is 0, indicating that the DSA size can change dynamically. A non-zero value indicates that the DSA size is fixed

CICS creates the following extended DSAs (EDSAs) above 16MB:

- **The extended user DSA (EUDSA)**  
The user-key storage area for all user-key task-lifetime storage above 16MB.
- **The extended read-only DSA (ERDSA)**  
The key-0 storage area for all reentrant programs and tables above 16MB.
- **The extended shared DSA (ESDSA)**  
The user-key storage area for any non-reentrant user-key RMODE(ANY) programs, and also for any storage obtained by programs issuing EXEC CICS GETMAIN commands for storage above 16MB with the SHARED option.
- **The extended CICS DSA (ECDSA)**  
The CICS-key storage area for all non-reentrant CICS-key RMODE(ANY) programs, all CICS-key task-lifetime storage above 16MB, and for all CICS control blocks that reside above 16MB.

To specify a fixed size for these four extended DSAs use the EUDSASZE, ERDSASZE, ESDSASZE, and ECDSASZE system initialization parameters respectively. The default size is 0, indicating that the DSA size can change dynamically. A non-zero value indicates that the DSA size is fixed.

You can vary these overall limits dynamically by using either the CEMT SET SYSTEM command or an EXEC CICS SET SYSTEM command, with the options DSALIMIT and EDSALIMIT.

---

## CICS storage keys

CICS exploits the ESA/390™ subsystem storage protection facility in a way that enables you to prevent CICS code and control blocks from being overwritten accidentally by your user application programs. It does not prevent:

- Deliberate overwriting of CICS code or control blocks. CICS cannot prevent a user application obtaining the necessary access key to modify CICS storage.
- Application programs and data being overwritten by another application program, although programs can be protected if they are written to reentrant<sup>1</sup> and AMODE(31),RMODE(ANY) standards, because CICS loads these programs into read-only storage, if requested. You can control this by means of the RENTPGM system initialization parameter. See the *CICS System Definition Guide* for information about the RENTPGM parameter.

### Requirements

CICS storage protection is optional. However, to use it you need the hardware facility that supports the VSE/ESA implementation. See “Hardware prerequisites” on page 5 for details.

CICS allows you to run your application programs in either *CICS key* or *user key*.

**User key** Defines the following storage and execution keys for user application programs:

1. The key of the storage in which user applications and their data areas normally reside. This storage is allocated in two DSAs—the user dynamic storage area (UDSA) below 16MB, and the extended user dynamic storage area (EUDSA) above it.
2. The execution (PSW) key with which application programs normally run. This gives user-key application programs read-write access to user-key storage, but read-only access to CICS-key storage.

User key is the CICS equivalent of the System/390® open key (key 9). Note that you cannot specify storage protection for CICS TS regions in the VSE/ESA F4 partition, because F4 has a fixed key of 9.

**CICS key** Defines the following storage and execution key for CICS programs:

1. The key of the storage in which CICS code and data areas normally reside. This storage is allocated in two DSAs—the CICS dynamic storage area (CDSA) is located below 16MB, and the extended CICS dynamic storage area (ECDSA) above it.
2. The execution (PSW) key that CICS runs with. This gives CICS read-write access to both CICS-key and user-key storage.

---

<sup>1</sup> reentrant. A program is considered reentrant if it is link-edited SVA attribute.

CICS key is the key in which CICS is given control by VSE at job step initiation (the partition key).

If you choose user key (the default), CICS storage is automatically protected from such application programs. The concept of isolating CICS code and control blocks from user application programs is illustrated in Figure 4.

If an application program executing in user key attempts to modify CICS-key storage, a protection exception occurs. This exception is processed by normal CICS error handling. The offending transaction terminates abnormally with an ASRA abend; it is not automatically disabled by CICS.

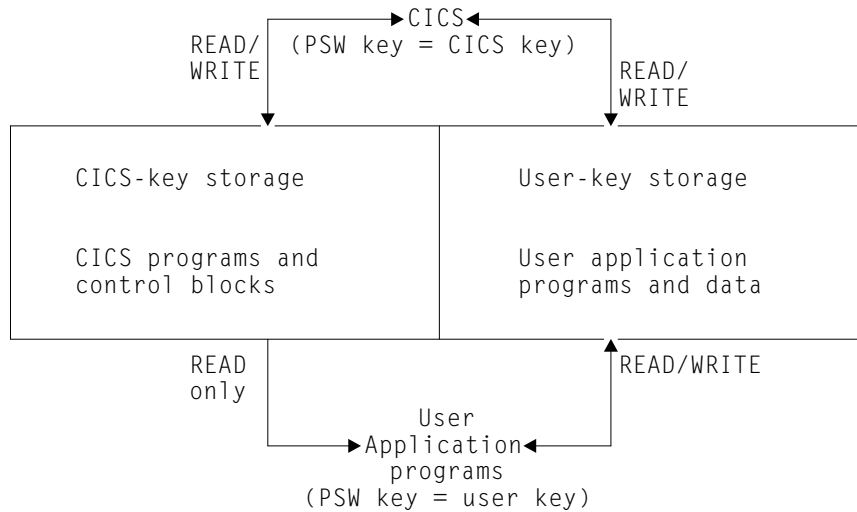


Figure 4. Protecting CICS code and control blocks from user application programs

---

## Using storage protection

This section describes how to set up your system to use storage protection, including the definition of either CICS-key or user-key storage by setting the following:

### TASKDATAKEY

Set this parameter of the TRANSACTION resource definition to specify the storage key of the storage that CICS allocates at task initialization for the duration of the task (task-lifetime storage), and which is accessible by the application. It also specifies the key of the storage that CICS obtains on behalf of all programs that run under the transaction.

### EXECKEY

Set this parameter of the PROGRAM resource definition to specify the key in which CICS gives control to the program, and to determine whether the program can modify CICS-key storage. For all except reentrant programs (that is, programs link-edited as SVA eligible), EXECKEY also defines, in conjunction with the residency mode, into which of the DSAs CICS loads the program.

### CWAKEY

Set this system initialization parameter to specify the storage key for the common work area (CWA), which is a single control block that is allocated at system CICS initialization and exists for the duration of that CICS session.

For detailed information about TASKDATAKEY or EXECKEY see the *CICS Resource Definition Guide* . For detailed information about CWAKEY, see the *CICS System Definition Guide* .

Set up storage protection as follows:

1. Specify the storage key for your application programs through the EXECKEY parameter on the PROGRAM resource definition.

The default is EXECKEY(USER) to ensure that, when you run with active storage protection, CICS code and control blocks are automatically protected without you having to change any existing PROGRAM resource definitions. You are recommended to run your applications in user key.

All the programs in a multi-program transaction must be defined with EXECKEY(CICS) if the associated transaction is defined with TASKDATAKEY(CICS). You cannot define a program to inherit its caller's execution key: the execution key and data storage keys are derived for each program from its PROGRAM and associated TRANSACTION resource definitions.

2. Identify programs to run in CICS key.

You can choose to run programs in CICS key, but CICS is not protected from them. The programs you might select to run in CICS key are typically those written by the system programmer, which are usually designed to provide special function in support of user applications. Such programs are generally considered to be an extension of CICS rather than part of an application.

CICS always gives control in CICS key to the following types of user-written program, regardless of the option specified on their PROGRAM resource definitions:

- Global user exits
- Task-related user exits
- User-replaceable modules
- Program list table (PLT) programs

The storage protection facility does not protect CICS code and control blocks from being overwritten by these types of program, or from ordinary application programs that you choose to execute in CICS key.

For more information about the execution of global user exits, task-related user exits, user-replaceable modules, and PLT programs in a CICS region running with storage protection, see the *CICS Customization Guide*.

3. Load programs into the ERDSA. Link-edit the programs with the SVA and RMODE(ANY) attributes. Use the TASKDATAKEY option to determine how data area storage is allocated for the ERDSA-resident programs.
4. Select the storage key for the CICS data area and application data areas that your applications can use.

You can choose the storage key for the common work area (CWA) and terminal control table user areas (TCTUAs) in a CICS region. If you specify user key, all programs have read-write access to these storage areas; if you specify CICS key, user-key application programs are restricted to read-only access.

Specify TASKDATAKEY(CICS) only for those transactions where all the component programs have EXECKEY(CICS), and for which you want to protect

their task life-time and working storage from being overwritten by user-key applications. For example, the CICS-supplied transactions, such as CEDF, are defined with TASKDATAKEY(CICS).

Storage area	Specifying the key
CWA (common work area)	CWAKEY system initialization parameter The default is user key.
TCTUA (terminal control table user area)	TCTUAKEY system initialization parameter The default is user key.
TWA (transaction work area)	TASKDATAKEY option on the TRANSACTION resource definition
EXEC interface block (EIB).	TASKDATAKEY option on the TRANSACTION resource definition.
The copies of working storage that CICS obtains for each execution of an application program.	TASKDATAKEY option on the TRANSACTION resource definition.
Any storage obtained for an application program in response to explicit or implicit storage requests.	TASKDATAKEY option on the TRANSACTION resource definition or the USERDATAKEY or CICS DATAKEY option on an EXEC CICS GETMAIN command. <b>Note:</b> The storage key options on an EXEC CICS GETMAIN command override the TASKDATAKEY option of the TRANSACTION resource definition.
Exits and user-replaceable programs	TASKDATAKEY option of the TRANSACTION resource definition of the calling transaction Some storage, such as global work areas, is always allocated from CICS-key storage.
PLT programs	For initialization PLT programs, storage is always allocated in CICS key. For shutdown PLT programs, the storage key is determined by the TASKDATAKEY option of the TRANSACTION resource definition of the transaction that invokes the shutdown command.

5. To use application development tools that rely on modifying the application program itself on programs loaded into the ERDSA, specify NOPROTECT on the RENTPGM system initialization parameter.
6. Set the STGPROT system initialization parameter to YES.

## Changes to CICS externals

Change	What it does	Where to find more information
System definition		
CWAKEY	Sets storage keys for the CWA (common work area)	<i>CICS System Definition Guide</i>

<b>Change</b>	<b>What it does</b>	<b>Where to find more information</b>
RENTPGM	Specifies whether CICS is to allocate the read-only DSAs from key 0, non-fetch protected storage.	<i>CICS System Definition Guide</i>
STGPROT	Enables storage protection	<i>CICS System Definition Guide</i>
TCTUAKEY	Sets storage keys for the TCTUA (terminal control table work area)	<i>CICS System Definition Guide</i>
Application programming		
EXEC CICS GETMAIN options: <ul style="list-style-type: none"> <li>• USERDATAKEY</li> <li>• CICS DATAKEY</li> </ul>	Sets storage keys for the application program's data storage areas. The setting on EXEC CICS GETMAIN overrides that on the transaction resource definition (TASKDATAKEY).	<i>CICS Application Programming Reference</i>
Resource definition		
EXECKEY option on program definition	Sets storage keys for programs	<i>CICS Resource Definition Guide</i>
TASKDATAKEY option on transaction definition	Sets storage keys for task life-time and program working storage.	<i>CICS Resource Definition Guide</i>
Problem determination		
Message DFHSR0622	<p>Informs the CICS console that either:</p> <ul style="list-style-type: none"> <li>• An application program executing in user key attempted to modify CICS key storage</li> <li>• A program resident in the RDSA or ERDSA is not truly resident and attempted to overwrite its own program storage.</li> </ul>	<i>VSE/ESA Messages and Codes Volume 3</i>
Additional diagnostic information passed to a user-written PEP for ASRA and ASRB abends	Includes the program name and key	<i>CICS Customization Guide</i>

Change	What it does	Where to find more information
<p>Additional diagnostic information in exception trace entries:</p> <ul style="list-style-type: none"> <li>• AP 0781 (for ASRA)</li> <li>• AP 0782 (for ASRB)</li> <li>• AP 0783 (for ASRD)</li> </ul> <p>and made available to any XPCTA and XPCHAIR global user exit programs that are enabled</p>	<p>Provides:</p> <ul style="list-style-type: none"> <li>• The name of the program in which the VSE abend or the program check occurred.</li> <li>• The offset at which the program occurred, if this can be determined.</li> <li>• The execution key at the time of the error (ASRA and ASRB only)</li> <li>• The name of any DSA that the program was attempting to overwrite (ASRA only).</li> </ul>	<p><i>CICS Customization Guide</i></p>



---

## Chapter 5. VTAM single-node persistent sessions

CICS support of VTAM persistent sessions exploits improvements in VTAM 4.2 to increase CICS availability, and provide restart-in-place of a failed CICS system (typically a TOR), without rebinding. In the event of a CICS failure, VTAM persistent sessions can:

- Save users having to reestablish sessions to regain their working environment.

When CICS has VTAM persistent sessions and fails, the user perception is that CICS is “hanging”: what is on the screen at the time of the failure remains until persistent session recovery is complete.

Without VTAM persistent sessions, all sessions existing on a CICS system are lost when that CICS system fails. In any subsequent restart of CICS, the rebinding of sessions that existed before the failure depends on the terminal’s AUTOCONNECT option. If AUTOCONNECT is specified for a terminal, the user of that terminal waits until the “good morning” (GMTRAN) transaction has run before being able to continue working. If AUTOCONNECT is not specified for a terminal, the user of that terminal has no way of knowing (unless told by support staff) when CICS is operational again unless the user tries to log on.

- Improve the availability of an entire CICSplex. Restarting a failed TOR is quicker, especially a CICSplex with one or more TORs that attach a large number of sessions.

Without VTAM persistent sessions, such a TOR may take a considerable time to restart after failure. If a CICSplex has only one TOR and it has failed, the entire CICSplex may become unavailable to end users. If a CICSplex has more than one TOR and one (or more) fails, large parts of the CICSplex may become unavailable to end users.

Support for VTAM persistent sessions in CICS TS Release 1 increases your options for providing system and user availability in the event of a CICS failure. You can choose:

- Emergency restart-in-place (no support for either XRF or VTAM persistent sessions)
- XRF takeover
- VTAM persistent sessions, complementing the basic emergency restart-in-place mechanism. Note that VTAM persistent sessions and XRF sessions cannot run simultaneously.

CICS support of persistent sessions includes the support of all LU–LU sessions except for LU0 pipeline and LU6.1 sessions.

If CICS is terminated through a CEMT PERFORM SHUTDOWN IMMEDIATE command, or if CICS fails, its sessions are placed in “recovery pending” state.

During emergency restart, CICS restores those sessions pending recovery from the CICS global catalog and the CICS system log to an “in session” state. This happens when CICS opens its VTAM ACB. From this state, CICS does one of three things:

1. Cleans up and recovers
2. Unbinds the sessions

### 3. Does not retain the sessions

CICS uses the time you set on the PSDINT system initialization parameter to determine for how long the sessions should be retained. If a failed CICS is restarted within this time and RECOVPTION on the TYPETERM resource definition is set to CLEARCONV or SYSDEFAULT, CICS can use the retained sessions immediately—there is no need for network flows to rebind them. If the terminal is not used during the period specified on the AIRDELAY system initialization parameter, the session is unbound.

#### Requirements

- **VTAM**

You need VSE/VTAM 4.2 with the inter-enterprise option installed. See “VSE/ESA Version 2 Release 4 requirements” on page 5 for more information about supported releases of VTAM in CICS TS Release 1.

- **Storage**

With VTAM persistent sessions, there is no significant change either in real storage requirements or in the storage required below the 16MB boundary. However, during the recovery process, VTAM persistent sessions require 75KB of virtual storage, plus 125KB of virtual storage per 1000 sessions, above the 16MB boundary.

If there is insufficient contiguous storage for this, the process is not aborted but smaller areas of storage are used. This means that more iterations of the code are required and processing is slower. If there is insufficient storage, VTAM persistent sessions is disabled, message DFHZC0003 is issued and an exception trace entry is made.

## Clean-up and recovery

Processing following emergency restart is LU dependent: cleanup and recovery for non-LU6 persistent sessions are similar to that for non-LU6 backup sessions under XRF. Cleanup and recovery for LU6.2 persistent sessions maintain the bound session when possible but there are cases where it is necessary to unbind and rebind the sessions, for example, where CICS fails during a session resynchronization.

After a successful CICS emergency restart, the recovery options defined on the RECOVPTION parameter of the TYPETERM resource definition for the terminals or sessions take effect.

**Note:** CICS does not maintain a user's signon in the event of a failure with persistent session support. If a user is signed on at the time of a CICS failure, they need to sign on again after their session is recovered, to re-establish their security environment.

## Unbinding sessions

CICS does not always reestablish sessions that are held by VTAM in a “recovery pending” state. However, CICS (or VTAM) unbinds “recovery pending” sessions in the following situations:

- CICS does not restart within the specified persistent session delay interval
- You perform a COLD start after a CICS failure

- CICS restarts with XRF=YES (when the failed CICS was running with XRF=NO)
- CICS cannot find a terminal control table terminal entry (TCTTE) for a session (for example, because the terminal was autoinstalled with AIRDELAY=0 specified)
- A terminal or session is defined with the recovery option (RECOVOPT) set to UNCONDREL or NONE
- A connection is defined with the persistent session recovery option (PSRECOVERY) set to NONE
- A *predatory takeover* occurs of sessions from one running CICS region to another (see below).
- Some other situation occurs in which APPC sessions are unbound, for example, if a bind was in progress at the time of the failure

In all these situations, the sessions are unbound, and the result is as if CICS has restarted following a failure without VTAM persistent session support.

### **Predatory takeovers**

If persistent session support is in use, a VTAM application with the same APPLID as that of an executing CICS system can assume control of the sessions of the executing CICS system. This is known as a predatory takeover.

An example of a predatory takeover is where two CICS regions are running under the same APPLID, but only one has VTAM open with active sessions. If you issue a SET VTAM OPEN (CEMT or EXEC CICS) command, VTAM passes the sessions to the second CICS.

CICS responds to predatory takeovers of this sort by always unbinding existing sessions when the VTAM ACB is dynamically opened.

### **Sessions not retained**

There are some circumstances in which VTAM does not retain LU–LU sessions:

- After a VTAM, VSE, or processor (CPC) failure.
- If you close VTAM with any of the following CICS commands:
  - SET VTAM FORCECLOSE
  - SET VTAM IMMCLOSE
  - SET VTAM CLOSED
- If you close the CICS node with the VTAM command VARY NET INACT ID=applid.
- If your CICS region performs a normal shutdown (with a PERFORM SHUTDOWN command).

### **Comparison of VTAM persistent sessions with XRF**

XRF was introduced in CICS/VSE® Version 2 to allow an alternate, partially initialized, CICS region to take over control from an active CICS system which had failed.

<b>VTAM persistent sessions</b>	<b>XRF</b>
Supports all LU types, except LU6.1 and LU0 pipeline sessions.	No support for LU6.2 connections and terminals, nor locally-attached terminals.
Require only a single CICS region	More complex system management because pairs of systems have to be defined and operated (instead of single systems)
User does not remain signed on after a persistent session recovery.	User remains signed on after an XRF takeover

---

## Using VTAM persistent sessions

- 1 Ensure that you have VTAM 4.2 (with the inter-enterprise option installed) and 75KB of virtual storage, plus 125KB of virtual storage per 1000 sessions, above the 16MB boundary.
- 2 Determine the recovery requirements of all terminals defined to CICS.
- 3 If you use FEPI applications, be aware of the effect of VTAM persistent sessions on the front- or back-end systems. See the *CICS Front End Programming Interface User's Guide* for details.
- 4 Enable persistent sessions by setting the persistent delay interval—the interval for which sessions should be retained— on the PSDINT system initialization parameter.  
  
You can change this interval using a CEMT SET VTAM or EXEC CICS SET VTAM command, but the changed interval is not stored in the CICS global catalog, so it is not restored in an emergency restart.
- 5 Ensure that the time interval set on the AIRDELAY system initialization parameter is sufficient.

---

## Changes to CICS externals

<b>Change</b>	<b>What it does</b>	<b>Where to find more information</b>
System definition		
PSDINT	Sets the interval for which sessions should be retained.	<i>CICS System Definition Guide</i>
Application programming		
New options on EXEC CICS INQUIRE and SET VTAM commands: <ul style="list-style-type: none"> <li>• PSDINTERVAL</li> <li>• PSDINTHRS</li> <li>• PSDINTMINS</li> <li>• PSDINTSECS</li> </ul>	Return, or set, the interval for which sessions should be retained.	<i>CICS System Programming Reference</i>
Resource definition		

<b>Change</b>	<b>What it does</b>	<b>Where to find more information</b>
PSRECOVERY option on the CONNECTION definition	Specifies whether, and how, LU6.2 sessions are recovered.	<i>CICS Resource Definition Guide</i>
RECOVNOTIFY option on the TYPETERM definition	Specifies how terminal end users should be notified that their session has been recovered.	<i>CICS Resource Definition Guide</i>
RECOVOPTION option on the TYPETERM and SESSIONS definitions	Specifies how you want CICS to recover the session and return the terminal to service	<i>CICS Resource Definition Guide</i>
CICS supplied transactions		
PSDINTERVAL option on CEMT INQUIRE and SET VTAM	Returns, or sets, the interval for which sessions should be retained.	<i>CICS-Supplied Transactions</i>
Problem determination		
Message DFHZC0003	Insufficient storage for the recovery process of VTAM persistent sessions.	<i>VSE/ESA Messages and Codes Volume 3</i>
Exception trace entry	Insufficient storage for the recovery process of VTAM persistent sessions.	<i>CICS Trace Entries</i>
Additional information in statistics	Return information about VTAM sessions, including the number that persisted, terminated, or were successfully restored.	<i>CICS Performance Guide</i>
Customization		
Predatory takeover	Controls how to proceed with a region where this condition has occurred	<i>CICS Customization Guide</i>
Additional information in NEP communication area	Provides recovery notification data	<i>CICS Customization Guide</i>



---

## Chapter 6. Intersystem session queue management

In the perfect world, work flow is even and there are enough intersystem sessions to prevent queueing. However, in the real world, workload varies and queues occur. Queues reduce the work flow between connected CICS regions and slow the end user. While an optimized level of queueing can be tolerated, abnormal or unexpected queueing should be dealt with or prevented.

For example, while waiting for free sessions, function shipping requests between CICS application-owning regions and connected file-owning regions can be queued in the issuing region. If a file-owning region deals with requests quickly, removing them from the queue, there will be an acceptable level of queueing. But if a file-owning region is unresponsive, the queue can become so long and occupy so much storage that the performance of connected application-owning regions is severely impaired. Further, the impaired performance of the application-owning region can spread to other regions. This condition is sometimes referred to as *sympathy sickness*, although it should more properly be described as intersystem queuing, which, if not controlled, can lead to performance degradation across more than one region.

To enable you to manage the number of queued requests, CICS Transaction Server for VSE/ESA Release 1 provides:

- New connection parameters
- A new global user exit, XZIQUE

---

### Connection parameters

CICS provides an internal solution based on values you specify for the QUEUELIMIT and MAXQTIME parameters of the CONNECTION resource definition. If you specify values for these parameters, CICS restricts any queues on the connection using the limits specified.

---

### The XZIQUE global user exit

XZIQUE enables a user-written exit program to detect queuing problems (bottlenecks) early. It allows you to deal with allocate requests originating from function shipping, transaction routing, and other forms of intercommunication request. This provides increased flexibility to help you to handle bottlenecks caused by these requests. An XZIQUE global user exit program can respond to situations in ways most appropriate for the special circumstances of each user installation.

The exit enables allocate requests to be queued or rejected, depending on the length of the queue. The exit also allows a connection on which there is a bottleneck to be cleared of its backlog.

---

## Using intersystem session queue management

- 1 Restrict the queues by specifying QUEUELIMIT and MAXQTIME parameters on the CONNECTION resource definition.
- 2 Use the sample XZIQUE program as a base.

---

## Changes to CICS externals

Change	What it does	Where to find more information
Resource definition		
QUEUELIMIT option on the CONNECTION definition	Specifies the maximum number of allocate requests that CICS is to queue while waiting for free sessions.	<i>CICS Resource Definition Guide</i>
MAXQTIME option on the CONNECTION definition	Specifies the time that queued allocate requests will wait for a free session.	<i>CICS Resource Definition Guide</i>
Global user exits		
XZIQUE	Enables allocate requests, originating from some form of intercommunication, to be queued or rejected, depending on the length of the queue.	<i>CICS Customization Guide</i>



## Part 2. Security

*Table 9. Security road map*

<b>If you want to...</b>	<b>Refer to...</b>
See a brief description of the new security domains	"Changes to resource definition" on page 40
Know about the general changes to resource definition for security	"Changes to resource definition" on page 40
Evaluate the changes made to transaction security	"Changes to transaction security" on page 40
Know how to uniquely identify individual system users to CICS	"Terminal user security" on page 43
Know about the obsolete operands affecting preset terminal security	"Preset terminal security" on page 44
Evaluate the changes made to sign-on procedures	"Changes to sign-on procedures" on page 44
Know about the new CICS default user function	"The CICS default user" on page 45
Know about changes that affect APPC (LU6.2) session security	"APPC (LU6.2) session security" on page 46
Know how changes to MRO bind and link security will affect you	"Multiregion (MRO) security (not supported by the BSM)" on page 47
Know about the new resource security checking function	"Resource security checking (not supported by the BSM)" on page 48
Know about the new command security checking function	"Command security checking (not supported by the BSM)" on page 53
Know about security changes affecting console operators	"Changes affecting console operators (not supported by the BSM)" on page 56
Look at the changes made to report security for the report controller	Chapter 8, "The report controller" on page 57
Look at the changes made to printer security for the report controller	"Printer security" on page 58
Learn about the new XRF sign-on retention facility	Chapter 9, "Enhancements to XRF" on page 59
Know more about overseer compatibility	"Overseer program compatibility" on page 60



---

## Chapter 7. CICS security

CICS internal security is withdrawn, and security in CICS is now provided through the services of an external security manager (ESM) that conforms to the system authorization facility (SAF) interface.

To provide the necessary security for CICS resources, CICS issues RACROUTE calls via the VSE/ESA Version 2 Release 4 SAF interface to route authorization requests to an ESM at appropriate points within CICS transaction processing. Note that if security is active, CICS issues security calls for all terminal users, including those that are not signed on. See “The CICS default user” on page 45 for information about the CICS default user ID, which provides security attributes for users that do not sign on.

You can choose between two levels of external security manager:

- The basic security manager (BSM) provided with the VSE/ESA 2.4 system package. The BSM is a limited-function ESM, and provides sign-on and transaction attach security only. User IDs and operator data previously defined in the CICS signon table (DFHSNT) are defined to the BSM using the VSE/ESA Interactive Interface (II).
- A full-function ESM, provided by an independent software vendor (ISV), that conforms to the SAF interface, and which provides support for all the various security checks issued by CICS.

It is important to secure your system. Using an ESM protects your application programs and CICS system components from damage (accidental or deliberate). CICS provides a variety of security and control mechanisms that limit the activities of CICS terminal users to only those functions that any particular individual is authorized to access.

Protect the following resources, because CICS does not do so:

- CICS own assets from interference from external access. Assets include program libraries, CICS regions, and so on.
- Data sets and databases used by CICS, and by CICS applications. These must be accessible to approved batch processing operations procedures only.
- Systems from application programs that use undocumented or unsupported interfaces to bypass CICS security.
- Application source libraries.

---

### The security domains

CICS internal security is replaced by two security domains. These domains incorporate the following new features:

- Control of more than one signon by the same user
- Improvements to security messages
- Warnings when a password is about to expire
- Increasing idle terminal sign-off timeout
- An optional facility for checking user authority to run transactions and access resources.

The two domains are:

- The **user** (US) domain  
The user domain provides all the facilities required for checking a user's authority to sign on to a terminal.
- The **security** (XS) domain  
The security domain manages the security capabilities of users and handles all the interfaces to the ESM. All interfaces to security services (for example, DFHSEC) are replaced with calls to the XS domain.

**Note:** An ISV-supplied ESM is required to take full advantage of all the new security features, because the BSM provided with VSE/ESA Version 2 Release 4 supports only limited function.

---

## Changes to resource definition

If you are using CICS internal security on CICS for VSE/ESA Version 2.3, you must check the security attributes on resource definitions, and define the required authorizations in your security database in place of the CICS internal resource security keys.

When you upgrade your CSD, the upgrade process updates the security keywords on all the CICS-supplied resource definitions in DFHLIST. This process includes removing the obsolete keywords, in addition to changing others.

## User-defined resource definitions

After you have upgraded the CICS-supplied definitions in your CSD, you must ensure that you upgrade any that you have copied and modified. Do this by deleting the modified CICS groups from your group list, and replace them with modified versions from the new DFHLIST. You must then review the security attributes in your own resource definitions. If you have any applications that use the obsolete CICS internal security, you must define your security requirements to an ESM.

---

## Vendor and in-house products

Any in-house or vendor products that use the undocumented DFHSEC macro to interface to CICS security facilities need to be changed. CICS offers only EXEC CICS SIGNON and EXEC CICS QUERY SECURITY commands. There are no XPI calls for security-related customization through global user exits.

---

## Changes to transaction security

CICS introduces transaction-attach security for non-terminal transactions. When transaction security is active (with system initialization parameters SEC=YES and XTRAN={YES|*classname*} specified), CICS always checks the authority of a user ID when attaching a transaction, even when the transaction is not associated with a terminal. This applies to:

- Transactions started when intrapartition transient data queues reach a trigger level
- Program list table (PLT) programs at startup

- Transactions started by an EXEC CICS START command
- System tasks that CICS creates when attaching its category 1 transactions (see below).

## Transaction categories for security checking and authorization

CICS-supplied transactions are classified into three categories for the purpose of security checking and authorization:

- Category 1** CICS internal-use-only transactions that no-one is permitted to use, and all terminal invocations of these must be denied.
- Category 2** CICS transactions that are supplied for initiation by a terminal user, or that are associated with a terminal. You must protect these transactions, and plan access lists carefully.
- Category 3** CICS transactions that everyone has access to without having to sign on. These transactions are exempt from security checks, and everyone can use them.

See the *CICS Security Guide* for details of all these transactions.

### Category 1 transactions

With the system initialization parameters SEC=YES and XTRAN={YES|*classname*} specified, CICS issues a security check against a user ID for *all* transactions (except for the special category 3 exempt transactions) at transaction-attach. For the CICS category 1 transactions, the security checks are performed early during CICS initialization against the CICS region user ID. Ensure that the user IDs under which you start your CICS regions are authorized to all the CICS transactions that CICS attaches as system tasks, otherwise system initialization fails.

## Started transactions

You can specify a user ID explicitly for transactions started by an EXEC CICS START command, where the command does not specify a terminal. This allows CICS to perform a transaction-attach security check, and, for users of full-function ESMs only, resource and command security checks, at execution time. A default mechanism is available. If you omit both the user ID and terminal from the START command, CICS associates the user ID of the transaction issuing the START command with the started transaction.

Transactions started by the EXEC CICS START TRANSID command (without specifying a TERMID or USERID) inherit the user ID associated with the initiating transaction.

The inherited user ID must be authorized to attach the started transaction. This means you must authorize user IDs to attach started transactions, when previously they only needed authorization to issue the start command for the transaction (in the XPCT class).

## Trigger-level transactions

If you migrate existing destination control tables that do not specify a user ID for trigger-level transactions, CICS issues the transaction-attach security check against the CICS default user ID.

## Program list table (PLT) programs at startup

You can specify a user ID for programs that are defined in a post-initialization program list table (PLTPI). These programs run under a new CICS-supplied transaction, CPLT, whose user ID is specified in the new system initialization parameter, PLTPIUSR.

## Surrogate user checking (not supported by the BSM)

To control non-terminal security, and the use of the CEDA INSTALL command for the installation of terminals with a preset user ID, CICS uses the surrogate user checking facility provided by an ESM (if it has one).

CICS uses this ESM facility to verify the authority of one user (the surrogate user) to submit a transaction, or install a preset terminal, on behalf of another user.

CICS performs surrogate user checks for:

- Non-terminal started transactions (that is, transactions initiated by an EXEC CICS START command that do not have an associated terminal).
- The installation of DCT entries during CICS initialization, for transactions initiated by a trigger level that do not have an associated terminal. A surrogate check is also performed if a user ID is dynamically set for a trigger-level DCT entry.
- The installation of program list tables at CICS initialization (PLTPI tables).
- The installation of terminals and sessions with preset security using the CEDA INSTALL command.

You do not have to extend your CICS security to use the surrogate user security checks, but if you do, you must ensure that the surrogate user profiles are defined for the appropriate user IDs. You enable surrogate user checking for non-terminal transactions, and for program list tables at initialization, by specifying XUSER=YES as a system initialization parameter. This is the default. To inhibit surrogate user checking, specify XUSER=NO.

The user IDs that must be authorized as surrogate users are:

### CICS region user IDs

The CICS region user ID must be authorized as a surrogate user of the following user IDs:

- All the user IDs specified on USERID parameters on DFHDCT TYPE=INITIAL and TYPE=INTRA macros for trigger-level transactions
- The CICS default user ID for trigger-level transactions that do not specify a user ID
- The user ID specified on the PLTPIUSR system initialization parameter.

### **User IDs of transactions that initiate START commands**

Any user that runs a transaction that issues an EXEC CICS START command, without an associated terminal, must be authorized as a surrogate of the user ID specified on the START command.

### **User IDs of transactions that specify ATIUSERIDs**

Any user of a transaction that issues an EXEC CICS SET TDQUEUE command to set a user ID for DCT trigger-level transactions must be authorized as a surrogate of the user ID specified on the ATIUSERID parameter.

### **User IDs of users of some CEDA INSTALL commands**

Any user that uses the CEDA INSTALL command to install terminals or sessions that have preset security must be authorized as a surrogate of the user IDs specified on the TERMINAL or SESSIONS definitions.

This function is controlled in previous releases by granting ALTER authority to the TERMINAL resource in the CCICSCMD general resource class.

### **Achieving compatibility with earlier versions**

Because of the extent of the changes, you might need time to migrate gradually to the new security facilities offered by surrogate user checking. One option is to set up your security in a way that achieves compatibility with earlier versions. You can do this by defining suitable generic *userid.DFHINSTL* and *userid.DFHSTART* profiles in the ESM SURROGAT general resource class.

For example:

- Initially, you can ensure that all surrogate checks succeed on START commands by defining a profile for \*.DFHSTART with UACC(READ), and progressively build profiles for specific users as required.
- You can ensure that nobody, unless properly authorized, can install terminals with preset security, by defining a profile for \*.DFHINSTL with UACC(NONE). You should then authorize only those users that currently have ALTER authority to the appropriate TERMINAL class profiles.

---

## **Terminal user security**

To secure resources from unauthorized access, CICS needs to be able to uniquely identify individual system users. You do this by defining your users to the VSE Interactive Interface (II).

Users identify themselves to CICS by specifying their user identification (user ID), and the associated password, or OIICARD (operator identification card), using the CICS-supplied sign-on transaction, CESN, or through the VSE/ESA II.

When a user signs on, CICS asks the ESM to verify the user ID and password against those held in its database. If the request to sign on is valid, the ESM creates an accessor environment element (ACEE), and passes the address of the ACEE back to CICS. Whenever CICS needs to determine if the signed-on user is authorized to perform a specific function, CICS passes this address as part of the request to the ESM.

---

## Preset terminal security

With the removal of the OPERRSL and OPERSEC parameters from the TERMINAL definition, preset terminal security is provided using the USERID parameter in an ESM environment. During CICS initialization (for dedicated terminals or during autoinstall processing) CICS issues a signon request for the user ID specified on the USERID parameter. Thus the terminal acquires the security defined for this user.

These changes mean that every CICS task must have an associated user ID, and therefore is effectively secured against accidental (or deliberate) integrity exposures.

---

## Changes to sign-on procedures

There are changes to the CICS sign-on and security facilities that affect the tasks of terminal operators and system programmers. You must update your operating procedures and, possibly, your installation programming standards to take these changes into account.

The process of signing on and off a CICS system is changed. You must use the new sign-on and sign-off transactions, CESN and CESF, rather than the CSSN and CSSF transactions, which are obsolete. The CESN and CESF transactions support more devices, but they do not include numeric-only terminals, or IBM 2741 and IBM 3741 devices with magnetic strip readers.

**Note:** Although CSSF is obsolete as a sign-off transaction at a user terminal, it is retained for use by old applications that issue EXEC CICS START CSSF commands.

One effect of removing CSSN is that your operators will need to sign on with an 8-character user ID, rather than with the 20-character operator name that could previously be used with CSSN.

The DFHSNP program, invoked by the CESN transaction, is changed and is now a pseudo-conversational transaction. DFHSNP conforms to the command-level application programming interface. Its source is no longer supplied in the VSE/ESA sublibrary, PRD1.BASE. The generated assembler-language code remains available on the optional source tape.

The syntax of the CESN sign-on command is as follows:

```
CESN [USERID=userid]  
    [,PS=password]  
    [,NEWPS=password]  
    [,LANGUAGE=language-code]  
    [,GROUPID=99999999] 1
```

### Notes

**1** The GROUPID keyword is only meaningful when you use an ESM. It does not apply to the BSM supplied by VSE.



If you enter only the transaction name “CESN” on a device that uses the IBM 3270 data stream, CICS displays a sign-on screen for you to enter your user ID and other data.

If you enter only the transaction name “CESN” on a device that does not use the IBM 3270 data stream, CICS prompts you to enter the required data.

The CESN transaction can be specified on a system initialization parameter as the “good morning transaction” (GMTRAN=CESN) so that it is started automatically without the operator having to invoke it.

You sign off by entering “CESF”, “CESF GOODNIGHT”, or “CESF LOGOFF”.

## Obtaining CICS-related data at signon

When handling the explicit signon of a CICS terminal user, CICS obtains the user's attributes by calling the ESM to request data from the user's CICS segment and LANGUAGE segment. If the CICS segment or the LANGUAGE segment data are present for the user, the ESM returns this data to CICS.

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 to CICS at system initialization.

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.  
A supported national language is a valid national language that has been specified on the NATLANG system initialization parameter and has the corresponding message definitions. See the *CICS System Definition Guide* for more information about defining the NATLANG parameter.
2. The PRIMARY(primary-language) parameter in the LANGUAGE segment of the user's ESM profile.
3. The SECONDARY(secondary-language) parameter in the LANGUAGE segment of the user's ESM profile.
4. The NATLANG parameter of the TERMINAL resource definition.
5. The language established for the default user

---

## The CICS default user

CICS uses the security attributes of the default user ID to perform all the security checks for any terminal users that do not sign on using the CESN sign-on transaction. These security checks apply to transaction-attach security checking (for ESMs and the BSM), and to resource and command security checking (ESMs only).

You define a default CICS user ID to the external security manager. When you have done this, define the default CICS user ID to CICS on the DFLTUSER system initialization parameter. 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 user ID specified on the system initialization parameter, initialization is terminated with a message.

The resource access authorizations that you give to the default user should be limited to those resources that you intend to make universally available, and therefore do not need to be restricted in any way. For example, on a development system where programmers need access to the CECI and CEDF transactions while developing their programs, you might create a default user ID that gives access to these transactions.

On a production system, however, the default user should not have access to any of the CICS-supplied transactions that are subject to security checking. See the *CICS Security Guide* for information about

If a task is started without an associated user ID, it is assigned the default user ID as specified on the DFLTUSER system initialization parameter. Generally, a task associated with the default user ID will have very limited authority.

Some resources now have additional keywords. For example, the destination control table (DCT) has a new USERID parameter that is used for trigger-level transactions not associated with a terminal.

The EXEC CICS START command is extended to include a new parameter, USERID, in support of the default user ID. The USERID parameter allows a task to run with the authority of the named user ID.

## Obtaining CICS-related data for the default user

When implicitly signing on the CICS default user during initialization, CICS obtains the attributes for the default user by calling 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 are present for the default userid, the ESM returns this data to CICS.

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:

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

---

## APPC (LU6.2) session security

The method you use for bind security on LU6.2 connections depends on the capabilities of your ESM:

- If your ESM supports bind-time session security, you can use the facilities provided by the APPCLU general resource class, and specify XAPPC=YES as a system initialization parameter. Defining BINDSECURITY=YES on the CONNECTION resource definition indicates that you want bind-time security checking on the connection, using the ESM's facilities. In this case, any BINDPASSWORD defined on the CONNECTION definition is ignored.

- If your ESM does not support bind-time security (for example, the BSM) you can continue to use the BINDPASSWORD option on the CONNECTION resource definition to implement bind-time security. To do this, specify:
  - XAPPC=NO as a system initialization parameter and
  - BINDSECURITY=NO on the CONNECTION resource definitionand CICS uses the BINDPASSWORD.

See the *CICS Security Guide* for information about LU6.2 bind security using APPCLU profiles.

---

## Multiregion (MRO) security (not supported by the BSM)

There are changes that affect two areas of MRO security:

- MRO bind security
- MRO link security

### Changes to MRO bind security

The SECURITYNAME parameter is no longer used for MRO bind-time security checking, nor is it used for any other security purpose on MRO links. The old CICS internal bind-time security mechanisms are replaced by calls to an ESM through the VSE/ESA SAF interface.

The CICS interregion communication program also uses the ESM to check that CICS regions logging on to IRC are the regions they claim to be.

This change to MRO logon and bind-time security also applies to earlier releases of CICS that use the CICS TS Release 1 interregion communication program, DFHIRP. As soon as the new DFHIRP is installed in the SVA, the ESM is used for bind security on **all** releases of CICS. As a consequence of this, the region user ID of the CICS for VSE/ESA Version 2.3 system needs to be defined with the correct ESM attributes.

### Changes to MRO link security

The SECURITYNAME parameter is obsolete for MRO connections. The primary method for specifying the user ID to be used for link security is the USERID parameter on the MRO SESSIONS definition.

To establish whether you want link security checking, or whether it is to be bypassed, CICS compares its own region user ID with the link user ID specified on the SESSIONS definition:

- If the user IDs are the same, link security checking is bypassed.
- If they are different, CICS “signs on” on the link user ID specified on the SESSIONS definition, and uses this on all link security checks.

#### The default link user ID

If you do not specify a link security user ID on the SESSIONS definition, CICS uses the region user ID of its partner as the default link user ID. To establish whether you want link security checking, or whether it is to be bypassed, CICS compares its partner’s user ID with its own region user ID:

- If the user IDs are the same, link security checking is bypassed.

- If they are different, CICS “signs on” on its partner’s region user ID as the link user ID and uses this for all link security checks.

When link security checking is bypassed, security checking is determined by the ATTACHSEC parameter on the CONNECTION definition. For information about the effect of the ATTACHSEC parameter, see Table 10.

<i>Table 10. MRO Security Options</i>		
Link Security	ATTACHSEC	
	LOCAL	IDENTIFY
Yes	CICS ignores the user ID associated with the transaction, and issues all security checks against the link user ID only.	CICS issues two security checks: <ol style="list-style-type: none"> <li>1. Against the user ID associated with the transaction</li> <li>2. Against the link user ID, ensuring that the transaction cannot access resources not authorized for the link.</li> </ol>
Bypassed	CICS issues all security checks against the CICS default user ID (the user ID specified on the DFLTUSER system initialization parameter)	CICS issues security checks against the user ID associated with the transaction.

### The APPL profile name for terminal-owning regions

All connected MRO regions pass the APPL profile name received from the local CICS region to which the terminal end user is signed on—the terminal-owning region—to the ESM. Thus users authorized to the APPL profile of the terminal-owning region to which they sign on are automatically authorized to all the MRO-connected regions in a CICSplex<sup>2</sup>

This does not apply to transaction routing or to function shipping requests that are passed over APPC links (that is LU6.2 links).

---

## Resource security checking (not supported by the BSM)

“Changes to transaction security” on page 40 describes how to control access to CICS transactions using transaction-attach security. This section 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 although a user may be authorized to invoke a particular CICS transaction, the user may not be authorized to access files or other general resources used within the transaction. Unlike transaction-attach security, which you cannot “switch off” for individual transactions, you can control resource security checking at the individual transaction level.

---

<sup>2</sup> A CICS complex. A CICSplex consists of two or more regions that are linked using CICS intercommunication facilities. For details see the *CICS Glossary*.

Resources defined to CICS to support application programs are also subject to security checking if resource or command security checking (see “Command security checking (not supported by the BSM)”) is specified.

You control who has access to the general resources used by CICS transactions by specifying:

- SEC=YES as a system initialization parameter
  - RESSEC=ALWAYS as a system initialization parameter
  - RESSEC(YES) in the TRANSACTION resource definition
- Note:** RESSEC is called RSLC in previous releases of CICS.  
RSLC(EXTERNAL) can be specified in compatibility mode only.
- The types of resource you want to protect by defining CICS system initialization parameters for the ESM general resource classes
  - The CICS resources to the ESM in resource class profiles, with appropriate access lists.

These resources are described briefly in Table 11, with the associated CICS system initialization parameter that you use to specify the ESM class names.

<i>Table 11 (Page 1 of 3). General resource checking by CICS</i>	
<b>System initialization parameter</b>	<b>Resource protected</b>
XAPPC	Partner logical units (LU6.2).
XCMD <b>1</b>	<p>The CICS system programming commands that are described in <i>CICS System Programming Reference</i>. EXEC CICS FEPI system commands are also controlled by this parameter.</p> <p>For XCMD=YES, the default resource class name is CCICSCMD, and the default grouping class name is VCICSCMD.</p> <p>If you choose your own name (XCMD=name), Cname is the resource class name and Vname is the grouping class name.</p>
XDCT <b>1</b>	<p>CICS extrapartition and intrapartition transient data queues. Define profiles in the ESM general resource class for TD queues to control who is allowed to access CICS transient data queues.</p> <p>For XDCT=YES, the default resource class name is DCICSDCT, and the default grouping class name is ECICSDCT.</p> <p>If you choose your own name (XDCT=name), Dname is the resource class name and Ename is the grouping class name.</p>

Table 11 (Page 2 of 3). General resource checking by CICS

System initialization parameter	Resource protected
XFCT <b>1</b>	<p>CICS file-control-managed VSAM and DAM files. Define profiles in the ESM general resource class for files to control who is allowed to access CICS VSAM and DAM files.</p> <p>For XFCT=YES, the default resource class name is FCICSFCT, and the default grouping class name is HCICSFCT.</p> <p>If you choose your own name (XFCT=name), Fname is the resource class name and Hname is the grouping class name.</p>
XJCT <b>1</b>	<p>CICS system log and journals Define profiles in the ESM general resource class for journals to control who is allowed to access CICS journals.</p> <p>For XJCT=YES, the default resource class name is JCICSJCT, and the default grouping class name is KCICSJCT.</p> <p>If you choose your own name (XJCT=name), Jname is the resource class name and Kname is the grouping class name.</p>
XPCT <b>1</b>	<p>CICS started transactions and EXEC CICS commands: COLLECT STATISTICS TRANSACTION, DISCARD TRANSACTION, CREATE TRANSACTION, INQUIRE TRANSACTION, INQUIRE REQID, SET TRANSACTION, and CANCEL. Define profiles in ESM general resource class for started-transactions to control who is allowed to start CICS transactions.</p> <p>For XPCT=YES, the default resource class name is ACICSPCT, and the default grouping class name is BCICSPCT.</p> <p>If you choose your own name (XPCT=name), Aname is the resource class name and Bname is the grouping class name.</p>
XPPT <b>1</b>	<p>CICS application programs. Define profiles in the ESM general resource class for programs to control who is allowed to invoke CICS application programs through transactions that issue a LINK, XCTL, or LOAD command.</p> <p>For XPPT=YES, the default resource class name is NCICSPPT, and the default grouping class name is MCICSPPT.</p> <p>If you choose your own name (XPPT=name), Mname is the resource class name and Nname is the grouping class name.</p>

<i>Table 11 (Page 3 of 3). General resource checking by CICS</i>	
<b>System initialization parameter</b>	<b>Resource protected</b>
XPSB <b>1</b>	<p>DL/I program specification blocks (PSBs). Define profiles in the ESM general resource class for PSBs to control who is allowed to access the DL/I PSBs used in CICS application programs.</p> <p>For XPSB=YES, the default resource class name is PCICSPSB, and the default grouping class name is QCICSPSB.</p> <p>If you choose your own name (XPSB=name), Pname is the resource class name and Qname is the grouping class name.</p>
XTRAN	<p>CICS transactions. Define profiles in the ESM general resource class for transactions to control who is allowed to run a CICS transaction.</p> <p>For XTRAN=YES, the default resource class name is TCICSTRN, and the default grouping class name is GCICSTRN.</p> <p>If you choose your own name (XTRAN=name), T:name is the resource class name and Gname is the grouping class name.</p>
XTST <b>1</b>	<p>CICS temporary storage destinations. Define profiles in the ESM general resource class for temporary storage queues to control who is allowed to access CICS temporary storage queues.</p> <p>For XTST=YES, the default resource class name is SCICSTST, and the default grouping class name is UCICSTST.</p> <p>If you choose your own name (XTST=name), Sname is the resource class name and Uname is the grouping class name.</p>
XUSER	Surrogate user security.

**Note for Table 11 on page 49:**

**1** You can use the default class name and resource name assigned to the resource during general resource security checking, or you can assign your own resource class name and group name.

See Appendix A, “System initialization parameters” on page 159 for more information about system initialization parameters.

**Note:** No authorization processing is done for BMS commands.

## New RESSEC keyword on the TRANSACTION resource definition

Specifying RESSEC=YES in the definition for a transaction together with the appropriate resource classes defined in the system initialization parameters provides security checking over and above transaction-attach security.

For most simple (or single-function) transactions, this extra security should not be necessary, but if you have complex or multiple-function transactions that offer users a choice of functions, or you are unsure about all the options available within a transaction, you may need this capability to restrict access to the data as well as to the transaction.

Before implementing resource security checking, you should consider the extra overhead that resource security checking involves, and only implement it if you believe the extra cost is worthwhile.

## The RESSEC system initialization parameter

You can impose the effect of RESSEC=YES for all CICS transactions by specifying ALWAYS on the RESSEC system initialization parameter. However, this is not recommended because:

- For most simple transactions, simply controlling access to the transaction is enough to control everything that the transaction can do.
- Invoking a resource check for every CICS resource results in extra overheads 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.

## Effects of resource security checking on CICS performance

#

A slight reduction in path length of a resource security check might be expected due to the removal of some interface layers.

Improvements in the amount of time taken to sign on the user IDs specified on the SECURITYNAME parameter for CONNECTIONS ("link security" user IDs) can be expected. This is because all such signons for the same user ID are shared; that is, only one invocation of RACROUTE REQUEST=VERIFY is performed.

## Resource security checking for DPL commands

CICS does not perform the security check in the case of a distributed program link (DPL) command, even when program resource checking is active. 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 local region. The security check is performed only in the CICS region in which the linked-to program 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 for MRO link security.



## Command security checking (not supported by the BSM)

You can use CICS command security to check system programming-type (SP) commands issued from CICS application programs, and the equivalent commands that can be issued from CEMT commands. Command security operates in addition to any transaction or resource security that may already have been defined for a transaction. The commands that are subject to command security checking are:

#

- ACQUIRE
- COLLECT
- CREATE
- DISABLE and ENABLE
- DISCARD
- EXTRACT
- INQUIRE
- PERFORM
- RESYNC
- SET.

The resources subject to command security checking are shown in Table 12.

<i>Table 12 (Page 1 of 2). Resources subject to command security checking</i>	
<b>Resource name</b>	<b>Related CICS commands</b>
AUTINSTMODEL	EXEC CICS, CEMT INQUIRE, and DISCARD AUTINSTMODEL
AUTOINSTALL	EXEC CICS and CEMT INQUIRE SET AUTOINSTALL
CONNECTION	EXEC CICS and CEMT INQUIRE SET CONNECTION and EXEC CICS CREATE CONNECTION
DSNAME	EXEC CICS and CEMT INQUIRE SET DSNAME
DUMP	EXEC CICS PERFORM DUMP and CEMT PERFORM SNAP
DUMPDS	EXEC CICS and CEMT INQUIRE SET DUMPDS
EXCI	CEMT INQUIRE SET EXCI
EXITPROGRAM	EXEC CICS ENABLE PROGRAM EXEC CICS DISABLE PROGRAM EXEC CICS EXTRACT EXIT EXEC CICS RESYNC ENTRYNAME
FEPIRESOURCE	Some EXEC CICS FEPI commands
FILE	EXEC CICS, CEMT INQUIRE SET, and DISCARD FILE and EXEC CICS CREATE FILE
IRC	EXEC CICS and CEMT INQUIRE SET IRC
JOURNALNUM	EXEC CICS and CEMT INQUIRE SET JOURNALNUM
LINE	CEMT INQUIRE SET LINE
LSRPOOL	EXEC CICS CREATE LSRPOOL
MAPSET	EXEC CICS CREATE MAPSET
MODENAME	EXEC CICS and CEMT INQUIRE SET MODENAME
MONITOR	EXEC CICS and CEMT INQUIRE SET MONITOR
PARTITIONSET	EXEC CICS CREATE PARTITIONSET

<i>Table 12 (Page 2 of 2). Resources subject to command security checking</i>	
<b>Resource name</b>	<b>Related CICS commands</b>
PARTNER	EXEC CICS, CEMT INQUIRE, and DISCARD PARTNER, and EXEC CICS CREATE PARTNER
PROFILE	EXEC CICS, CEMT INQUIRE, and DISCARD PROFILE, and EXEC CICS CREATE PROFILE
PROGRAM	EXEC CICS and CEMT INQUIRE SET, and DISCARD PROGRAM, and EXEC CICS CREATE PROGRAM
RECONNECT	CEMT PERFORM RECONNECT
REQID	EXEC CICS INQUIRE SET REQID
RESETTIME	EXEC CICS and CEMT PERFORM RESETTIME
SECURITY	EXEC CICS and CEMT PERFORM SECURITY REBUILD
SESSIONS	EXEC CICS CREATE SESSIONS
SHUTDOWN	EXEC CICS and CEMT PERFORM SHUTDOWN
STATISTICS	EXEC CICS, CEMT INQUIRE SET STATISTICS, COLLECT STATISTICS, and PERFORM STATISTICS RECORD
STORAGE	EXEC CICS and CEMT INQUIRE STORAGE
SYSDUMPCODE	EXEC CICS and CEMT INQUIRE SET SYSDUMPCODE
SYSTEM	EXEC CICS and CEMT INQUIRE SET SYSTEM
TASK	EXEC CICS and CEMT INQUIRE SET TASK and TASKLIST
TCLASS or TRANCLASS	EXEC CICS and CEMT INQUIRE SET TCLASS or EXEC CICS and CEMT INQUIRE SET TRANCLASS, and EXEC CICS CREATE TRANCLASS, and EXEC CICS DISCARD TRANCLASS
TDQUEUE	EXEC CICS and CEMT INQUIRE SET TDQUEUE
TERMINAL	EXEC CICS and CEMT INQUIRE SET TERMINAL and NETNAME and EXEC CICS CREATE TERMINAL
TRACEDEST	EXEC CICS and CEMT INQUIRE SET TRACEDEST
TRACEFLAG	EXEC CICS INQUIRE SET TRACEFLAG
TRACETYPE	EXEC CICS INQUIRE SET TRACETYPE
TRANDUMPCODE	EXEC CICS and CEMT INQUIRE SET TRANDUMPCODE
TRANSACTION	EXEC CICS, CEMT INQUIRE SET, and DISCARD TRANSACTION and EXEC CICS CREATE TRANSACTION
TSQUEUE	EXEC CICS and CEMT INQUIRE TSQUEUE
TYPETERM	EXEC CICS CREATE TYPETERM
VTAM	EXEC CICS and CEMT INQUIRE SET VTAM

You can request command security checking by coding the CMDSEC option on a TRANSACTION resource definition. For each of these commands, it is the *resource* on which the command operates that the user authorizes, and not the command itself.

When running CICS with command security checking, CICS calls the ESM to check whether a terminal user is authorized to use a command that operates on specific resources. In this context, the resource is the *object* specified on the CREATE, INQUIRE, SET, PERFORM, or COLLECT command (for example, FILE, PROGRAM, TRANSACTION and so on), and **not** the 8-byte DFHEIxxx module name, as is the case in CICS for VSE/ESA Version 2.3.

CICS raises a NOTAUTH condition if a user is not authorized for the resource associated with the command, or if the resource identifier is not defined to your ESM.

**Note:** Command security checking is not supported by the basic ESM available as part of the VSE/ESA Version 2 Release 2 system package. It is only available with a full-function ESM.

For details of CICS command security see the *CICS Security Guide*

## Distinguishing between READ, UPDATE, and ALTER intent

If you specify SEC=YES and XCMD=YES as system initialization parameters, CICS requires an access level appropriate to the access intent associated with the command. If you want to use this facility, ensure that your ESM can distinguish between READ, UPDATE, and ALTER access intent, and define the appropriate access levels for all users of transactions that issue SP-type commands (see Table 13 for details). The default on the SEC system initialization parameter is YES, but the XCMD is NO.

<i>Table 13. Required authorizations for CICS command types</i>	
<b>Permitted access</b>	<b>Command types</b>
READ access	INQUIRE and COLLECT
UPDATE access	SET, PERFORM, DISCARD, ENABLE, DISABLE, EXTRACT and RESYNC
ALTER access	CREATE

## Using command and resource security together

If you specify both command and resource security checking, CICS issues two calls to the ESM for each command, the first to check the command, and the second to check the user's access to the resource on which the command operates. For example, if you specify both command and resource security on a transaction that issues the command:

```
EXEC CICS SET FILE('FILEA') OPENSTATUS(18)
```

CICS first issues a call to check that the user is authorized to use the SET FILE command. If that call is successful, CICS issues the second to check that the user is also authorized to access the file called FILEA.

## Specifying RESSEC and CMDSEC for non-terminal transactions

Specifying the RESSEC and CMDSEC options on TRANSACTION resource definitions controls resource and command security checking on all transactions, including transactions that are attached without an associated terminal. If you specify resource or command security checking for a non-terminal transaction, the security check succeeds or fails depending on the access authority of the user associated with the transaction.

---

## Changes affecting console operators (not supported by the BSM)

CICS passes the console name to the ESM as the “port-of-entry” name at sign-on. Users who need to sign on to CICS from VSE consoles must be given READ access to a CONSOLE class profile of the same name as the console name.

---

## Changes to system initialization parameters

The following security function system initialization parameters are new or are changed:

- ESMEXITS (new)
- PLTPIUSR (new)
- PLTPISEC (new)
- SEC (new)
- SECPRFX (new)
- XCMD (new)
- XDCT (changed)
- XFCT (changed)
- XJCT (changed)
- XPCT (changed)
- XPPT (changed)
- XPSB (changed)
- XTST (changed)
- XUSER (new)

For a brief summary of these parameters, see Appendix A, “System initialization parameters” on page 159. For a full description, see the *CICS System Definition Guide*.

---

## Chapter 8. The report controller

There are some general enhancements to the report controller facility, and changes to report security and printer security.

---

### Enhancements to the report controller

The following enhancements are made to the report controller:

- Default forms control buffer (FCB) specification for printers

This allows a specific FCB to be associated with a specific printer. The specification is made in the TERMINAL resource definition for the printer, and can be altered with the RCF printer control panels.

- Control of initial skip processing

VSE/POWER provides the facility to specify what actions should be taken when a report that does not begin with a 'skip to channel 1' control command is printed. The facility is provided at both global and individual printer level. The report controller provides complementary support for its printing tasks.

- Additional formatting information for Escape format reports

For Escape reports that are formatted with MCC or ASA control characters internally, the report controller provides a second temporary storage queue from which the user can retrieve the control character information associated with the data records.

- NLS enablement for the Report Browse panel

The Report Browse function is enhanced to allow the heading and function key information of its display panel to be translated to another national language.

- Support for double-byte character set (DBCS) data by browse function

Reports containing DBCS data can now be viewed using the browse facility.

- Support for VSE/POWER SYSID

In a shared spool environment a specific VSE image can be specified to process a particular spool file. Support is added to enable you to specify a SYSID when creating a report with a new SPOOLOPEN keyword, PWRSYSID. The CEMS control transaction provides complementary support for this new function.

- Restart of the new CXPB transaction

In the event of a cancellation of the internal RCF transaction that controls the attachment of printers, CICS provides an option to restart the task from the CEMS operator panel.

---

### Report security

There are no CICS internal security facilities in CICS TS Release 1. To implement security for reports requires the services provided by a ISV-supplied external security manager (ESM) product. (Note that the BSM does not provide security facilities for the report controller). To provide security compatibility in an

environment managed by an ESM, the report controller requires three sets of resource names defined in the ESM's FACILITY general resource class:

- For report security, define profiles for DFHRCF.RSL01 through DFHRCF.RSL24 and DFHRCF.RSLPU
- For printer security, define profiles for DFHRCF.PRSL01 through DFHRCF.PRSL24 and DFHRCF.PRSLPU
- For browse security, define profiles for DFHRCF.BRSL01 through DFHRCF.BRSL24 and DFHRCF.BRSLPU

In each case, the 01 through 24 and PU suffix in these profile names corresponds to the RSL value of a report or printer (see "Printer security" for information about printer RSLs). For more information about resource security, see the *CICS Security Guide*.

Reports are protected by the DFHRCF.RSL $nn$  resource profile where  $nn$  is the RSL value of the report. Whenever a program attempts to append to a report (log and resumable reports), or a CEMS or CEOS user attempts to access a report, CICS calls the ESM to ensure the user has the required UPDATE authority.

See the *CICS Report Controller Planning Guide* for more information about the report controller.

---

## Printer security

The report RSL value is also used when printing a report on a CICS terminal printer. In this case, the user ID associated with the terminal printer must have READ authority to DFHRSL.RSL $nn$ .

A printer intended for RCF use can be given for control purposes. You specify this using the SPOOLPRTRSL keyword in the terminal resource definition for the printer. User of the CEMS or CEOS transaction require UPDATE authority to DFHRCF.PRSL $nn$  to be able to control the printer.

---

## Browse security

The implementation of the report controller browse function is changed to use the facilities of an ESM.

In earlier releases of CICS, authority to browse is controlled using the RSL of a dummy program, DFHPSBRS, which has to match one of the user's RSL keys.

To browse a report under the new ESM security facilities, a user requires READ authorization to the relevant DFHRCF.BRSL $nn$  profile, where  $nn$  is the RSL value of the report.

---

## Chapter 9. Enhancements to XRF

There are several enhancements to the CICS extended recovery facility in CICS TS Release 1. These enhancements present no obstacle to migration, but it is worth taking note of some of the related changes that you may need to make if you want to make the best use of the new facilities. For full details, see the *CICS XRF Guide*.

---

### XRF sign-on retention

The introduction of a mechanism for retaining a terminal's signed-on status over an XRF takeover means that you can now decide whether your XRF-capable terminals can remain signed on following a takeover. For example, those that are in secure areas can be allowed to remain signed on, while those that are not can be signed off.

You can control which terminals are allowed to remain signed on after a takeover by using the following parameters, which are new in CICS TS Release 1:

**XRFSIGNOFF({NOFORCE|FORCE})**

Specify this attribute on the TYPETERM resource definition to control sign-off for groups of terminals.

**XRFSOFF={NOFORCE|FORCE}**

Specify this system initialization parameter to control sign-off globally for the CICS system.

The default is NOFORCE, meaning that terminals are not signed off, so that, unless you specify these new parameters, you can migrate your existing system definitions without any effect.

### Controlling XRF timeout

If a terminal remains signed on over an XRF takeover, but the operator leaves during the takeover period, there is a potential security exposure. To keep the system secure, you can specify a timeout period for the whole system, so that terminals that are retained over a takeover but are then not used for the specified period can be automatically signed off. You do this by specifying a timeout period that begins when a takeover is started.

You do this using the system initialization parameter, XRFSTME, as follows:

```
XRFSTME=nn|5
```

where *nn* is in minutes.

---

### XRFTRSZ SIT parameter removed

The XRFTRSZ system initialization parameter is obsolete because the trace table is acquired above the 16MB in CICS TS Release 1, and has a fixed size of 64K.

---

## Change to the APPLID system initialization parameter

In CICS TS Release 1 the generic and specific APPLIDs must not be the same if you have also specified XRF=YES as a system initialization parameter. If the generic applid and the specific applid are the same, and XRF=YES is specified, CICS issues message DFHXG6477.

See the *CICS System Definition Guide* for information about the APPLID system initialization parameter.

---

## Overseer program compatibility

There is a change to the format of records that the CICS availability manager (CAVM) writes to the CAVM data sets. Because of this, you cannot use an overseer from an earlier release of CICS to oversee CICS TS Release 1 XRF regions. However, the CICS TS Release 1 overseer does cater for both the old and the new CAVM records. As a result, if you are operating CICS with XRF at different release levels, and you want to use an overseer, you must use a CICS TS Release 1 version. If you use the CICS-supplied sample overseer program, you can continue to use the existing program as long as you reassemble and link-edit it against the CICS TS Release 1 libraries.

The CICS TS Release 1 sample overseer does not take advantage of any new data passed to the overseer by CICS. See the *CICS Customization Guide* for information about writing an XRF overseer program.



---

## Chapter 10. APPC (LU6.2) persistent verification

Persistent verification allows you to distribute parts of your applications out of a CICS host onto an APPC (LU6.2) device such as a PC. It enables you to make a valid signon to a remote system over multiple conversations for as long as you need it.

Persistent verification:

- Provides at least the same level of security as that provided with LU2 terminal attachment.
- Minimizes the number of a times a password needs to be sent to a remote system, thereby improving response times and decreasing overheads.
- Reduces the number of calls made to an external security manager (ESM), thereby reducing the number of unnecessary I/O operations.

### Requirements

You **must** have an LU6.2 device that can send out persistent verification indicators. See Chapter 2, "Hardware and software requirements" on page 5 for a list of supported devices.

To sign on, you must provide a valid user ID and password in the attach to the remote system. When a sign on request is processed, assuming that your user ID and password are valid, CICS adds your user ID to a table called the persistent verification "signed-on-from" list. After you have successfully signed on to a remote system, you can request access to secure resources owned by CICS without having to supply a password in attach requests.

Your entry remains in this list until one of the following happens:

- You sign off.
- The entry is timed out.
- The connection to your system is lost.
- CICS is restarted.
- CICS receives an invalid attach request from you.

---

## Using persistent verification

1. Set the length of time an entry can be in the "signed-on list" with the PVDELAY system initialization parameter.
2. Enable connections and terminals for persistent verification with the PERSISTENT and MIXIDPE options on the ATTACHSEC parameter of TERMINAL and CONNECTION resource definitions.

## Changes to CICS externals

Change	What it does	Where to find more information
System definition		
PVDELAY	Sets the length of time for which an entry can be in the "signed-on-from list".	<i>CICS System Definition Guide</i>
Resource definition		
PERSISTENT option on ATTACHSEC on CONNECTION and TERMINAL resource definitions.	Requires a userid and password to be supplied on the first attach request.	<i>CICS Resource Definition Guide</i>
MIXIDPE option on ATTACHSEC on CEDA DEFINE CONNECTION and CEDA DEFINE TERMINAL	Supports incoming attach requests that are using either PERSISTENT or IDENTIFY security types.	<i>CICS Resource Definition Guide</i>
Monitoring		
Statistics written to DMF data handler data sets	Provides new information relating to persistent verification, for example, PVDELAY and reused sign-on entries.	<i>CICS Performance Guide</i>

---

## Part 3. Improvements to application programming

This part describes the features provided to improve application programming.

*Table 14. Road map for application programming*

<b>If you want to...</b>	<b>Refer to...</b>
Find out which application programming functions are obsolete or changed.	Chapter 11, "Changes for application programming" on page 65
Use CPI-C to design applications that can be easily integrated with other applications and run in multiple environments.	Chapter 12, "CPI-C" on page 69
Understand how to use front-end CICS programs.	Chapter 13, "FEPI" on page 71
Use the external CICS interface to enable a non-CICS program to communicate with a CICS program.	Chapter 14, "EXCI" on page 77
Cancel AIDs for terminals and connections.	Chapter 15, "Cancel start requests" on page 83
Use shared data tables for improved file control.	Chapter 16, "Shared data tables" on page 85



---

## Chapter 11. Changes for application programming

CICS continues to extend the CICS application programming interface (API), a process started in CICS for VSE/ESA Version 2.3. The API includes functions that were previously only available using the macro-level programming interface or by directly accessing CICS control blocks.

All applications that use commands documented in the *CICS Application Programming Reference* manual are source- and object-compatible, provided that the function is still supported. Execution of obsolete commands causes the application program to abend.

Note that the preceding paragraph is a statement of upward-compatibility only. There is no guarantee that every program translated or link-edited with the CICS TS Release 1 link-edit stub will be able to run on an earlier version or release of CICS for VSE.

---

### Obsoleted function

The following functions are removed:

- **DFHEICAL macro**

In previous releases of CICS for VSE, the translator generated a DFHEICAL macro for EXEC CICS assembler rather than the DFHECALL macro. The DFHEICAL macro continued to be shipped so that pre-translated program segments could continue to be used without retranslation. However, the DFHEICAL macro is no longer supported. Any program segments that need to be assembled also need to be retranslated. The object code generated is unchanged.

- **Access to CICS control blocks**

CICS restricts access to CICS control blocks by:

- Withdrawing support for the EXEC CICS ADDRESS CSA command to prevent command-level applications from accessing CICS internal control blocks through the CSA.
- Introducing a new trapping mechanism for CICS macro invocations and references to CICS internal control blocks by non-CICS code at execution time (see “Trapping of CICS macro invocations and references to control blocks” on page 67).

These control blocks are available only when passed explicitly as parameters to user exits and user-replaceable modules, and, even then, only particular fields are supported.

The CSA and TCA addresses are no longer passed to user exits.

- **Programming languages**

CICS provides upward compatibility at source and object level for command-level programs written in assembler language, COBOL, and languages supported by Language Environment for VSE/ESA (LE/VSE) that executed correctly under CICS/VSE Version 2, subject to the constraints outlined in Table 15.

Language	Restriction
All languages	The function must still be supported, otherwise execution of obsolete commands causes the program to terminate abnormally.  Source compatibility with other members of the CICS family is maintained, subject to the defined common subset of the CICS API.
Assembler language	Assembler language macro-level application programs are <b>not</b> supported.
C	C/370 command-level application programs are <b>not</b> supported. All C/370 programs must be recompiled using the IBM C for VSE/ESA compiler.
COBOL	COBOL macro-level application programs are <b>not</b> supported.
PL/I	CICS does <b>not</b> support DOS PL/I command- or macro-level programs. All DOS PL/I programs must be recompiled using the LE-conforming IBM PL/I for VSE/ESA compiler.
LE/VSE	Programs written and compiled using the LE-conforming compilers that are currently available for COBOL, PL/I, and C, are both source and object compatible, providing that they do not execute obsolete commands.
RPG	RPG application programs are <b>not</b> supported.

In addition, a degree of object compatibility may be possible between application programs compiled using the LE-conforming compilers and generated against CICS TS Release 1 and CICS for MVS/ESA™. However, there are instances where retranslation and recompilation are necessary for particular commands. For example, the EXEC CICS WRITE OPERATOR command is not object compatible. In these cases, you must retranslate and recompile the application programs when moving them between operating system platforms.

- **Macro-level application programs**

Support for the macro-level programming interface is withdrawn. CICS programming enhancements are made only to the API commands.

To protect your investment in application development, new application programs must be written using the CICS command-level API, and must conform to the general-usage application programming interface described in the *CICS Application Programming Reference* and the *CICS System Programming Reference* manuals.

You must decide if your existing macro applications have a future— if so, you should aim to convert (or replace) them as soon as possible.

If you cannot convert (or rewrite) macro-level programs immediately, you will need to retain a copy of CICS for VSE/ESA Version 2.3 to run these programs. In addition, they will not benefit from the API enhancements.

The macros affected are those documented in the following manuals:

- The *Application Programming Reference manual (Macro Level)* for CICS/VS 1.7
- The *CICS/DOS/VS 1.7 Customization Guide*
- The *CICS/VSE Customization Guides* for CICS/VSE Version 2.

CICS abends macro-level programs with abend code ASRD.

## Trapping of CICS macro invocations and references to control blocks

This mechanism causes any non-CICS code (written in any language) to abend if it attempts to invoke a CICS macro or access a CICS internal control block.

Whenever the address of the CICS CSA or TCA is passed across an EXEC or LINK or XCTL boundary, or to a user exit, the storage addressed resides in a non-user-key fetch-protected subpool. Unauthorized CICS macro invocations, or attempts to reference internal CICS control blocks result in a program check. CICS takes the following actions:

1. Issues a message (if possible) naming the offending program or transaction.
2. Abends the program or transaction. If the unauthorized access attempt was issued from a global user exit, then the region abends.

CICS can disable rather than abend a user transaction that invokes a macro-level program. This is controlled by a new system initialization parameter, DISMACP. The transaction is disabled if the abend is allowed to percolate (or is reissued by a HANDLE routine with the same abend code) all the way to the highest program level of the transaction. CICS system transactions (those starting with the letter C) cannot be disabled.

3. Assigns a unique transaction abend code. This allows the program error program to perform customized actions.

---

## CICS internal macros

There is no guarantee that CICS internal macros or control blocks will be source- or object-compatible between releases. There is no guarantee that the code will be shipped if it is not required for user exit programs, or for table generation. CICS will endeavor to maintain source-compatibility for internal macros known to be used by system programmers (for example, the LOCATE macros and DFHWTO). For LOCATE macros, CICS reserves the right to return the address of a copy of a control table entry, rather than the real entry in response to a LOCATE. Where copies are provided, any changes made to the copy will not be reflected in the original control block.

#

Where possible, CICS provides commands that are equivalent in function to those macros that have been discontinued. These are shown in the *CICS Application Programming Reference*.

Some of the removed macros, for example, most DFHBIF macros do not have directly equivalent command-level replacements. These macros are generally for functions that are rarely used, and which are no longer supported or which are no longer required owing to changes in the way CICS operates.

## Running CICS systems in parallel

If you run CICS regions in parallel with CICS for VSE/ESA Version 2.3, any programs you write for use on both systems must be written to use only those functions that are available in both releases.

If you write your programs according to the documented application programming interface guidelines, you should not have any difficulty migrating your programs. As you begin to work with this release, you need to spend time becoming familiar with the changes to the programming environment. For example, before you develop new programs, you should study the detailed books in the CICS library for information about new methods of coding and debugging programs. Areas of the programming environment in which there are important changes include:

- System and application problem determination methods
- Dump
- Trace
- Messages and codes

If you produce your own programming documentation, you must update it to reflect the changes made to these areas. See the *CICS Migration Guide* for more information.



---

## Chapter 12. CPI-C

The Systems Application Architecture (SAA) aims to provide a framework for designing and developing application programs that are consistent across different systems.

SAA offers the *common programming interface (CPI)* for designing application programs that can be easily integrated with other applications and can run in multiple SAA environments.

CPI Communications (CPI-C) provides an alternative API to existing CICS APPC support. CPI-C provides DTP on APPC connections and can be used in COBOL, C, PL/I, and assembler language.

The CPI-C API can be used in APPC networks that include multiple system platforms and that would benefit from the consistency of a common API.

CPI-C can converse with applications on any system that provides an APPC API, including CICS platforms. You may use EXEC CICS APPC API commands on one end of a conversation and CPI Communications commands on the other.

---

### Using CPI-C

**1** Define the following resources:

- PROFILE
- CONNECTION
- SESSIONS

**2** Define CICS partner information (*side information*). CPI-C requires this information to begin a conversation with a partner program. You can do this in one of two ways:

- Create a PARTNER definition
- In an application program, set SYMDESTNAME to a null value, issuing the appropriate CPI SET calls. See the *Common Programming Interface Communication Reference*, SC26-4399, for details.

#  
#  
#  
#  
#

**3** Use the CPI-C API to develop applications to run in APPC networks. The API is defined as a general call interface and is described in *Common Programming Interface Communication Reference*, SC26-4399.

**4** Resolve the application's calls to the CPI-C interface by link-editing it with the CICS CPI-C stub (DFHCPLC). See the *CICS System Definition Guide* for more information.

---

## Changes to CICS externals

Change	What it does	Where to find more information
Resource definition		
PARTNER resource definition	Defines CICS partner information for use with CPI-C	<i>CICS Resource Definition Guide</i>
Problem determination		
CCPI log	Holds CPI-C information and error messages	<i>CICS Problem Determination Guide</i>
CP component identifier in trace	Allows you to control tracing of CPI-C (level 1 and exception tracing)	<i>CICS Problem Determination Guide</i>
DFHCPnnnn messages	Report events associated with CPI-C	<i>VSE/ESA Messages and Codes Volume 3</i>

---

## Chapter 13. FEPI

The Front End Programming Interface (FEPI) enables you to write CICS applications that access CICS programs. In other words, it provides a front end to these programs.

At some point, you may want to change the way you use applications with CICS, for example, to extend their use by incorporating them into other applications. Without FEPI, this would mean changing the application programs themselves. In many cases, however, you cannot change the way the applications are used because you cannot change the application programs themselves. This is the case when, for example:

- The application was bought as a package, so the source code is not available.
- The source code has been lost or the program logic is so complex (or undocumented) that any changes are considered too risky.

In such circumstances, FEPI increases the flexibility of your system by:

- Allowing existing CICS application programs to be used in different ways, in different combinations, in different environments, and on different systems, **without** changing them.
- Enabling a new program to access an existing program by acting as if it were a terminal accessing the existing program.
- Letting you write programs in a way that adds new function to old programs.

FEPI applications run in a *front-end* CICS system and access applications in a *back-end* CICS system. FEPI lets them do this by simulating a terminal connected to the back-end system. The front-end and back-end can be the same CICS system.

Back-end systems are known as *targets*. The connections to them are arranged in *pools*, each of which has associated *properties*.

---

### Overview of using FEPI

Using FEPI requires careful planning. For detailed information, see the *CICS Front End Programming Interface User's Guide*.

These are the key steps you should consider:

- 1 Plan your configuration, in particular:
  - Details of the back-end applications and systems
  - Names of nodes and targets
  - Operator control requirements
  - Journaling requirements
  - Signon and signoff procedures
  - Special event handling
  - Pools required for control purposes
  - Pools required for functional reasons
  - Number of nodes
  - Setup program organization

- 2 Decide how to organize pools and their properties, the connections between nodes and targets in your system.
- 3 Ensure that sufficient storage is available on the CDSA and ECDSA. FEPI does not require VSE/ESA storage additional to that needed for a normal CICS region.
- 4 Define transient data queues in the DCT to hold information about FEPI. There are samples in DFH0IZRQ to help you.
- 5 Specify FEPI=YES system initialization parameter to make FEPI available.
- 6 If you may want to protect FEPI system programming commands, see the command security options available with your ESM. You should ensure that authorized users of CEMT are also authorized to use the FEPI commands.
- 7 Make sure that the FEPI resource definition group, DFHFEPI, is in your startup list.
- 8 Define transactions that are to be started by FEPI as CICS started tasks, with SPURGE=NO and TPURGE=NO to prevent them being accidentally canceled.
- 9 Define FEPI nodes to VTAM.
- 10 Define and manage LUs (simulated terminals) for use by FEPI back-end systems.
- 11 Write a setup program, using the FEPI system programming commands, to define FEPI nodes, targets, property sets, and pools. Define the setup program to CICS and associate it with a transaction. Use the sample setup program, DFH0xZXS to help you. See step 14 for details of the sample material.
- 12 Write a PLT program to start the setup transaction. Define the PLT program to CICS and include it in the second part of the PLTPI list.
- 13 You may also need to write:
  - A monitoring program to handle exceptional events
  - Global user exit programs
  - Specialized operator transactions to simplify control of FEPI resources
  - Any common functions not provided by individual FEPI applications.
- 14 Use the FEPI high-level programming interface or data-stream-level commands to build up applications. Define these applications to CICS just like any other application.

An integrated set of sample programs are provided to show you what FEPI can do. To customize these samples, change the customization data copy book (DFH0BZCx), the setup program (DFH0xZXS), and the resource definitions (DFH0IZRx), where the letter x indicates the following:

**Copy books** DFH0BZCx

A for the assembler version, C for C, O for COBOL, and P for PL/I.

**Setup programs** DFH0xZXS

A for the assembler version, C for C, and O for COBOL.

### Resource definitions DFH0IZRx

C for the back-end CICS region;

D for the front-end CICS region;

I for the back-end IMS™ system;

Q for the transient data queues.

## Changes to CICS externals

Change	What it does	Where to find more information
System definition		
SIT option, FEPI=YES	Controls whether FEPI is available.	<i>CICS System Definition Guide</i>
SZ option on SPCTRxx and STNTRxx system initialization parameters	Control tracing of FEPI	<i>CICS System Definition Guide</i>
Application programming		
EXEC CICS FEPI command set, including: <ul style="list-style-type: none"> <li>• ALLOCATE</li> <li>• AP NOOP</li> <li>• CONVERSE</li> <li>• EXTRACT</li> <li>• FREE</li> <li>• ISSUE</li> <li>• RECEIVE</li> <li>• REQUEST</li> <li>• START</li> <li>• SEND</li> </ul>	Allows you to write application programs to use FEPI.	<i>CICS Front End Programming Interface User's Guide</i>
DFH0xZyy set of sample programs in source form	Shows you what FEPI can do	<i>CICS Front End Programming Interface User's Guide</i>
SZ option on EXEC CICS INQUIRE and SET TRACETYPE commands	Control tracing of FEPI	<i>CICS Front End Programming Interface User's Guide</i>
FEPI translator option	Instructs the translator to process FEPI commands.	<i>CICS Front End Programming Interface User's Guide</i>
FEPIRESOURCE option added to EXEC CICS QUERY SECURITY command	To query and check command security of FEPI resources	<i>CICS Front End Programming Interface User's Guide</i>
System programming		
EXEC CICS FEPI command suite, including: <ul style="list-style-type: none"> <li>• ADD</li> <li>• DELETE</li> <li>• DISCARD</li> <li>• INQUIRE</li> <li>• INSTALL</li> <li>• SET</li> </ul>	Allows you to manage the FEPI pools, targets, connections, and properties.	<i>CICS Front End Programming Interface User's Guide</i>

<b>Change</b>	<b>What it does</b>	<b>Where to find more information</b>
Language-specific copybooks, DFHSZAPx	Contain declarations for RESP2 values on commands	<i>CICS Front End Programming Interface User's Guide</i>
CICS-supplied transactions		
Extensions to CEMT, including: <ul style="list-style-type: none"> <li>• DISCARD FENODE, FEPOOL, FEPROPSET, and FETARGET</li> <li>• INQUIRE FECONNECTION, FENODE, FEPOOL, FEPROPSET, and FETARGET</li> <li>• SET FECONNECTION, FENODE, FEPOOL, and FETARGET</li> </ul>	Allow you to manage FEPI pools, connections, properties, and targets.	<i>CICS Front End Programming Interface User's Guide</i>
SZ component identifier added to CETR	Allows you to trace FEPI	<i>CICS Front End Programming Interface User's Guide</i>
Resource definition		
DFHFEPi resource definition group	Contains the CSZI transaction for initializing the FEPI interface and sample resource definitions for front-end CICS, back-end CICS, and CICS TD queues.	<i>CICS Resource Definition Guide</i>
Problem determination		
SZ component identifier in INFO/ANA dump and CICS trace	Allows you to control tracing of FEPI (level 1 trace only)	<i>CICS Front End Programming Interface User's Guide</i>
DFHSZnnnn messages	Report events associated with FEPI	<i>CICS Front End Programming Interface User's Guide</i>
Additional data in CICS monitoring	Report events associated with FEPI	<i>CICS Front End Programming Interface User's Guide</i>
Additional information in CICS statistics	Reports on FEPI pools, connections, and targets.	<i>CICS Front End Programming Interface User's Guide</i>
Customization		
XSZARQ global user exit	Gives control to a global user exit program after execution of FEPI commands.	<i>CICS Front End Programming Interface User's Guide</i>

<b>Change</b>	<b>What it does</b>	<b>Where to find more information</b>
XSZBRQ global user exit	Gives control to a global user exit program before execution of FEPI commands.	<i>CICS Front End Programming Interface User's Guide</i>





---

## Chapter 14. EXCI

The external CICS interface (EXCI) makes it easier for you to reach a CICS application from a non-CICS environment. EXCI is an application programming interface that allows a non-CICS program (a client program) running in VSE to call a program (a server program) running in a CICS TS Release 1 region. A communications area enables the exchange of information. The CICS program is invoked as if linked by another CICS program.

The EXCI allows you to allocate and open sessions to a CICS system and to pass distributed program link (DPL) requests over them. These sessions are called *pipes*. A pipe is a one-way communication path between a sending process and a receiving process. When EXCI is used, each pipe maps onto one MRO session and carries information from the client program to the CICS server region. The CICS interregion communication program (DFHIRP) supports these DPL requests from a non-CICS client program.

A client program that uses the external CICS interface can operate multiple sessions for different users (either under the same or separate tasks) all coexisting in the same VSE address space without knowledge of, or interference from, each other. Where a client program attaches another client program, the attached program runs under its own task.

There are two programming interfaces that programs running in VSE can use to call application programs running in a CICS region: (1) the EXEC CICS LINK PROGRAM command interface, and (2) the EXCI CALL interface. See “Choosing between the EXEC CICS LINK or CALL interface” on page 78 to decide which you should use. The EXCI CALL interface allows you to:

- Allocate and open sessions to a CICS system from non-CICS programs running under VSE/ESA
- Issue distributed program link (DPL) requests on these sessions from the non-CICS programs.

The provision of this new programming interface means that, typically, a VSE batch job can:

- Update resources with integrity while CICS is accessing them.
- Take CICS resources offline (and back online) at the start (and end) of the batch job. For example, you can:
  - Open and close CICS files.
  - Enable and disable transactions in CICS (and so eliminate the need for a master terminal operator during system backup and recovery procedures).

### Requirements

Client programs running in a VSE address space can communicate only with CICS server regions running under CICS TS Release 1 or a later, upward-compatible release. This is because of the changes to the MRO connection definition to support the external CICS interface.

Also, the client program can connect to the server CICS region only through the CICS TS Release 1, or later, interregion communication program, DFHIRP.

## Choosing between the EXEC CICS LINK or CALL interface

The CICS-supplied sample EXCI programs illustrate that you can use both the CALL interface (all six commands) and the EXEC CICS LINK command in the same program, to perform separate requests. However, it is unlikely that you would want to do this in a production program.

### Using EXEC CICS LINK

For low-frequency or single DPL requests you are recommended to use the EXEC CICS LINK command.

It is easier to code, and therefore less prone to programming errors.

However, each invocation of an EXEC CICS LINK command causes the external CICS interface to perform all the functions of the CALL interface.

### Using the EXCI CALL interface

For multiple or frequent DPL requests from the same client program, you are recommended to use the EXCI CALL interface.

This is more efficient, because you need perform the Initialize\_User and Allocate\_Pipe commands once only, at or near the beginning of your program, and the Deallocate\_Pipe once on completion of all DPL activity. In between these functions, you can open and close the pipe as necessary, and while the pipe is opened, you can issue as many DPL calls as you want.

---

## Using EXCI

The following steps are a guide to using the external CICS interface:

1. EXCI programs that use the EXEC CICS LINK form of the external CICS interface must be translated before assembly or compilation, using the translator option EXCI.
2. When you link edit any program that uses the external CICS interface to pass DPL requests to a server region, include the application program stub, DFHXCSTB (supplied in the VSE/ESA sublibrary, PRD1.BASE).
3. Write EXCI applications to 31-bit addressing standards and specify AMODE(31) for your EXCI client program.
4. Include in the program source of your external, non-CICS program, the appropriate CICS-supplied copybook of RESPONSE and REASON codes.
5. On the LIBDEF statement, include those libraries that contain the CICS-supplied external interface modules needed by the job and by the client

program. The external interface modules, including the sample client program, are in the VSE/ESA sublibrary, PRD1.BASE.

6. Ensure that the size of the GETVIS available in the partition is large enough to take the size of internal trace table specified on the TRACESZE parameter in the DFHXCPT options table.
7. Use the CICS-supplied sample programs supplied in the VSE/ESA sublibrary, PRD1.BASE, to help you write your own application programs.
8. For any external CICS interface program that uses EXCI LINK or CALL API, specify OS390 as the last parameter on the // EXEC statement.
9. If you want to use the COBOL, PL/I, or C version of the EXCI client program, translate, compile and link-edit the program into a suitable library. If you use the assembler version, specify the APPLID of your target CICS server region as a parameter (PARM=) on the EXEC statement for the client program. Also include LE/VSE sublibrary, PRD2.SCEEBASE, in the library concatenation.
10. In all CICS regions that are to be accessed by one of the sample EXCI client programs:
  - a. Install the resource definition groups DFH\$EXCI and DFH\$FILA (preferably by adding them to your startup group list).
  - b. When the resource definition groups are installed, ensure that interregion communication (IRC) is open, either during CICS initialization or by using CEMT SET IRC OPEN.

## Changes to CICS externals

Change	What it does	Where to find more information
System programming		
New CONNTYPE keyword on EXEC CICS INQUIRE CONNECTION	Allows an application program to determine the type of external CICS interface connection, GENERIC or SPECIFIC (for use by multiple users or dedicated for use by a single user).	<i>CICS System Programming Reference</i>
New CVDA value (EXCI) on the EXEC CICS INQUIRE CONNECTION PROTOCOL option	Identifies that the connection uses EXCI for communication between CICS and non-CICS systems.	<i>CICS System Programming Reference</i>
Application programming		
DFHXCSTB (stub)	Enables EXCI commands to invoke the external CICS interface.	<i>CICS External CICS Interface</i>

<b>Change</b>	<b>What it does</b>	<b>Where to find more information</b>
Sample programs in PRD1.BASE: <ul style="list-style-type: none"> <li>• DFH\$AXCC (Assembler client program)</li> <li>• DFH\$AXCS (Assembler server program)</li> <li>• DFH0CXCC (COBOL client program)</li> <li>• DFH\$PXCC (PL/I client program)</li> <li>• DFH\$DXCC (C client program)</li> </ul>	Show you how to code client applications that use both the EXCI CALL interface and the EXEC CICS LINK command	<i>CICS External CICS Interface</i>
INITIALIZE_USER call	Initializes the user environment, including obtaining authority to use IRC facilities.	<i>CICS External CICS Interface</i>
ALLOCATE_PIPE call	Allocates a single session, or pipe, to a CICS region.	<i>CICS External CICS Interface</i>
OPEN_PIPE command	Causes IRC to connect an allocated pipe to a receive session of the appropriate connection defined in the CICS region named either on the Allocate_Pipe command, or in DFHXCURM.	<i>CICS External CICS Interface</i>
DPL_Request call	Issues a distributed program link request across an open pipe connected to the CICS system on which the server (or target) application program resides.	<i>CICS External CICS Interface</i>
CLOSE_PIPE call	Disconnects an open pipe from CICS. The pipe remains in an allocated state, and its tokens remain valid.	<i>CICS External CICS Interface</i>
DEALLOCATE_PIPE call	Deallocates a pipe from CICS. On completion of this command, the pipe can no longer be used, and its associated tokens are invalid.	<i>CICS External CICS Interface</i>

Change	What it does	Where to find more information
Language-specific copybooks, supplied in PRD1.BASE library: <ul style="list-style-type: none"> <li>DFHXCPLD (Assembler)</li> <li>DFHXCPLH (C)</li> <li>DFHXCPLP (COBOL)</li> <li>DFHXCPLL (PL/I)</li> </ul>	Map the storage areas for the <i>return_area</i> and <i>dpl_retarea</i> parameters of the EXCI CALL commands and provide EQUATE statements for each type of EXCI CALL.	<i>CICS External CICS Interface</i>
Language-specific copybooks, supplied in PRD1.BASE library: <ul style="list-style-type: none"> <li>DFHXCRCO (Assembler)</li> <li>DFHXCRCR (C)</li> <li>DFHXCRCO (COBOL)</li> <li>DFHXCRCR (PL/I)</li> </ul>	Contain all the possible return codes.	<i>CICS External CICS Interface</i>
External interface EXEC CICS LINK command.	Links from a VSE/ESA client program to the specified server program in a CICS region. With the exception of APPLID and RETCODE parameters, the external CICS interface parameters on EXEC CICS LINK are the same as for a CICS-CICS DPL command.	<i>CICS External CICS Interface</i>
Translator option, EXCI	Specifies that the translator is to process EXEC CICS LINK commands for the External CICS Interface (EXCI).	<i>CICS Application Programming Guide.</i>
Resource definition		
CONNTYPE attribute on the CONNECTION resource definition	Indicates whether the connection is generic or specific (on EXCI connections only)	<i>CICS Resource Definition Guide.</i>
EXCI value on the PROTOCOL attribute of the CONNECTION and SESSIONS resource definitions	Indicates that the connection is intended for use by a program using the external CICS interface.	<i>CICS Resource Definition Guide.</i>
Group DFH\$EXCI	Contains definitions for the sample server transaction, server program, EXCI connections and sessions.	<i>CICS Resource Definition Guide.</i>
Group DFH\$FILE	Contains the definition for the supplied sample VSAM file, FILEA, which is referenced by the EXCI sample programs.	<i>CICS Resource Definition Guide.</i>

Change	What it does	Where to find more information
Customization		
New user-replaceable module, DFHXCURM	Solves problems of availability in the target CICS region by enabling you to route requests to different CICS systems.	<i>CICS External CICS Interface</i>
Options table, DFHXCOPT, generated by the DFHXCOPT macro	Enables you to specify a number of parameters that are required by the external CICS interface.	<i>CICS External CICS Interface</i>
Problem determination		
New messages: <ul style="list-style-type: none"> <li>• DFHIR3799</li> <li>• DFHEX0001—3</li> <li>• DFHEX0005</li> <li>• DFHEX0010—16</li> <li>• DFH7004I and DFH7005I</li> </ul>	Report on EXCI activity.	<i>VSE/ESA Messages and Codes Volume 3.</i>
Trace entries	Written by the external trace interface to the internal trace table (if you have trace switched on).	<i>CICS Diagnosis Reference</i>
VSE SDUMPs	Dump external CICS interface control blocks, as well as trace entries.	<i>CICS Problem Determination Guide</i>
VSE abends 0401—0415	Issued during an external CICS interface job	<i>VSE/ESA Messages and Codes Volume 3</i>
New user-replaceable program, DFHXCTRA (for use under the guidance of IBM service).  It is the equivalent of DFHTRAP used in CICS.	Requests the external CICS interface to write a trace entry on its behalf; instructs the external CICS interface to take an SDUMP or to disable DFHXCTRA.  The CICS-supplied sample version of DFHXCTRA is supplied in PRD1.BASE.	<i>CICS External CICS Interface and CICS Problem Determination Guide</i>
Copybook DFHXCTRD, supplied in the VSE/ESA sublibrary, PRD1.BASE	Defines the parameter list passed to the DFHXCTRA user-replaceable program.	<i>CICS External CICS Interface</i>

---

## Chapter 15. Cancel start requests

CICS interval control is enhanced to enable you to cancel scheduled requests (automatic initiate descriptors (AIDs) on an AID chain) without having to access CICS control blocks. You can now cancel AIDs using EXEC CICS or CEMT SET TERMINAL or CONNECTION commands. There is also a new global user exit which is invoked on cancelation of an AID resulting from a start command with data, for example, EXEC CICS START TERMID FROM(*data*) or FEPI START, executed in a local system.

The general-use programming interface solution provided for this purpose ensures upward compatibility of any user application programs written to perform the cancel task.

The AIDs you can cancel could, for example, represent:

- Scheduled requests to start new transactions, and
- Allocated requests from existing transactions waiting for a named terminal facility, such as a named terminal or a named LU6.2, MRO, or LU6.1 connection.

**Note:** You *cannot* selectively cancel AIDs. You can cancel all eligible AIDs for a particular terminal or connection at the time of issuing the command, but any subsequent AIDs are not canceled.

---

### Using the cancel function

- 1 Cancel AIDs using one of the following commands:
  - CEMT SET TERMINAL or EXEC CICS SET TERMINAL
  - CEMT SET CONNECTION or EXEC CICS SET CONNECTIONwith either the CANCEL or FORCECANCEL option.
- 2 To deal with the commonly used type of schedule request resulting from an EXEC CICS START TERMID command issued by a user program, on the EXEC CICS START TERMID command, specify that data is to be associated with a schedule request, for later retrieval by the started task. When such a request is canceled, the associated temporary storage queue is deleted.
- 3 Use the XALCAID global user exit.

---

### Changes to CICS externals

Change	What it does	Where to find more information
System programming		
CANCEL option on EXEC CICS SET TERMINAL and SET CONNECTION	Cancels scheduled AIDs associated with the terminal or connection.	<i>CICS System Programming Reference</i>

<b>Change</b>	<b>What it does</b>	<b>Where to find more information</b>
FORCECANCEL option on EXEC CICS SET CONNECTION	Forcibly removes some system AIDs associated with the connection that cannot be cancelled with the CANCEL option.	<i>CICS System Programming Reference</i>
CICS-supplied transactions		
CANCEL option on CEMT SET TERMINAL and SET CONNECTION	Cancels scheduled AIDs associated with the terminal or connection.	<i>CICS-Supplied Transactions</i>
FCANCEL option on CEMT SET CONNECTION	Forcibly cancels certain system AIDs associated with the connection that cannot be cancelled with the CANCEL option.	<i>CICS-Supplied Transactions</i>
Customization		
XALCAID global user exit	Invoked when an AID with data is canceled by CEMT, by EXEC CICS SET TERMINAL or SET CONNECTION, or during reinstallation of a terminal or connection.	<i>CICS Customization Guide</i>
Problem determination		
DFHTF0100 and DFHTF0101 messages	Indicate how many AIDs have been canceled and how many remain.	<i>VSE/ESA Messages and Codes Volume 3</i>



## Chapter 16. Shared data tables

CICS file control is enhanced by the addition of shared data tables (SDTs).

Shared data tables make the optional data tables facility in CICS/VSE Version 2 obsolete and, by exploiting ESA/390 data spaces and cross-memory (XM) services, bring several important advantages:

- Greatly reduce the cost of file sharing by eliminating the overhead of function shipping for most read and browse requests within the same VSE image.
- Provide support for browse requests, and generic and GTEQ non-update reads. These enhancements can result in:
  - Improved CICS-maintained data table (CMT) performance and applicability, by making greater use of the data table rather than VSAM
  - Improved user-maintained data table (UMT) usability by expanding the available API.
- Allow users to share data between CICS regions on the same VSE image with good performance, full integrity, recovery, and availability.
- Free the FOR to process other requests if cross-memory functions are used (because these use the AOR to process the requests)
- Improve availability because it is possible for any number of files referring to the same source data set to be open at the same time to retrieve data from the same CMT.
- Result in smaller dumps and improved security because the data is stored in the data space and is not displayed in partition dumps.
- Enhance the XDTRD user exit to allow users to skip over a range of records while loading a data table.

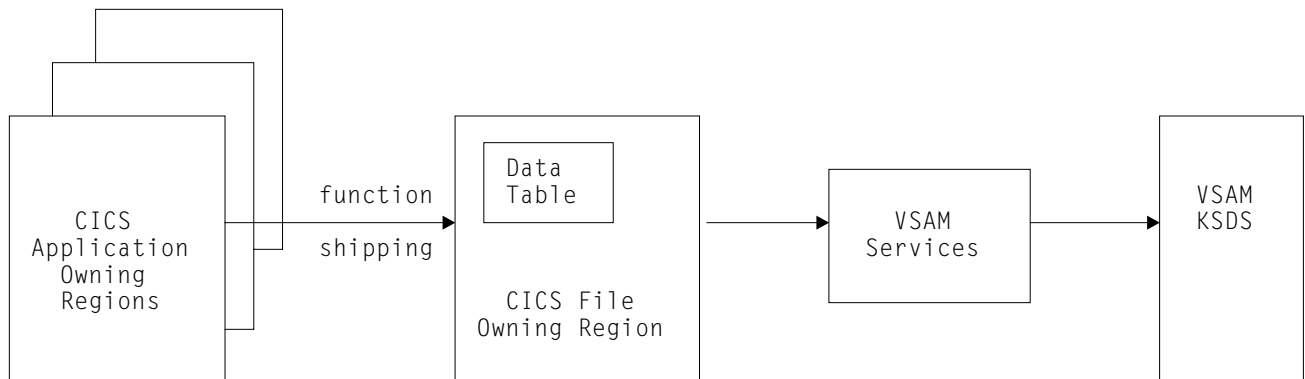


Figure 5. Data access using function shipping

Regardless of how many regions will share the data table, CICS stores records for a data table in a data space. Each CICS file-owning region has its own data space. A data space improves the integrity and isolation of data compared to the CICS address space used before. Data in an address space is available to all tasks running in that address space, whereas access to a data space can be restricted.

The data space is created when the first file to be defined as a data table is opened in the region. The data space is used by all CICS data tables that are owned by that region, and it is retained until CICS is shut down in that region.

The data space storage used by the data table is freed when the file is closed in the FOR. The storage is made available for reuse in such a way so as to protect the integrity of any AOR that was using the data table.

The AOR uses cross-memory services to execute shared data table services in an FOR to access a data table. Function shipping, with its attendant overhead, is used only for:

- Update requests
- Requests with an explicit SYSID
- Retrievals that cannot be satisfied by sharing

Figure 6 shows how data is accessed using shared data tables support.

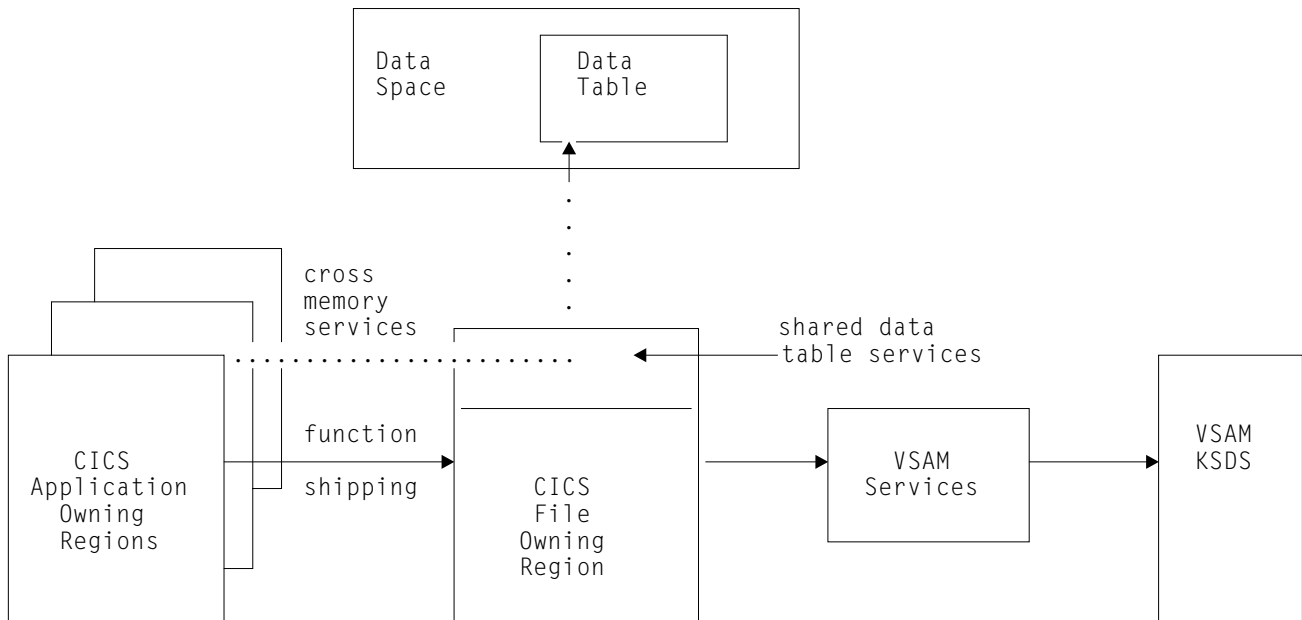


Figure 6. Data access using function shipping and shared data tables

## Using shared data tables

For detailed information about using shared data tables, see the *Shared Data Tables Guide*.

No changes are required to the file definitions for existing data tables.

- 1 Ensure that any user exit programs written for the data table global user exits, XDTRD, XDTAD, and XDTLC, are written to command-level standards.
- 2 Ensure that the DFHDTSVC, DFHCSEOT, and DFHDTSAN modules are loaded into the shared virtual area (SVA).
- 3 Define security authorization so that FORs can act as shared data table servers, and AORs can access files owned by servers, depending on the level of authority required.

- 4 You may need to increase the size of the data space available in the VSE system.

The data space for shared data tables is acquired in units of 16MB. Ensure that the following values allow for at least this amount of storage in addition to any other data space and address space requirements:

- VSIZE, specified at IPL
- DSIZE, specified in the SYSDEF DSPACE statement

---

## Changes to CICS externals

Shared data tables support completely replaces the existing data tables services introduced in CICS/VSE 2.2

Change	What it does	Where to find more information
Resource definition		
Files and data tables (CICS- and USER-maintained) can now be defined using RDO.	Provides an alternative to the DFHFCT macro definitions.	<i>CICS Resource Definition Guide</i>
Customization		
Enhancements to the XDTRD and XDTAD global user exits	Enable the exits to receive information about the source data set name and the type of data table and, for XDTRD only, to request that subsequent records up to a specified key are skipped.	<i>CICS Customization Guide</i>



---

## Part 4. Improvements to system programming

This part describes the features provided to improve system programming.

*Table 16. Road map for system programming improvements*

<b>If you want to...</b>	<b>Refer to...</b>
Allow system programmers to use all the EXEC CICS API and EXEC CICS SPI commands	Chapter 17, "Changes to global user exits in the AP domain" on page 91
Access a set of CICS services for exit programs to extend the functions of a CICS system beyond those provided as part of the standard system.	Chapter 18, "XPI" on page 101
Enable system programs to obtain information about CICS	Chapter 19, "Access to CICS state data" on page 105



---

## Chapter 17. Changes to global user exits in the AP domain

If you need to extend and customize CICS in ways not provided within the standard CICS system, you will find significant improvements to the CICS global user exit interface that make it work more efficiently, and allow you to use more CICS facilities from within user exits.

However, these improvements have made it necessary to standardize the global user exit interface. The linkage mechanism now follows the general linkage conventions for registers and parameter lists. Rewrite your global user exit programs to ensure that they conform to the new interface, and that they can run in a 31-bit addressing environment. Global user exits written to the interface available in CICS for VSE/ESA Version 2.3 will not work correctly when you migrate.

Changes have been made to the exit points available, and to the functions for which you use them. CICS also provides a new exit programming interface (XPI). For detailed information about global user exits and the new XPI, see the *CICS Customization Guide*.

### Summary of changes to global user exits

The following list summarizes the changes to the global user exit interface:

- Registers and parameter lists are standardized, and standard return code equates are provided for all global user exits. Standard names for the return codes are provided where the same return code meaning is used at more than one exit point. This makes the exit programs independent of specific values. Standards conform to VSE linkage conventions.
- Exit programs must be written to allow for the 31-bit addressing of assembler-language calls that invoke some CICS services.
- No parameters are passed in registers.
- The standard global user exit parameter list, which is mapped by the DFHUEPAR DSECT and generated by the DFHUEEXIT macro, is changed by the addition of the following new DFHUEPAR parameters:
  - Address of the stack entry (required if you use the XPI)
  - Address of the storage for the XPI parameters
  - Address of the trace flag
  - Address of the exit-specific parameters
  - Address of the caller's task indicators (data location and key)

An individual exit may not provide all of this information. For more information, see the *CICS Customization Guide*.

- The register 1 parameter list mechanism is extended to pass standard and exit-specific parameters.
- A new exit programming interface (XPI) is provided. The XPI is a set of macro function calls that can be used in global user exits to access particular CICS services.
- New global user exits are added in the restructured parts of CICS. For example, in the dispatcher, dump, message, monitor, and statistics domains. The restructuring effort has led to some parts of CICS behaving differently, meaning that some user exits can no longer be provided.

- All exits, including the new global user exits in domains, can be enabled by using an EXEC CICS ENABLE command. Transaction backout exits can also be enabled using the TBEXITS system initialization parameter. The EXITS system initialization option is withdrawn.
- A new PLT phase at initialization executes before recovery, to allow exits to be enabled. Using the DFHDELIM entry, reassemble your PLT to separate this new first phase from the second.
- The exit handling mechanism is changed. Check that your programs do not rely on any assumptions about exit-handling (for example, the internal exit numbers). The exit numbers associated with the exit identifiers in the DFHUEXIT TYPE=EXITIDS macro are not guaranteed from release to release. Rewrite any programs from previous releases that relied on such assumptions.
- The UEPCSA and UEPTCA parameters are reserved fields, and are defined as follows:

```

UEPTCA  DS  A           (reserved)
UEPCSA  DS  A           (reserved)

```

These two fields point to fetch-protected storage in all the AP domain global user exit points. Any reference to these fields in AP domain global user exit programs causes an ASRD abend at runtime. For all the other domain global user exits, the fields are set to zero values as in earlier releases.

You should re-examine your reasons for using these parameters, and where applicable choose a suitable alternative method to access the information you require.

If you have used the CSA and TCA to gain addressability to other CICS control blocks in order to obtain CICS state data, you must now use an exit-specific parameter or an XPI call.

## Summary of global user exit points

Table 17 shows the global user exits with a brief description of where or when they are invoked. Although many are new, you should still check your usage of exits already in existence as new parameters may have been added to them (such as XPI parameters).

<i>Table 17 (Page 1 of 5). Alphabetical list of global user exit points</i>			
<b>Exit name</b>	<b>Module or domain</b>	<b>Where or when invoked</b>	<b>Status</b>
XAKUSER	Activity keypoint program	Immediately before the 'end of keypoint' record is written.	New
XALCAID	Terminal allocation program	Whenever an AID with data is canceled.	New
XALTENF	Terminal allocation program	When an ATI request from transient data or interval control requires a terminal that is unknown in this system.	Changed
XDBDERR	Dynamic transaction backout program	When the DL/I backout routine detects an error.	Changed
XDBFERR	Dynamic transaction backout program	If the dynamic transaction backout program or the file control program detects an error.	Changed



Table 17 (Page 2 of 5). Alphabetical list of global user exit points

Exit name	Module or domain	Where or when invoked	Status
XDBIN	Dynamic transaction backout program	Each time CICS reads a non-DL/I dynamic log record.	Changed
XDBINIT	Dynamic transaction backout program	On entry to the dynamic transaction backout program.	Changed
XDSAWT	Dispatcher domain	After an operating system wait.	New
XDSBWT	Dispatcher domain	Before an operating system wait.	New
XDTAD	Shared data tables management program	When a write request is issued to a data table.	Changed
XDTLC	Shared data tables management program	At the completion of loading of a data table.	Changed
XDTRD	Shared data tables management program	During the loading of a data table, whenever a record is retrieved from the source data set.	Changed
XDUCLSE	Dump domain	After the domain closes a transaction dump data set.	New
XDUOUT	Dump domain	Before the domain writes a record to the transaction dump data set.	New
XDUREQ	Dump domain	Before the domain takes a system or transaction dump.	New
XEIIN	EXEC interface program	Before the execution of an EXEC CICS command.	Changed
XEIOUT	EXEC interface program	After the execution of an EXEC CICS command.	Changed
XFCAREQ	File control EXEC interface program	Before CICS processes a file control SPI request.	New
XFCAREQC	File control EXEC interface program	After CICS processes a file control SPI request.	New
XFCNREC	File control open/close program	When a mismatch is detected between the backout recovery setting for a file and its associated data set during file open processing.	New
XFCREQ	File control EXEC interface program	Before CICS processes a file control API request.	Changed
XFCREQC	File control EXEC interface program	After a file control API request has completed.	New
XFCSREQ	File control file state program	Before a file OPEN, CLOSE, ENABLE, or DISABLE command is attempted.	Changed
XFCSREQC	File control file state program	After a file OPEN, CLOSE, ENABLE, or DISABLE command has been completed.	Changed
XGMTEXT	“Good morning” message program	Before the “good morning” message is sent.	Changed
XICERREQ	Interval control EXEC interface program	Before CICS processes an interval control API request.	New

Table 17 (Page 3 of 5). Alphabetical list of global user exit points

Exit name	Module or domain	Where or when invoked	Status
XICEREQC	Interval control EXEC interface program	After an interval control API request has completed.	New
XICEXP	Interval control program	After expiry of an interval control time interval.	Changed
XICREQ	Interval control program	At the start of the interval control program, before request analysis.	Changed
XICTENF	Interval control program	When an EXEC CICS START command requires a terminal that is unknown in this system.	Changed
XISCONA	Intersystem communication program	After an attempt to allocate a session for function shipping fails because no sessions are immediately available.	New
XISLCLQ	Intersystem communication program	After: <ul style="list-style-type: none"> <li>An attempt to allocate a session for a function-shipped START NOCHECK request fails because the remote system is not in service</li> <li>A connection to the remote system cannot be established</li> <li>No sessions are immediately available and your XISCONA exit program has specified that the request is not to be queued in the issuing region.</li> </ul>	Changed
XJCWB	Journal control program	Before each physical WRITE to each journal.	New
XJCWR	Journal control program	After building a journal record in the buffer, but before writing it.	Changed
XKCREQ	Task control program	Before an ENQUEUE or DEQUEUE request.	Changed
XMEOUT	Message domain	Before a message is sent from the message domain to its destination.	New
XMNOUT	Monitoring domain	Before a record is either written to DMF or buffered before a write to DMF.	New
XPCABND	Program control program	Before a dump call is made.	Changed
XPCFTCH	Program control program	Before an application program is given control.	Changed
XPCHAIR	Program control program	Before a HANDLE ABEND routine is given control.	New
XPCREQ	Program control program	Before a LINK request is processed.	Changed
XPCREQC	Program control program	After a LINK request has been completed.	Changed

Table 17 (Page 4 of 5). Alphabetical list of global user exit points

Exit name	Module or domain	Where or when invoked	Status
XPCATA	Program control program	After an abend occurs, and before the environment is modified.	New
XRCDBER	DL/I backout program	When an error condition has been returned from the DL/I control program during backout processing, or if an error has been detected by DFHDLBP itself.	Changed
XRCINIT	DL/I backout program	At the beginning and end of the DL/I backout program.	Changed
XRCINIT	File control backout program	At the beginning and end of the file control backout program.	Changed
XRCINIT	Message and ISC state recovery program	At the beginning and end of the message and ISC state recovery program.	Changed
XRCINIT	User backout program	At the beginning and end of the user backout program.	Changed
XRCINPT	File control backout program	Whenever a non-DL/I record is read from the restart data set.	Changed
XRCINPT	Message and ISC state recovery program	Whenever a non-DL/I record is read from the restart data set.	Changed
XRCINPT	User backout program	Whenever a non-DL/I record is read from the restart data set.	Changed
XRCFCER	File control backout program	When the file control program returns an error during backout processing.	Changed
XRCOPER	File control backout program	If an error occurs while opening a file control data set.	Changed
XRMIIN	Resource manager interface program	Before execution of an EXEC SQL or RMI command.	New
XRMIOU	Resource manager interface program	After execution of an EXEC SQL or RMI command.	New
XRSINDI	Resource management modules	Immediately after a successful install or discard of a resource.	New
XSNOFF	Security manager domain	After a terminal user signs off.	New
XSNON	Security manager domain	After a terminal user signs on.	New
XSRAB	System recovery program	When the system recovery program finds a match for a VSE abend code in the SRT.	New
XSTERM	System termination program	During a normal system shutdown, immediately before TD buffers are cleared.	New
XSTOUT	Statistics domain	Before a statistics record is written to DMF.	New
XSZARQ	Front End Programming Interface	After a FEPI request has completed.	New
XSZBRQ	Front End Programming Interface	Before a FEPI request is actioned.	New

Table 17 (Page 5 of 5). Alphabetical list of global user exit points

Exit name	Module or domain	Where or when invoked	Status
XTCATT	Terminal control program	Before task attach.	Changed
XTCIN	Terminal control program	After an input event.	Changed
XTCOUT	Terminal control program	Before an output event.	Changed
XTDEREQ	Transient data EXEC interface program	Before CICS processes a transient data API request.	New
XTDEREQC	Transient data EXEC interface program	After a transient data API request has completed.	New
XTDIN	Transient data program	After receiving data from SAM (extrapartition) or VSAM (intrapartition).	Changed
XTDOUT	Transient data program	Before passing data to SAM (extrapartition) or VSAM (intrapartition).	Changed
XTDREQ	Transient data program	Before request analysis.	Changed
XTSEREQ	Temporary storage EXEC interface program	Before CICS processes a temporary storage API request.	New
XTSEREQC	Temporary storage EXEC interface program	After a temporary storage API request has completed.	New
XTSIN	Temporary storage control program	After an input event.	Changed
XTSOUT	Temporary storage control program	Before an output event.	Changed
XTSREQ	Temporary storage control program	Before request analysis.	Changed
XXMATT	Transaction manager domain	When a user transaction is attached.	New
XXRSTAT	XRF request processing program	After a VTAM failure or a predatory takeover.	Changed
XZCATT	VTAM terminal management program	Before task attach.	Changed
XZCIN	VTAM working set module	After an input event.	Changed
XZCOUT	VTAM working set module	Before an output event.	Changed
XZCOUT1	VTAM working set module	Before a message is broken into RUs.	Changed
XZIQUE	VTAM working set module	<ol style="list-style-type: none"> <li>1. When an allocate request for a session is about to be queued.</li> <li>2. When an allocate request succeeds following previous suppression of queuing.</li> </ol>	New

See the *CICS Customization Guide* for detailed information about these global user exits.

## Obsolete exits

The following exit points are obsolete:

Exit	Location
XFCIN	Located in the file control module, DFHFCP, and invoked just before the input event.
XFCINC	Located in the file control module, DFHFCP, and invoked by modules DFHFCVS and DFHFCBD after the input event.
XFCOUT	Located in the file control module, DFHFCP, and invoked before the output event.
XKCDISP	Located in the task control module DFHKCP, and invoked before task dispatch.
XSCREQ	Located in the storage control program DFHSCP, and invoked before request analysis.
XTCRDAT	Located in the terminal control program DFHTCP, and invoked after 2741 read attention.
XKCBWT and XKCAWT	Replaced by the new exits XDSBWT and XDSAWT respectively.
The task-related user exit point located in the monitoring program (DFHCMP)	The restructure of CICS monitoring means that it is replaced by the global user exit point, XMNOUT, in the monitoring (MN) domain. If you use a monitoring task-related user exit on your present CICS system, you must rewrite it to conform to the global user exit interface.

## Reassembling global user exit programs

The CICS global user exit programming interface is product sensitive, and is dependent on the implementation of CICS code from release to release. All global user exit programs must be reassembled, using the CICS TS Release 1 libraries, after you have modified them for any changes to parameters.

Note the changes summarized in this chapter (and described in greater details in the *CICS Customization Guide*) and modify your global user exit programs accordingly. Note particularly that:

- The standard global user exit parameter list is changed for all global user exits.
- The exit-specific parameter lists are new for all global user exit points.

---

## Task-related user exits

The CICS task-related user exit programming interface (also known as the resource manager interface or RMI) is used primarily by CICS application programs to access external resource managers. It:

- Enables full recursion, which means that you can issue a resource manager request from within a task-related user exit that is already processing another request for the same or a different resource manager. For example, you can issue an SQL(\*) request from within a DB2® for VSE task-related user exit while it is processing another SQL request.

- Introduces a new option, LINKEDITMODE, on the EXEC CICS ENABLE command, to invoke a task-related user exit in its link-edit AMODE instead of having to invoke it in the same AMODE as its original caller. If you do not specify the LINKEDITMODE option, your task-related user exits are invoked in the same AMODE as their original callers. A task-related user exit link-edited as AMODE(31),RMODE(ANY) and enabled with the LINKEDITMODE option of the EXEC CICS ENABLE command is placed with its associated control blocks, parameter lists, and work areas above 16MB.
- Passes parameters to task-related user exits or to application programs using the new system initialization parameter INITPARM. For example, you can use INITPARM to pass parameters to a PLTPI program to be executed in the final stage of CICS initialization; the parameters are retrieved using an EXEC CICS ASSIGN INITPARM command.
- Supports command-level API and SPI commands.
- Denies direct access to control blocks by task-related user exit programs.
- Enables a task-related user exit program to call itself recursively with DFHRMCAL.

The UEPCSA and UEPTCA parameters are reserved fields. As for global user exits, these two fields point to fetch-protected storage. Any reference to these fields in a task-related user exit results in an ASRD abend at run time.

Re-examine your reasons for using these parameters, and where applicable choose a suitable alternative method to access the information you require.

If your task-related user exit programs use the CSA and the TCA to gain addressability to other CICS control blocks in order to obtain state data, change them to use the CICS API or SPI command-level interface instead.

---

## Changes to user-replaceable programs

All user-replaceable programs:

- Must be written to command-level standards.
- Must not access internal CICS control blocks. Command-level API and SPI commands can be used.
- Can be written in any language supported by CICS, except for the user-replaceable journaling programs, DFHXJCC and DFHXJCO—these must be written in assembler language.
- Must be modified to access the information that CICS passes to them. User-replaceable programs are now passed a communication area (COMMAREA) containing the information they require; in previous releases they were given access to control block fields.
- Must be link-edited as AMODE(31) (including any programs they call).

Reassemble all user-customized error programs (NEPs, TEPs, and PEPs) before they can be migrated.

Table 19 summarizes the changes made to user-replaceable programs.

<i>Table 19. Changes made to user-replaceable programs</i>			
<b>URM</b>	<b>Purpose</b>	<b>Status</b>	<b>Notes</b>
DFHACEE	Cross system identification program	Obsolete	Use an external security manager (ESM)
DFHDYP	Dynamic routing program	Review	*
DFHPEP	Program error program	Review	*
DFHPGADX	Program autoinstall program	New	*
DFHRTY	Transaction restart program	Obsolete	Use DFHREST instead
DFHTEP	Terminal error program	Changed	Sequential devices only
DFHUAKP	Activity keypoint program	Obsolete	Replaced by global user exit
DFHXJCC and DFHXJCO	Journal open and close programs	Review	Assembler language only
DFHXSE	External security interface program	Obsolete	Use an ESM
DFHXSP	External security interface program	Obsolete	Use an ESM
DFHZATDX	Terminal autoinstall program	Review	*
DFHZATDY	APPC connections autoinstall program	New	*
DFHZNEP	Node error program	Review	*

**Note:** “Review” in the **Status** column means that the user-replaceable program is essentially unchanged but should be checked for macro usage, access to CICS control blocks, and so on.





## Chapter 18. XPI

### The exit programming interface (XPI)

A new exit programming interface (XPI) is introduced to enable you to use CICS domain services from global user exit programs.

The CICS XPI gives you access to a set of CICS services for exit programs; the interface consists of a set of calls to CICS domains listed in Table 20 on page 101. The XPI is different from CICS macro- and command-level programming because it does not need the usual CICS transaction environment, although it is dependent on a special exit programming environment. To set up this special programming environment, include the CICS-supplied exit macro, DFHUEXIT, at the beginning of your exit program.

The XPI can be used by assembler-language global user exit programs, which must be in 31-bit addressing mode when making XPI calls. They can reside above 16MB.

Because XPI calls use the services of the new CICS domains, there are limits to their use in global user exit programs invoked from domains. Global user exit programs must observe the restrictions described in the *CICS Customization Guide*. Otherwise, exit programs may behave unpredictably and this could in turn affect the operation of CICS. In particular, when using a domain service in an exit from the same domain, follow the published guidance about preventing recursion.

Table 20 lists the CICS XPI calls to servicing domains.

Domain	Call	Function	Description
Dispatcher	DFHDSSRX	ADD_SUSPEND	Obtains a suspend token for a task
Dispatcher	DFHDSSRX	DELETE_SUSPEND	Releases a suspend token associated with a task
Dispatcher	DFHDSSRX	SUSPEND	Suspends execution of the issuing task
Dispatcher	DFHDSSRX	RESUME	Resumes execution of a suspended task
Dispatcher	DFHDSSRX	WAIT_EXTERNAL	Requests a wait on one or more VSE ECBs
Dispatcher	DFHDSATX	CHANGE_PRIORITY	Changes the priority of an issuing task
Dump	DFHDUDUX	SYSTEM_DUMP	Requests a system dump
Dump	DFHDUDUX	TRANSACTION_DUMP <b>1</b>	Requests a transaction dump
Kernel	DFHKEDSX	START_PURGE_PROTECTION	Inhibits purge for current task
Kernel	DFHKEDSX	STOP_PURGE_PROTECTION	Reenables purge for current task
Loader	DFHLDLDX	DEFINE_PROGRAM	Defines new program to LD domain
Loader	DFHLDLDX	ACQUIRE_PROGRAM	Loads a program, or, if it is already loaded, obtain its load and entry point addresses
Loader	DFHLDLDX	RELEASE_PROGRAM	Releases storage occupied by a program, or decrement its use count by one

Table 20 (Page 2 of 3). CICS XPI calls to servicing domains

Domain	Call	Function	Description
Loader	DFHLDLDX	DELETE_PROGRAM	Deletes a program definition from the list of current programs
Monitoring	DFHMNMNX	INQUIRE_MONITORING_DATA <b>1</b>	Processes a user event monitoring point (EMP)
Monitoring	DFHMNMNX	MONITOR <b>1</b>	Retrieves the current monitoring data for the issuing task
Program manager	DFHPGISX	INQUIRE_PROGRAM	Retrieves information about the attributes of a specified program
Program manager	DFHPGISX	INQUIRE_CURRENT_PROGRAM	Retrieves information about the attributes of the program that is currently running
Program manager	DFHPGISX	SET_PROGRAM	Sets selected attributes in the definition of a specified program
Program manager	DFHPGISX	START_BROWSE_PROGRAM	Retrieves a token that enables users to begin browsing through program definitions, optionally starting at the definition of a specified program
Program manager	DFHPGISX	GET_NEXT_PROGRAM	Retrieves the next program definition during a browse sequence initiated by the START_BROWSE_PROGRAM call
Program manager	DFHPGISX	END_BROWSE_PROGRAM	Ends a browse of program definitions initiated by the START_BROWSE_PROGRAM call
Program manager	DFHPGAQX	INQUIRE_AUTOINSTALL	Retrieves the settings of the autoinstall function for programs, map sets, and partition sets
Program manager	DFHPGAQX	SET_AUTOINSTALL	Changes the settings of the autoinstall function for programs, map sets, and partition sets
State data access	DFHAPIQX	INQ_APPLICATION_DATA	Retrieves application system data in the AP domain
State data access	DFHSAIQX	INQUIRE_SYSTEM	Retrieves CICS system data in the AP domain
State data access	DFHSAIQX	SET_SYSTEM	Sets CICS system data values in the AP domain
Storage manager	DFHSMMCX	GETMAIN	Obtains and initializes storage
Storage manager	DFHSMMCX	FREEMAIN	Releases storage acquired using a GETMAIN call
Storage manager	DFHSMSRX	INQUIRE_ACCESS	Retrieves the access key of an element of storage specified by start address and length
Storage manager	DFHSMMCX	INQUIRE_ELEMENT_LENGTH	Retrieves the start address and length of an element in task lifetime storage
Storage manager	DFHSMSRX	INQUIRE_SHORT_ON_STORAGE	Discovers whether CICS is short on storage
Storage manager	DFHSMMCX	INQUIRE_TASK_STORAGE	Inquire about a task's lifetime storage
Trace	DFHTRPTX	TRACE_PUT	Writes a trace entry to the active trace destinations
Transaction manager	DFHXMSRX	INQUIRE_DTRTRAN	Retrieves the name of the dynamic transaction routing transaction definition
Transaction manager	DFHXMSRX	INQUIRE_MXT	Retrieves the current value of the MXT system initialization parameter

<i>Table 20 (Page 3 of 3). CICS XPI calls to servicing domains</i>			
<b>Domain</b>	<b>Call</b>	<b>Function</b>	<b>Description</b>
Transaction manager	DFHXMCLX	INQUIRE_TCLASS	Retrieves data about a specified transaction class
Transaction manager	DFHXMIDX	INQUIRE_TRANDEF	Retrieves data about a specified transaction definition
Transaction manager	DFHXMIQX	INQUIRE_TRANSACTION	Retrieves data about an attached transaction definition
Transaction manager	DFHXMIQX	SET_TRANSACTION	Changes the task priority and transaction class of the current task
User journaling	DFHJCJCX	WRITE_JOURNAL_DATA <b>1</b>	Writes a record to a CICS journal

**Notes for Table 20 on page 101:**

**1** There is a restriction on using the XPI early during initialization. The XPI INQUIRE\_MONITOR\_DATA, MONITOR, TRANSACTION\_DUMP, and WRITE\_JOURNAL\_DATA functions should not be used in exit programs enabled during the first phase of the PLTPI. Information about the PLTPI is given in the *CICS Customization Guide*.



---

## Chapter 19. Access to CICS state data

CICS enables you to access CICS state data without referring directly to CICS control blocks. Access to CICS data through control blocks has been inhibited by removing macro support and access to particular control blocks.

Access to CICS state data is provided through enhancements to the system programming interface (SPI) and the exit programming interface (XPI). These enhancements:

- Enable CICS system programmers to obtain information about CICS through supported programming interfaces.
- Remove dependence on, and knowledge of, CICS control block structures.
- Provide a release-independent solution to enable CICS programs that access state data to work, unmodified, from release to release by means of the SPI enhancements.

---

### Using CICS state data

Review your programs to find those that use direct access to CICS control blocks, and replace these accesses with the equivalent SPI or XPI commands, as appropriate.

---

### Changes to CICS externals

Some of the changes to CICS externals designed to provide access to CICS state data were made in CICS/VSE 2.3 in preparation for the withdrawal of access to CICS control blocks. These changes were the addition of:

- A number of new EXEC CICS INQUIRE and SET commands
- A number of new options on existing INQUIRE and SET commands.

See the *CICS/VSE 2.3 Release Guide* for details of these earlier changes.

Changes made in CICS TS Release 1 are shown in Table 21.

<b>Change</b>	<b>What it does</b>	<b>Where to find more information</b>
System programming		
New command: EXEC CICS INQUIRE AUTINSTMODEL	Finds out whether a terminal autoinstall model is installed.	<i>CICS System Programming Reference</i>
New command: EXEC CICS INQUIRE DELETSHIPED	Retrieves information about system settings that control the CICS timeout delete mechanism.  Also available as a SET command.	<i>CICS System Programming Reference</i>

Table 21 (Page 2 of 3). Changes to programming interfaces for access to CICS state data

Change	What it does	Where to find more information
New command: EXEC CICS INQUIRE DSNAME	Retrieves information about an external data set.  Also available as a SET command.	<i>CICS System Programming Reference</i>
New command: EXEC CICS INQUIRE EXITPROGRAM	Returns information about programs enabled at global and task-related user exits	<i>CICS System Programming Reference</i>
<p>New options on EXEC CICS INQUIRE SYSTEM:</p> <ul style="list-style-type: none"> <li>• CDSASIZE</li> <li>• CICSTSLEVEL</li> <li>• CMDPROTECT</li> <li>• DFLTUSER</li> <li>• DSALIMIT</li> <li>• DUMPING</li> <li>• ECDSASIZE</li> <li>• EDSALIMIT</li> <li>• ERDSASIZE</li> <li>• ESDSASIZE</li> <li>• EUDSASIZE</li> <li>• PROGAUTOCTLG</li> <li>• PROGAUTOEXIT</li> <li>• PROGAUTOINST</li> <li>• PRTYAGING</li> <li>• RDSASIZE</li> <li>• REENTPROTECT</li> <li>• SDSASIZE</li> <li>• STOREPROTECT</li> <li>• UDSASIZE</li> </ul> <p>A new CVDA, CANCELED is added to the SHUTSTATUS option.</p> <p>New CVDA's of SOS and SOSABOVE are added to the SOSSTATUS option.</p> <p>The AMAXTASKS, CUSHION, and STALL options are obsolete.</p>	Retrieves CICS system information.	<i>CICS System Programming Reference</i>
New command: EXEC CICS INQUIRE PARTNER	Retrieves information about a partner.	<i>CICS System Programming Reference</i>
New command: EXEC CICS INQUIRE PROFILE	Determines whether a transaction profile is installed.	<i>CICS System Programming Reference</i>
New command: EXEC CICS INQUIRE STATISTICS	Retrieves information about CICS statistics settings.  Also available as a SET command.	<i>CICS System Programming Reference</i>

Table 21 (Page 3 of 3). Changes to programming interfaces for access to CICS state data

<b>Change</b>	<b>What it does</b>	<b>Where to find more information</b>
New command: EXEC CICS INQUIRE STORAGE	Retrieves information about task storage.	<i>CICS System Programming Reference</i>
New command: EXEC CICS INQUIRE SYSDUMPCODE	Retrieves information about a system dump table entry.  Also available as a SET command.	<i>CICS System Programming Reference</i>
New command: EXEC CICS INQUIRE TRACEFLAG	Retrieves information about trace flags.  Also available as a SET command.	<i>CICS System Programming Reference</i>
New command: EXEC CICS INQUIRE TRACETYPE	Retrieves information about CICS system tracing.  Also available as a SET command.	<i>CICS System Programming Reference</i>
New command: EXEC CICS INQUIRE TRANCLASS	Retrieves information about a transaction class.  Also available as a SET command.	<i>CICS System Programming Reference</i>
New command: EXEC CICS INQUIRE TRANDUMPCODE	Retrieves information about a transaction dump table entry.  Also available as a SET command.	<i>CICS System Programming Reference</i>
New command: EXEC CICS INQUIRE TSQUEUE	Retrieves information about a temporary storage queue.	<i>CICS System Programming Reference</i>
<b>Customization</b>		
XPI INQUIRE_SYSTEM function	Gives you access to the CICS system data in the AP domain.	<i>CICS Customization Guide</i>
XPI SET_SYSTEM function	Sets CICS system data in the AP domain	<i>CICS Customization Guide</i>
XPI INQ_APPLICATION_DATA function	Gives you access to application system data in the AP domain	<i>CICS Customization Guide</i>





## Part 5. Improved system management

This part describes the features provided to improve system management in CICS.

If you want to...	Refer to...
Allow a dynamic transaction routing program to maintain better information about the state of routed transactions.	Chapter 20, "Improved dynamic transaction routing" on page 111
Dynamically define programs, mapsets, and partition sets on their first usage.	Chapter 21, "Autoinstall for programs, mapsets, and partitionsets" on page 113
Define which CICS journals should be automatically archived by CICS when they are closed for output.	Chapter 22, "Automatic journal archiving" on page 117
Find out about improved CICS monitoring	Chapter 23, "Improved monitoring facilities" on page 121
Find out about improved CICS statistics	Chapter 24, "Improved collection of statistics" on page 129
Find out about improvements to resource definition online (RDO) for files.	Chapter 25, "Improved RDO" on page 135.
Use the DFH\$IWUP utility program to resolve resources that may be in-doubt in the time between a syncpoint and a failure in the CICS region.	Chapter 26, "Resolving in-doubt resources" on page 141.
Find out about miscellaneous changes to CICS, including: <ul style="list-style-type: none"> <li>• Dynamic management of DSAs</li> <li>• Recovery of VSAM files</li> <li>• Recovery from DTB failures</li> <li>• Support for forward recovery utilities</li> <li>• Improvements to MRO and to transaction management.</li> </ul>	Chapter 27, "Miscellaneous changes to CICS" on page 143.
Find out which functions for system definition and resource definition are changed or obsolete.	Chapter 28, "Changes for system and resource definition" on page 147
Find out about how you can add new resource definitions directly to a running CICS region using the EXEC CICS CREATE command	Chapter 29, "Creating and installing resources using EXEC CICS CREATE" on page 153



---

## Chapter 20. Improved dynamic transaction routing

The CICS dynamic transaction routing(DTR) mechanism is enhanced to help dynamic transaction routing programs to keep track of routed transactions. The new dynamic transaction routing mechanism:

- Enables a DTR program to make more intelligent routing decisions, for example, based on workload goals
- Makes it possible to detect inter-transaction affinities as they are created.

The CICS relay program, DFHCRP, is replaced by a new relay program, DFHAPRT. The new relay program invokes the user-replaceable DTR program at several new points, as follows:

- Before routing a remote, terminal-oriented, transaction that is initiated by automatic transaction initiation (ATI)
- At the end of a transaction that DFHAPRT has routed locally, if the initial invocation requests re-invocation at termination
- If a routed transaction abends, to indicate that an abend occurred before returning a response to the CICS transaction manager

DFHCRP is moved to the DFHCOMP1 compatibility resource group.

The DFHAPRT program passes information to the DTR program by means of a communication area (COMMAREA). The COMMAREA has some new function codes and new fields, as described in the *CICS Migration Guide*

Consider the additional advantages from using CICSplex System Manager (CICSplex® SM). For information about CICSplex SM, see the *CICSplex SM: Concepts and Planning* manual.

---

### Using DTR enhancements

- 1 On the DTRTRAN system initialization parameter, specify a single, common definition for all remote transactions that are to be dynamically routed. This is the recommended method.

Alternatively, you can use the REMOTESYSTEM option of the transaction resource definition to specify the system name of a default CICS region to which the transaction is to be routed. If you do not specify a REMOTESYSTEM name on the transaction definition, the system name of the local CICS region is passed to the dynamic transaction routing program.

- 2 Ensure that none of your application programs use EXEC CICS LINK to link to the CICS relay program, DFHCRP.
- 3 If you want to share the CSD with a CICS for VSE/ESA Version 2.3 system, and will be using the relay program from that version, the definition is provided in the DFHCOMP1 compatibility group.
- 4 Either customize the sample program, DFHDYP, or replace it with your own routing program.

To use EDF to test your routing program, rename the program to something that does not start with the letters “DFH”.

## Changes to CICS externals

Change	What it does	Where to find more information
System definition		
DTRTRAN system initialization parameter	Names the transaction definition required for dynamic transaction routing	<i>CICS System Definition Guide</i>
CICS-supplied transactions		
CTRX	Defines a dynamic transaction routing definition.	<i>CICS-Supplied Transactions</i>
Problem determination		
New trace entry and exit points	Causes the values of the parameters related to the DFHAPRT gate in the AP domain to be recorded (level 1 trace)	<i>CICS Problem Determination Guide</i>
Minor changes toabend codes ACRA—ACRH		<i>VSE/ESA Messages and Codes Volume 3</i>
New ACRMabend code		<i>VSE/ESA Messages and Codes Volume 3</i>

---

## Chapter 21. Autoinstall for programs, mapsets, and partitionsets

Before CICS provided support for autoinstall of resources, they had to be defined individually to CICS before they could be used. The autoinstall function is extended to defines user programs, mapsets, and partitionsets dynamically on the first usage.

*Program autoinstall* (autoinstall for programs, mapsets, and partitionsets) brings several benefits:

- **Reduced system administration costs**

You do not have to define resources to CICS before they can be used. Moreover, you do not have to maintain these definitions.

- **Saving in virtual storage within the CICS address space**

The definitions of autoinstalled resources do not occupy table space until they are generated.

- **Faster restart**

Cold starts are faster with program autoinstall because you can omit all predefined program definitions from startup group lists.

For warm and emergency restarts, the speed depends on whether you use program autoinstall with or without cataloging. (See the *CICS Recovery and Restart Guide* for information on cataloging.)

- With cataloging, restart times are similar to those of restarting a CICS region that is not using program autoinstall. This is because, in both cases, resource definitions are reinstalled from the catalog during the restart. The definitions after the restart are those that existed before the system was terminated.
- Without cataloging, CICS restart times are improved because CICS does **not** install definitions from the CICS global catalog. Instead, definitions are autoinstalled as required whenever programs, mapsets, and partitionsets are referenced following the restart.

- **System autoinstall**

The transaction list table (XLT), initialization and shutdown program list tables (PLTPI and PLTSD), and the programs defined within these tables are all autoinstalled by CICS *system autoinstall*. System autoinstall does not require model definitions nor the support of the user-replaceable autoinstall module.

### Requirements

- A customized version of the autoinstall user-replaceable module DFHPGADX.
- The CSPL transient data queue, if you want to log messages associated with autoinstall for programs. See the *CICS Resource Definition Guide* for details on how to define queues.

If the autoinstall program function is enabled, and an implicit or explicit load request is issued for a previously undefined program, mapset, or partitionset, CICS

dynamically creates a definition, and installs and catalogs it, as appropriate. An implicit or explicit load occurs when:

- CICS starts a transaction.
- An application program issues one of the following commands:
  - EXEC CICS ENABLE (for a global user exit or task-related user exit program)
  - EXEC CICS LINK
  - EXEC CICS LOAD
  - EXEC CICS RECEIVE or SEND MAP
  - EXEC CICS RECEIVE PARTN
  - EXEC CICS SEND PARTNSET
  - EXEC CICS XCTL
- After an EXEC CICS HANDLE ABEND PROGRAM(...) command has been issued, a condition is raised and CICS transfers control to the named program.
- CICS calls a user-replaceable module for the first time.

## Model definitions and a user-replaceable module

Like the terminal autoinstall facility, the program autoinstall facility uses model definitions, together with a user-replaceable module, to create explicit definitions—in this case for programs, mapsets, and partitionsets that need to be autoinstalled. CICS calls the user-replaceable module, with a parameter list that gives the name of the appropriate model definition. On return from the user-replaceable module (depending on the return code) CICS creates a resource definition from information in the model and parameters returned by the user-replaceable module.

Note that CICS does not call the user-replaceable module for any CICS programs, mapsets, or partitionsets—that is, any objects that begin with the letters DFH.

## Autoinstall processing of mapsets

Mapsets are processed differently depending on whether autoinstall is active in the CICS region.

### Autoinstall is not active

CSD definition is required. CICS attempts to load a referenced mapset with a suffix. If this fails, CICS tries an unsuffixed version. If that is unsuccessful, abend APCT is issued.

### Autoinstall is active

Using autoinstall, CICS attempts to load the referenced suffixed mapset or partitionset, then the unsuffixed one. Each time, if the resource is not defined, a definition is autoinstalled. The transaction requesting the resource abends only if no version of the resource exists in the library, either suffixed or unsuffixed.

---

## Using autoinstall for programs, mapsets, and partitionsets

- 1 Decide what to do with existing program and mapset definitions in the CSD—either delete them and replace with the autoinstall mechanism or leave them as explicit definitions.
- 2 Decide whether you want to use program autoinstall with or without cataloguing.
- 3 If want to have autoinstalled definitions recorded in the CICS catalog, specify this using the PGAICTLG system initialization parameter, or using the CEMT (or EXEC CICS) SET SYSTEM PROGAUTOCTLG command.
- 4 Customize the autoinstall user-replaceable module, DFHPGADX, if necessary.
- 5 Define the CSPL transient data queue to log messages associated with program autoinstall.
- 6 Include the CSD group DFHPGAIP, containing the default resource definitions for autoinstall, in the startup list.
- 7 Make sure that there is a program resource definition in the CSD for your autoinstall program and for any program it references. Include the group containing these definitions in your startup list.

Set your autoinstall program with the PGAIEXIT system initialization parameter or the CEMT or SPI SET SYSTEM PROGAUTOEXIT command.

- 8 Activate program autoinstall with the PGAIPGM system initialization parameter or the CEMT or SPI SET SYSTEM PROGAUTOEXIT command.

---

## Changes to CICS externals

Change	What it does	Where to find more information
System definition		
PGAICTLG system initialization parameter	Specifies whether autoinstalled program definitions are to be catalogued.	<i>CICS System Definition Guide</i>
PGAEXIT system initialization parameter	Specifies the name of the autoinstall exit program	<i>CICS System Definition Guide</i>
PGAIPGM system initialization parameter	Specifies the state of the program autoinstall function at initialization.	<i>CICS System Definition Guide</i>
Resource definition		
DFHPGAPG default model program definition	Defines programs for autoinstall	<i>CICS Resource Definition Guide</i>
DFHPGAMP default model mapset definition	Defines mapsets for autoinstall	<i>CICS Resource Definition Guide</i>
DFHPGAPT default model partition set definition	Defines partitionsets for autoinstall	<i>CICS Resource Definition Guide</i>

<b>Change</b>	<b>What it does</b>	<b>Where to find more information</b>
LANGUAGE option on program definition can now be omitted	CICS determines the language when the program is loaded.	<i>CICS Resource Definition Guide</i>
CICS-supplied transactions		
PROGAUTOCTLG option on CEMT INQUIRE or SET SYSTEM	Displays or specifies whether autoinstalled programs are to be catalogued	<i>CICS-Supplied Transactions</i>
PROGAUTOEXIT option on CEMT INQUIRE or SET SYSTEM	Displays or specifies the name of the user-provided program called by the program autoinstall code to select or modify a model definition.	<i>CICS-Supplied Transactions</i>
PROGAUTOInst option on CEMT INQUIRE or SET SYSTEM	Displays or specifies whether autoinstall for programs is active or inactive.	<i>CICS-Supplied Transactions</i>
System programming		
PROGAUTOCTLG option on EXEC CICS INQUIRE or SET SYSTEM	Specifies whether autoinstalled programs are to be catalogued	<i>CICS System Programming Reference</i>
PROGAUTOEXIT option on EXEC CICS INQUIRE or SET SYSTEM	Specifies the name of the user-provided program called by the program autoinstall code to select or modify a model definition.	<i>CICS System Programming Reference</i>
PROGAUTOINST option on EXEC CICS INQUIRE or SET SYSTEM	Specifies whether autoinstall for programs is active or inactive.	<i>CICS System Programming Reference</i>
Customization		
New user-replaceable module. Language-specific versions are in the VSE/ESA sublibrary PRD1.BASE. <ul style="list-style-type: none"><li>• DFHPGADX (Assembler)</li><li>• DFHPGODX (COBOL)</li><li>• DFHPGHDX (C)</li><li>• DFHPGLDX (PL/I)</li></ul>	Provides CICS with the information it needs, such as the autoinstall model name, to complete an autoinstall request.	<i>CICS Customization Guide</i>
Language-specific copybooks: <ul style="list-style-type: none"><li>• DFHPGAD (Assembler)</li><li>• DFHPGCO (COBOL)</li><li>• DFHPGCH (C)</li><li>• DFHPGCL (PL/I)</li></ul>	Contain the parameter lists—information you need to tailor the user-replaceable module.	<i>CICS Customization Guide</i>



---

## Chapter 22. Automatic journal archiving

CICS automatic journal archiving allows you to define which CICS journals are automatically archived by CICS when they are closed for output.

Automatic journal archiving brings the following benefits:

- **Prompt submission of archive jobs**

Under normal conditions, this eliminates the need for CICS tasks to wait for a journal archive job to complete, and therefore removes the need for you to code the Jouropt=PAUSE operand on the DFHJCT TYPE=ENTRY macro to prevent the overwriting of journals.

- **Eliminates operator intervention to archive journals**

Unless the archive is to tape, the operator needs to intervene only if an automatic journal archiving job fails and results in a journal data set that is not ready to use when CICS tries to open it for output.

- **Provides greater security**

CICS does not reuse a journal data set until the archive job is complete so any journal data that might be required for recovery is preserved.

- **Removes the need for your own archiving facility**

You can choose to use either the DFHXJCO and DFHXJCC user-replaceable modules or automatic journal archiving to submit archive jobs.

If you decide to use automatic journal archiving, you need to define DFHJACD, the journal archive control data set.

For more information about this new data set and about the new automatic archiving facility, see the *CICS System Definition Guide*.

Whenever a journal eligible for automatic journal archiving is closed for output, the journal archiving program (DFHJAP) is called automatically to create an archiving job, from skeletal JCL stored in a user-defined VSE/ESA sublibrary.

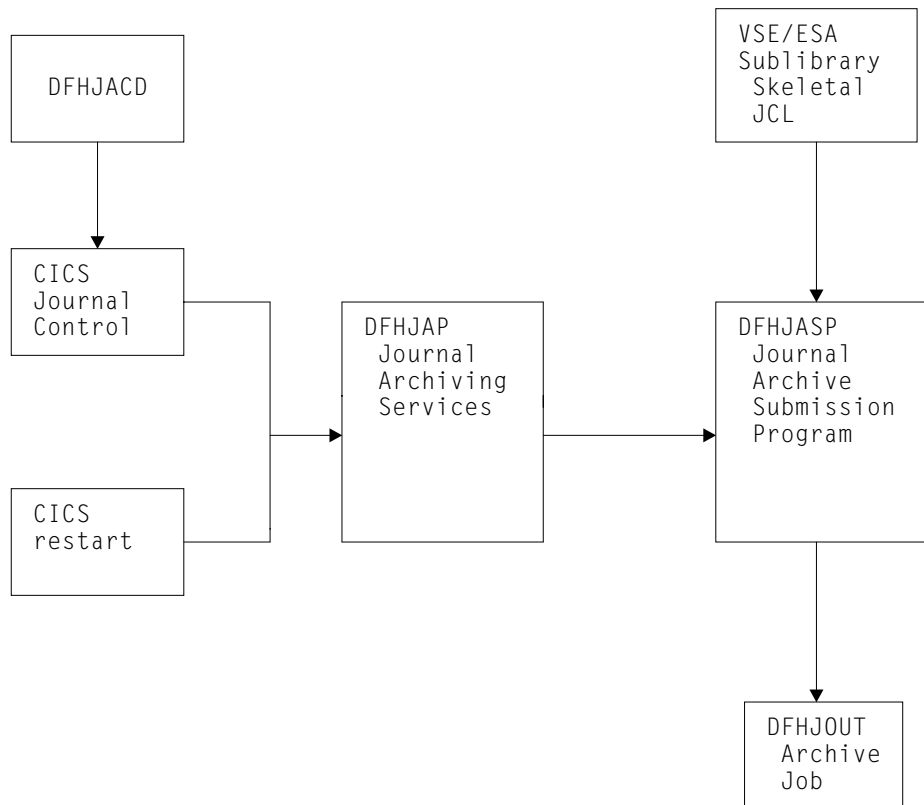


Figure 7. CICS automatic journal archiving—schematic representation

Before CICS attempts to open a journal for output, it calls DFHJAP to check whether the journal has been archived successfully.

The submission of archiving jobs and the reuse of journal data sets is controlled by the journal archiving control data set (JACD). The JACD contains the current status of journal data sets. The JACD is a VSAM RRDS data set which contains one record slot for every possible journal data set. The maximum is 198; that is 99 journals defined as DISK2 (for data sets A and B). Reference is always made to the JACD before journals are reused.

An automatic journaling utility program (DFHJACDU) is supplied so that you can:

- List the current status of all journal data sets.
- Check the status of a given journal for a status of NOTREADY. To be invoked by archive job before archive submission to ensure the job can proceed.
- Update the status of a given journal from NOTREADY to READY. To be invoked by archive job after the archive has completed to reset the status to READY.

See the *CICS Operations and Utilities Guide* for more information about the DFHJACDU utility program.

---

## Using automatic journal archiving

The following steps are a guide to using automatic journal archiving:

1. Code the AUTOARCH option on the JOUROPT operand of the DFHJCT TYPE=ENTRY macro. This invokes automatic journal archiving for the journal specified on the JFILEID operand.
2. If you code AUTOARCH, also code JTYPE=DISK2 in the JCT.
3. Code the ARCHJCL parameter of the DFHJCT TYPE=ENTRY macro. This defines the name of the archiving job skeletal JCL in the journal archiving JCL sublibrary.
4. Define the following:
  - A journal archive control data set (DFHJACD) as a VSAM relative-record data set (RRDS).
  - A journal archive JCL VSE/ESA sublibrary into which you catalog your skeletal archiving jobs.
5. Create a skeletal archive job and catalog it with a member name of the value on the JCT ARCHJCL= parameter, and a member type of DFHJASP. A sample job DFH\$ARCH is supplied in the VSE/ESA sublibrary, PRD1.BASE.
6. If you currently use the DFHXJCO and DFHXJCC user-replaceable modules to automate archiving, rewrite your versions of these URMs or replace them with the CICS-supplied dummy versions.
7. Ensure the VSE/ESA sublibrary containing skeletal JCL updated with journal-specific information is in the LIBDEF SOURCE concatenation chain.

---

## Changes to CICS externals

Change	What it does	Where to find more information
System Definition		
New data set DFHJACD	Used for automatic journal archiving	<i>CICS System Definition Guide.</i>
Operations		
New DFHJACDU utility program	Inspects and updates journal status	<i>CICS Operations and Utilities Guide.</i>
Resource Definition		
ARCHJCL parameter of DFHJCT TYPE=ENTRY macro	Specifies name of the sublibrary member containing skeletal JCL	<i>CICS Resource Definition Guide.</i>
AUTOARCH option on the JOUROPT parameter of the DFHJCT TYPE=ENTRY macro	Causes automatic journal archiving to be invoked	<i>CICS Resource Definition Guide.</i>



---

## Chapter 23. Improved monitoring facilities

There are significant changes to monitoring to improve:

- Robustness of monitoring
- Availability of monitoring services
- Offline performance analysis and tuning
- Data correlation.

---

### Obsolete function

The following functions are obsolete:

- Accounting class data
- Global performance class data
- EVENT parameter on the DFHMCT TYPE=INITIAL macro
- ACCOUNT option on the CLASS parameter of the DFHMCT TYPE=EMP macro
- ACCOUNT option on EXEC CICS TRACEID MONITOR command
- CSTT transaction (replaced by the new option on the CEMT transaction)
- DFHEMP macro

---

### Differences in data collection

Statistical and monitoring data is written to Data Management Facility (DMF) Data Handler data sets instead of CICS journals. This data can be used for subsequent processing by the monitoring report sample program, DFH\$MOLS, using monitoring dictionary data produced by the new monitoring utility, DFHMNDUP.

All the DMF records written by CICS conform to the System Management Facility (SMF) relocatable format for the SMF header and SMF product section, including the use of record sub-types. This means that CICS monitoring, statistics, and journaling can all output SMF type 110 records. The record sub-type field in the SMF header part of each SMF type 110 record can be used to identify the type of data in the record (for example, the sub-type for the CICS statistics record is X'0002').

General information about creating data sets is provided in the *CICS System Definition Guide*.

Detailed information about the DMF utility itself and about DFHMNDUP is provided in the *CICS Operations and Utilities Guide*.

Transaction monitoring collects performance class and exception class data:

- **Performance class records**  
These records provide detailed transaction and resource usage information that you can use for accounting, performance analysis, and capacity planning. Data contains information relating to individual transaction usage, such as response time, CPU usage, time spent waiting for I/O, and so on. At least one performance record is written for each transaction. You can request

performance class records for long-running transactions and by unit-of-work (syncpoints).

- **Exception class records**

These records provide information about exceptional conditions experienced by individual transactions. An exception record is created every time the resources that are being monitored by exception class monitoring become constrained by system bottlenecks. Information in these records can help you identify the constraints that affect the performance of your transactions.

Exception records are *fixed format*, and the exception data section now contains a single exception record representing a single exception condition.

The format of an exception record can be mapped to the MNEXCDS DSECT which you can generate using the DFHMNEXC macro.

## Effects of the restructure of CICS internal code

- New monitoring code is now located above 16MB, together with its control blocks and all user storage.
- The monitoring (MN) domain now controls all monitoring functions within CICS. The monitoring domain runs its own validation checks on some of its control blocks so that it can detect errors at the earliest opportunity.

All monitoring modules have recovery routines that handle program checks, abends and loops. Furthermore, program checks in the MN domain that result from invalid addresses being passed on user event monitoring points (EMPs) can be detected early, and an appropriate response is returned to the application program.

- The monitoring area is not part of the TCA so it should no longer be susceptible to transaction work area (TWA) overruns.
- The DFHEMP macro is not supported. Change applications that use DFHEMP to use either EXEC CICS ENTER TRACEID MONITOR (supported for compatibility with earlier releases of CICS) or EXEC CICS MONITOR.

## Changes to the monitoring control table (DFHMCT)

The monitoring control table definition is revised to make the installation and customization of CICS monitoring easier. In addition, you can now use CICS monitoring without defining *any* monitoring control tables. If you switch on CICS monitoring without an MCT, CICS dynamically builds one using default values. These defaults are as follows:

- All monitoring classes available (exception and performance)
- All CICS-defined data fields are collected
- CPU=NO
- CONV=NO

### The DFHMCT TYPE=INITIAL macro

The EVENT parameter is no longer required. The CPU and CONV parameters are added to the TYPE=INITIAL macro.

```
DFHMCT TYPE=INITIAL
      [,SUFFIX=xx]
      [,CPU=YES|NO]
      [,CONV=YES|NO]
```

### The DFHMCT TYPE=EMP macro

The ACCOUNT operand of the CLASS parameter is obsolete, as is the ACCOUNT parameter and its options. The other parameters remain as they were in previous versions.

```
DFHMCT TYPE=EMP
      ,CLASS=PERFORM
      ,ID={number} (PP,number|entryname.number)
      [,CLOCK=(number,name1[,name2,...])]
      [,COUNT=(number,name1[,name2,...])]
      [,FIELD=(1,name)]
      [,PERFORM=(option[,...])]
```

### The DFHMCT TYPE=RECORD macro

This macro definition is required only when you want to use performance class data fields selectively. It is no longer required for exception class monitoring. The accounting class and global performance data collection are withdrawn in this version. Only the following parameters are now supported on the TYPE=RECORD macro instruction:

```
DFHMCT TYPE=RECORD
      ,CLASS=PERFORM
      [,EXCLUDE={ALL} (n1[,...])]
      [,INCLUDE=(m1[,...])]
```

### The DFHMCT TYPE=FINAL macro

This macro is unchanged.

There are new sample MCTs for a TOR (DFHMCTT\$), an AOR (DFHMCTA\$), and an FOR (DFHMCTF\$). These sample MCTs show the types of data fields that can be excluded to reduce the size of the performance class record produced.

## New monitoring system initialization parameters

The MONITOR system initialization parameter, with its positional operands, is replaced by the following parameters:

- The MN parameter to activate or deactivate CICS monitoring.
- The MNEXC parameter to activate or deactivate exception class monitoring.
- The MNPER parameter to activate or deactivate performance class monitoring.
- The MNCONV parameter to specify whether conversational tasks are to have separate performance class records for each terminal I/O request.
- The MNSYNC parameter to specify whether or not separate performance class records are to be produced for each syncpoint request. This excludes syncpoint requests during task termination or rollback.
- The MNTIME parameter to specify whether monitoring should return the timestamp fields (transaction start and stop times) in GMT or local time to an application using the SPI command EXEC CICS COLLECT STATISTICS MONITOR(*taskno*).
- The MNFREQ parameter to specify the class interval when monitoring should automatically produce a separate performance class record for a long-running transaction.

See the *CICS System Definition Guide* manual for more detailed information about these parameters.

## Controlling monitoring

There are new CEMT and EXEC CICS INQUIRE and SET MONITOR commands that enable you to change monitoring options dynamically. Convert any programs that control monitoring online to use the new EXEC CICS commands.

The new CICS monitoring utility, DFHMNDUP, generates a monitoring dictionary record (in a sequential data set) for use with monitoring data extracted from DMF data sets.

CICS writes a new performance dictionary record to the currently open DMF data set, and then begins to write the related monitoring data whenever:

- CICS is started with the performance class active, and CICS monitoring switched on.
- The status of the monitoring performance class is changed from inactive to active, and with CICS monitoring switched on. If monitoring is off and the monitoring performance class is switched from inactive to active, a dictionary record is scheduled for the next time monitoring is activated.

Try the enhanced monitoring report sample program, DFH\$MOLS, as an example of how to process CICS monitoring data. DFH\$MOLS can process data from multiple CICS systems.

There is a global user exit point in the MN domain, XMNOUT. XMNOUT replaces the monitoring task-related user exits of CICS/VSE Version 2. Any program that used the monitoring task-related user exits in CICS/VSE Version 2 must be rewritten to use XMNOUT, and to conform to the new user exit linkage conventions. Control is passed to XMNOUT before the monitoring record is output to DMF. On return from XMNOUT, the monitoring record is output or suppressed depending on the exit return code.

---

## Trace

- Each entry and exit to the MN domain is traced, giving request parameter lists, together with the result of the domain call.
- Exception events are always traced, even if you have switched trace off. This ensures that you receive important trace information in the event of a failure.
- You can activate tracing by terminal, transaction, or class. There are different levels of trace for each of the domains, each level giving more detail. The trace point IDs for the MN domain are in the form MN xxxx; the corresponding trace levels are MN 1, MN 3 and Exc.

---

## Dump

A VSE SDUMP is produced for all system dumps and is written to a VSE dump library.

A monitoring domain summary is included in the SDUMP, giving details about items such as:



- Which monitoring services are active
- Whether the XMNOUT exit is active
- Name of the MCT in use
- User EMP address list
- Highest and lowest EMP ids
- Number of user clocks
- CPU option setting
- CONV option setting
- User area length
- User dictionary length
- User dictionary address

---

## Using new CICS monitoring functions

- 1 Ensure that you create DMF data sets that are large enough to store the increased volume of information being written to them.
- 2 Rewrite any program that used the monitoring task-related user exits in CICS for VSE/ESA Version 2.3 to use the new monitoring global user exit, XMNOUT, and to conform to the conventions for user exit linkage.
- 3 Ensure that the CSTT transaction is not used. This transaction is replaced in CICS TS Release 1 by CEMT INQUIRE or SET MONITOR.
- 4 Rewrite any programs (offline and online) that interpret monitoring data to use the new monitoring dictionary and record formats.
- 5 Reassemble or recompile any programs that use the DFHEMP macro. This macro is not supported. Use either EXEC CICS ENTER TRACEID MONITOR or EXEC CICS MONITOR.
- 6 Ensure that any use of EXEC CICS ENTER TRACEID MONITOR does not include the ACCOUNT option. ACCOUNT is ignored at execution time and flagged as “obsolete” by the translator.
- 7 Remove obsolete DFHMCT options (see table below)
- 8 Convert any programs that control monitoring online to use the new EXEC CICS INQUIRE and SET MONITOR commands.
- 9 Set monitoring options at system initialization using the new SIT parameters or dynamically using the EXEC CICS commands and CEMT commands.
- 10 To process monitoring information offline, make sure that DMF is active.

---

## Changes to CICS externals

Change	What it does	Where to find more information
System definition		
MN system initialization parameter	Activates CICS monitoring	<i>CICS System Definition Guide</i>
MNEXC system initialization parameter	Activates exception class monitoring	<i>CICS System Definition Guide</i>

<b>Change</b>	<b>What it does</b>	<b>Where to find more information</b>
MNPER system initialization parameter	Activates performance class monitoring	<i>CICS System Definition Guide</i>
MNCONV system initialization parameter	Specifies whether conversational tasks are to have separate performance class records for each terminal I/O request.	<i>CICS System Definition Guide</i>
MNSYNC system initialization parameter	Specifies whether separate performance class records are produced for each syncpoint request.	<i>CICS System Definition Guide</i>
MNTIME system initialization parameter	Specifies whether monitoring will return the timestamp fields to an application using the EXEC CICS COLLECT STATISTICS MONITOR command.	<i>CICS System Definition Guide</i>
MNFREQ system initialization parameter	Specifies the class interval when monitoring automatically produces a separate performance class record for a long-running transaction.	<i>CICS System Definition Guide</i>
Obsolete system initialization parameter, MONITOR		<i>CICS System Definition Guide</i>
DFHMNDUP utility	Generates a monitoring dictionary record (in a sequential data set) for use with monitoring data extracted from DMF data sets.	<i>CICS System Definition Guide</i>
DFH\$MOLS monitoring report sample program	Enhanced example of how to process CICS monitoring data	<i>CICS System Definition Guide</i>
Resource definition		
DMF initialization table	Collects monitoring and statistics information	<i>CICS System Definition Guide</i>
Obsolete options: <ul style="list-style-type: none"> <li>• EVENT on DFHMCT TYPE=INITIAL</li> <li>• ACCOUNT on CLASS parameter of DFHMCT TYPE=ENTRY</li> <li>• DATASET, FREQ, and MAXBUF on DFHMCT TYPE=RECORD</li> </ul>		<i>CICS Resource Definition Guide</i>

<b>Change</b>	<b>What it does</b>	<b>Where to find more information</b>
New sample MCTs: <ul style="list-style-type: none"> <li>• DFHMCTT\$ (TOR)</li> <li>• DFHMCTA\$ (AOR)</li> <li>• DFHMCTF\$ (FOR)</li> </ul>	Show the types of data fields that can be excluded so that you can reduce the size of the performance class record produced by CICS monitoring.	<i>CICS Resource Definition Guide</i>
System programming		
New commands, EXEC CICS INQUIRE and SET MONITOR	Control monitoring options	<i>CICS System Programming Reference</i>
Obsolete option, ACCOUNT, on EXEC CICS ENTER TRACEID MONITOR		<i>CICS Application Programming Reference</i>
CICS-supplied transactions		
New commands, CEMT INQUIRE and SET MONITOR	Control monitoring options	<i>CICS-Supplied Transactions</i>
Obsolete transaction, CSTT	Replaced by the MONITOR option on CEMT INQUIRE and SET	<i>CICS-Supplied Transactions</i>
Customization		
XMNOUT global user exit	Replaces the monitoring task-related user exits of CICS for VSE/ESA Version 2.3	<i>CICS Customization Guide</i>
Problem determination		
New monitoring message set, MN	Reports activity of CICS monitoring	<i>VSE/ESA Messages and Codes Volume 3</i>
Trace point IDs MN xxxx	Trace points for the monitoring domain.	<i>CICS Trace Entries</i>
Monitoring domain summary in the VSE SDUMP	Provides information about monitoring activity at the time of a system dump.	<i>CICS Problem Determination Guide</i>



---

## Chapter 24. Improved collection of statistics

The new statistics (ST) domain controls the collection of resource statistics for a CICS system. The ST domain collects statistical data as follows:

- At intervals you specify
- At system quiesce
- At logical end-of-day
- When requested by you

---

### Obsolete function

The following are obsolete:

- The CSTT transaction (replaced by new CEMT commands)
- The CSSM and CSSN transient data queues
- Data sets for DFHSTM and DFHSTN

### CSTT statistics transaction replaced by CEMT enhancements

The CSTT transaction, used for controlling CICS statistics in previous versions, is replaced by enhancements to the CEMT transaction. There are three new CEMT statistics options:

- INQUIRE STATISTICS, which causes CICS to return information about the status of CICS statistics collection
- SET STATISTICS, which enables you to set the collection options
- PERFORM STATISTICS, which causes CICS to write to the DMF data set the data collected since the start of the current interval.

### CSSM and CSSN transient data queues withdrawn

The transient data queues CSSM and CSSN are no longer used, because all statistics data is written to a DMF data set. You should reassemble your destination control table (DCT) without the definitions for CSSM and CSSN.

### DLBL statements for DFHSTM and DFHSTN

Because CSSM and CSSN are no longer used, remove the DLBL statements for DFHSTM and DFHSTN extrapartition transient data sets from the CICS startup job stream. If you have a job step in your CICS startup job stream to execute the statistics offline utility DFHSTUP, which uses these data sets, remove it and design a separate VSE job to print statistics from an unloaded DMF data set.

Design the new DFHSTUP job based on your installation's standard methods of handling DMF data sets. There is a sample job in the *CICS Operations and Utilities Guide*, with notes, illustrating how you can process statistics data written to DMF data sets. Note that the new DFHSTUP includes an internal call to an external sort program (such as DFSORT™/VSE). Include DLBL statements for the necessary sort work files in the DFHSTUP job step.

## Statistics Data Records

Each statistics record contains an SMF header, an SMF product section, and one CICS data section. A new field in the SMF product section gives the time that CICS last reset the statistics counters.

---

## Differences in data collection

The following changes to data collection are made:

- All statistics data is written to DMF data sets, rather than to CICS journals.

Statistical information written to DMF data sets can subsequently be used by the statistics offline utility program (DFHSTUP) to produce formatted reports.

Each statistics record contains an SMF header, an SMF product section, and one CICS data section. A new field in the SMF product section gives the time that CICS last reset the statistics counters.

Because the ST domain writes to DMF data sets, you must ensure that your DMF data sets are large enough to accommodate the volume of CICS data. When you calculate the size of your DMF data sets, remember to take into account space for CICS monitoring data, which is also written to SMF data sets. See the *CICS Performance Guide* for details of CICS statistics records and their sizes.

- Changes to the method of collecting statistics:

- **Interval statistics**

CICS collects statistics at user-specified intervals, and writes them to a DMF data set, unless statistics collection is set off. CICS always *collects* statistics—users are not able to prevent interval collection—but they are written to DMF only if the recording option is switched on. You can switch statistics on or off by means of either a CEMT or an EXEC CICS command. The user-specified interval has to be between 1 minute and 24 hours; the default is 3 hours.

- **Requested statistics**

You can also request statistics during collection intervals, in which case CICS will write the statistics accumulated from the start of the current interval. Like the interval statistics it collects, CICS writes requested statistics to a DMF data set, but will not reset the counters as it does for a normal interval collection.

- **End-of-day statistics**

CICS collects and writes end-of-day statistics at logical end-of-day, or when a normal shut down has been performed. The end-of-day statistics data represents the final interval of the day, and CICS writes these to a DMF data set. (The default logical end-of-day is midnight.)

- **Unsolicited statistics**

CICS writes statistics about dynamically allocated and deallocated resources (for example, autoinstalled terminals that are logged off and have their definitions deleted from the CICS system) before they are deleted. Such statistics might otherwise be lost. CICS writes these unsolicited statistics to a DMF data set.

Other changes include:

- Changes to the statistics recording status
- More information in statistics records
- More informative statistical data produced for functional areas.
- Statistical timestamps in both GMT and local time
- Improvements in the consistency of the last reset time and the use of local time
- Synchronization of times between offline and online reporting of statistics at the last reset time

---

## Changes to controlling statistics

- **System initialization parameter, STATRCD**

In addition to using CEMT and the API to switch statistics on or off, you can also use the STATRCD system initialization parameter to control statistics status at CICS startup. The default for the STATRCD parameter is **OFF**. Status is recorded in the CICS global catalog during warm or emergency restarts.

- **New EXEC CICS commands**

There is now a programming interface to CICS statistics using the following new EXEC CICS commands, which allow you to control the collection and writing of statistics to the DMF data set, and request the collection of statistics for a specified resource type.

- **EXEC CICS INQUIRE and SET STATISTICS**

Use the INQUIRE and SET STATISTICS commands to retrieve and to change the recording of statistics information.

- **EXEC CICS COLLECT STATISTICS**

Use the COLLECT STATISTICS command to retrieve the current statistics for a single resource, or global statistics for a class of resources.

- **EXEC CICS PERFORM STATISTICS RECORD**

Use the PERFORM STATISTICS RECORD command to record statistics immediately.

For more information about these commands, see the *CICS System Programming Reference* manual.

- **New CEMT commands to replace the CSTT transaction**

There are three new CEMT statistics options that enable you to control the collection of CICS statistics, and to write the data collected since the start of the current interval to the currently open DMF data set.

- INQUIRE STATISTICS, which causes CICS to return information about the status of CICS statistics collection

- SET STATISTICS, which enables you to set the collection options

- PERFORM STATISTICS, which causes CICS to write to the DMF data set the data collected since the start of the current interval.

- **New parameters on the XSTOUT global user exit**

More information is passed to the XSTOUT global user exit. The DFHUEPAR parameters include the following SMF product section fields as XSTOUT exit-specific parameters:

- The collection date in the form *mmdyy* and *mmdyyy*
- The collection time in the form *hhmmss*
- The length of the interval in the form *hhmmss*
- The interval number

- **Enhancements to the statistics utility program, DFHSTUP**

You can select reports by resource type, and set new control parameters to select (or ignore) specific types of resource in a DFHSTUP report.

The FC report produced by DFHSTUP has been made easier to read.

- **A new sample suite of programs, DFH0STAT**

These programs illustrate the different ways of using the EXEC CICS INQUIRE and COLLECT STATISTICS commands to produce statistical information about a CICS region. The reports include a CICS and VSE storage analysis. Statistics obtained using these commands can be used to help in specifying the various DSASIZE parameters.

For more information about DFH0STAT, see the *CICS Operations and Utilities Guide* and the *CICS Performance Guide*.

---

## Using the new CICS statistics function

- 1 Ensure that you create DMF data sets that are large enough to store the increased volume of statistical information being written to them. For more information see the *CICS Operations and Utilities Guide*.
- 2 Rewrite any programs (offline and online) that interpret statistics data to use the SMF-format data, instead of the format previously used for CSSM and CSSN.
- 3 Reassemble DCTs to remove the definitions for CSSM and CSSN.
- 4 Review your operating procedures for the use of CSSM, CSSL, and CSSN and modify them if necessary.
- 5 Change any operating procedures that use the CSTT transaction to use CEMT instead.
- 6 Remove any references to DFHSTM or DFHSTN from any CICS startup job stream.
- 7 If you have a jobstep in your CICS startup jobstream to execute DFHSTUP, remove it. Design a separate VSE/ESA job to print statistics from an unloaded DMF data set.
- 8 To process statistical information offline, make sure that DMF is active. For details see the *CICS Operations and Utilities Guide*.

#



## Changes to CICS externals

Change	What it does	Where to find more information
System definition		
STATRCD system initialization parameter	Allows you to set the status of statistics recording status on or off at system initialization	<i>CICS System Definition Guide</i>
Enhancements to the statistics utility program, DFHSTUP	Enables you to select reports by resource type and set new control parameters to select or ignore specific types of resource in a DFHSTUP report	<i>CICS System Definition Guide</i>
New sample programs, DFHOSTAT	Illustrate the different ways of using EXEC CICS INQUIRE and EXEC CICS COLLECT STATISTICS commands to produce CICS statistics	<i>CICS System Definition Guide</i>
Obsolete transient data queues, CSSM and CSSN		<i>CICS System Definition Guide</i>
Resource definition		
DMF initialization table	Collects monitoring and statistics information	<i>CICS Resource Definition Guide</i>
System programming		
EXEC CICS COLLECT STATISTICS	Retrieves current statistics for a single resource, or global statistics for a class of resources	<i>CICS System Programming Reference</i>
EXEC CICS INQUIRE and SET STATISTICS	Sets the collection options for CICS statistics	<i>CICS System Programming Reference</i>
CICS-supplied transactions		
Obsolete transaction, CSTT	Replaced by extensions to CEMT (see below)	<i>CICS-Supplied Transactions</i>
CEMT INQUIRE STATISTICS	Returns information about the status of CICS statistics collection	<i>CICS-Supplied Transactions</i>
CEMT SET STATISTICS	Enables you to set the collection options for CICS statistics	<i>CICS-Supplied Transactions</i>
CEMT PERFORM STATISTICS	Causes CICS to write the data collected since the start of the current interval	<i>CICS-Supplied Transactions</i>
Customization		

Change	What it does	Where to find more information
New parameters on the XSTOUT global user exit	Pass the following information to the global user exits: <ul style="list-style-type: none"> <li>• Collection date and time</li> <li>• Length of the interval</li> <li>• Interval number</li> </ul>	<i>CICS Customization Guide</i>

---

## Chapter 25. Improved RDO

This chapter describes the many extensions to CICS resource definition online (RDO) facilities.

---

### RDO for files

CICS supports the online definition of files. You can use RDO to manage the following resources that are associated with CICS files:

- VSAM files
- Remote VSAM files
- Remote DAM files
- VSAM local shared resource (LSR) pools
- Shared data tables

For an active CICS system to use a file, install the file definition onto the system. CICS file control uses the definition to:

- Find the file when it needs to access it
- Keep count of the number of tasks using the file
- Maintain a record of the address of the associated ACB
- Capture processing statistics.

The FILE definition can be used to describe the physical and operational characteristics of a file to CICS file control. The definition includes keywords that provide information about record characteristics, types of operations allowed on the file, recovery attributes, and the operations that are to be journaled. This information is used to generate control information used by CICS as well as an access control block (ACB). CICS files correspond to physical data sets that must be defined to VSAM before they are used. However, it is not necessary to provide start up JCL for the files. Labels for these are built and submitted dynamically by CICS at file-open time by a new subcomponent.

When multiple CICS systems are connected, they can share each other's files. All files usually have to be defined to CICS. The exception is for files on other systems that use their system's SYSID. If a file on another system does not use the appropriate SYSID, it must be defined to CICS as 'remote'.

The resource attributes needed by CICS for remote files are not specified to the access method used, so you can define both remote DAM and remote VSAM files using RDO.

If you name a REMOTESYSTEM, you may also supply a REMOTENAME, which is the name of the file used in the remote system.

If you specify a REMOTESYSTEM name that corresponds to the system in which the install has been made, a local FCT entry will be created. Otherwise a remote FCT entry will be created.

---

## Specifying read-only access to the CSD

You can specify read-only access to the CSD to enable use of DFHCSDUP LIST and EXTRACT commands while CEDA or CEDB is running. To do this, on the PARM parameter of the EXEC statement:

```
// EXEC PGM=DFHCSDUP,SIZE=DFHCSDUP,PARM='CSD(READONLY)'
```

Default access remains read-write.

---

## Associating descriptive comments with resource definitions in the CSD

You can associate a descriptive comment with any resource definition in the CSD.

It is possible to create a 58-character description field and assign it to a resource definition newly created or any existing resource definitions. The field can be changed at any time by using the CEDA or CEDB DEFINE and ALTER commands. This facility is available for all CSD resource types that can be defined in CICS.

The DESCRIPTION field is included on the CEDA and CEDB DEFINE and ALTER panels, and is printed in the offline utility listing of resource definitions produced by the DFHCSDUP "LIST...OBJECTS" command. It is also included in the list of attributes of the resource definition logged in the CSDL transient data queue when a successful DEFINE or ALTER command is performed.

---

## Invoking DFHCSDUP from user programs

You can invoke the DFHCSDUP utility program from your own programs when running in batch mode. This enables you to create a flexible interface to DFHCSDUP, based on your system needs. Appropriate entry parameters are specified to cause DFHCSDUP to pass control to an exit routine at one of five new user exit points:

- |                       |  |
|-----------------------|--|
| <b>Initialization</b> | Invoked once, on entry to DFHCSDUP. It allows a routine to perform exit-related initialization.                                    |
| <b>Termination</b>    | Can be used for final housekeeping duties. It is invoked before a normal or abnormal termination of DFHCSDUP.                      |
| <b>Extract</b>        | Is invoked at various stages during processing of the EXTRACT command.   |
| <b>Put-message</b>    | Is invoked whenever a message is issued. You can use your own put-message routine to terminate DFHCSDUP.                           |
| <b>Get-command</b>    | Reads in command lines. It can be invoked multiple times, at the point where DFHCSDUP would normally read in commands from SYSIPT. |

You can use these new exit points to pass commands to DFHCSDUP, or to respond to messages produced by DFHCSDUP processing.

---

## Using the DFHCSDUP ALTER command

CICS supports the DFHCSDUP ALTER command. It can be used, for example, to make a program resident, and to change the status of a whole group of programs. ALTER must not be used to change the value of the attributes of a TYPETERM definition on which other attributes depend.

Changes to resource definitions in the CSD do not take effect until the group in which the resource definition resides is installed.

---

## Sharing the CSD

If you want to share your CSD with earlier releases, you should not experience any compatibility difficulties. Some attributes are obsolete and are removed from the CSD definitions. However, CICS continues to display these attributes on CEDx panels as protected fields.

To share the CSD between releases:

1. Upgrade the CSD.
2. Update the protected fields when they are in ALTER mode by pressing the compatibility (PF2) key.
3. Add the compatibility group, DFHCOMP1, to any CICS for VSE/ESA Version 2.3 group list. DFHCOMP1 contains definitions for all the CICS for VSE/ESA Version 2.3 resources that are now obsolete.

**Note:** All updates to the CSD must be performed from a CICS TS region or using a CICS TS version of DFHCSDUP.

---

## Dual-purpose resource definition

Dual-purpose resource definition is extended to include transaction routing and function shipping.

This enables you to review the way in which resources are organized within your resource definition GROUPs. For example, it is possible to have two GROUPs, one containing all the resources for a CICS TOR, and one containing all the resources for a CICS AOR. When using shared resource definitions, you may decide to maintain three GROUPs instead, containing:

1. Resources specific to the TOR
2. Resources specific to the AOR
3. Resources that will be installed in both the TOR and AOR.

You can define these resources as local or as remote.

In addition to sharing terminal definitions between systems, CICS extends dual-purpose resource definition to include transaction definitions and file definitions.

For transaction definitions, dual-purpose resource definition means, for example, that it is possible to share a single transaction definition between a terminal-owning region (TOR) and an application-owning region (AOR). When the definition is installed on the TOR, CICS compares the SYSIDNT name with the

REMOTESYSTEM name. If they are different, a remote transaction definition is created. When the definition is installed on the AOR, CICS compares the REMOTESYSTEM name with the SYSIDNT name. If they are the same, a local transaction definition is installed.

Similarly, it is possible to share a FILE definition between an AOR and a data-owning region (DOR), with the correct type of file definition resulting on each system.

---

## Removing installed resource definitions

You can now use either the CEMT DISCARD transaction, or the EXEC CICS DISCARD command in an application program to remove the following installed resource definitions from the active CICS system:

- AUTINSTMODEL
- FILE
- PARTNER
- PROFILE
- PROGRAM
- TRANCLASS
- TRANSACTION.

The DISCARD function is the opposite of the INSTALL function. A resource definition removed using the DISCARD command remains in the CSD and can be reinstalled when needed using CEDA, unlike a resource definition removed using the DELETE command, which is deleted from the CSD and has to be redefined and reinstalled if needed again.

Resources currently in use are protected from being discarded. You cannot discard CICS-owned resources (names beginning with "DFH") and CICS-supplied transactions (names beginning with "C").

You can control the use of DISCARD because users of the DISCARD command must have resource authority (in the case of files, programs, and transactions), and read authority.

Resources are logged to the following new transient data queues when they are successfully discarded or installed:

- CAIL** Autoinstall models
- CSFL** Files
- CSKL** Profiles and transactions
- CSRL** Partners
- CSPL** Programs

Logging gives you additional monitoring and statistical data, and can help you to determine why a program or transaction failed, or why a particular resource could not be accessed.

---

## Using the programmable interface to CEDA

You can use the programmable interface to the RDO transaction, CEDA. You can invoke the functions provided by RDO from application programs using commands such as:

```
EXEC CICS LINK PROGRAM('DFHEDAP')
          COMMAREA(cedaparm)
```

where DFHEDAP is the name of the entry point in the RDO program, and *cedaparm* is a user-defined name of the parameter list. See the *CICS Customization Guide* for more information.

---

## Installing single resources

The CEDA command language is extended so that you can use the INSTALL command for a **single** resource, rather than having to INSTALL the whole group.

This can avoid the problem of installing a group in which one of the resources is busy. If a group is installed, the install fails if **any** of the resource definitions in the group is busy. *Single-resource* install means that only the resources required are installed.

The EXPAND GROUP display is improved. The letter 'i' entered beside a resource within the group performs the INSTALL of that resource.

**Note:** There are some intercommunication resources that cannot be installed in this way because they are too strongly bound to other members of the group, for example, some CONNECTION and SESSION definitions, or TERMINAL and related TYPETERM definitions.

You can install the following resources only as part of a group:

- CONNECTION definitions, except a CONNECTION with ACCESSMETHOD(INDIRECT)
- SESSION definitions

**Notes:**

1. Use 'i' command against resources displayed by the CEDA EXPAND (or DISPLAY) GROUP command to install single resources.
2. For TERMINAL and related TYPETERM definitions, ensure the TYPETERM that is referred to by a TERMINAL definition is installed **before** the TERMINAL definition that refers to it.

---

## Immediate commitment of resources

As in previous releases, resources are committed at the group level, and at the resource level.

The following resource types are committed by individual resource, and not by the complete group. For these resources, if an INSTALL command fails part way through, they are left in a committed state:

- AUTINSTMODEL
- FILE

- LSRPOOL
- PROFILE
- PROGRAM
- MAPSET
- PARTITIONSET
- PARTNER
- TRANCLASS
- TRANSACTION

The following resources are committed at the group level. These resources are backed out after a partially successful install:

- CONNECTION
- SESSIONS
- TERMINAL
- TYPETERM



---

## Chapter 26. Resolving in-doubt resources

If a CICS region fails during a two-phase commit syncpointing process, there is a period during which it does not know whether it has to back out or commit its updates. This period is known as the *in-doubt window* which is when:

- A subordinate CICS region (or other remote system) has replied in response to a PREPARE request, and
- Has written a log record of its response to signify that it has entered the in-doubt state, and
- Does not yet know the decision of its coordinator (to commit or backout).

During this period, the resource data is potentially inconsistent between the failed CICS region and other CICS regions that have updated resources within the same unit of work.

The danger is that after the failed CICS region has restarted, new transactions can use the inconsistent data; they may read or update the data, and, in either case, the results of these new transactions can be incorrect.

If a CICS region fails, it issues message DFHZN2101 to identify any units of work, relating to specific tasks, which are considered to be in-doubt. When the CICS region recovers from the failure, it issues further messages to resolve the in-doubt tasks to "In sync" or "Out of sync". The messages indicate that data integrity may be exposed, but they do not indicate which resources and records are exposed. Also, only CICS resources are reported; not DB2 or other resource types. To resolve the in-doubt resource updates, interpret the journal data to see whether there has been a loss of integrity.

---

### DFH\$IWUP utility program

The in-doubt window resolution utility program, DFH\$IWUP, helps you to resolve in-doubt resources relating to:

- DFHZN2103 messages (for databases out of synchronization) after CICS has reported the result of its recovery logic
- Any units of work whose outcome is unknown after a recovery failure. (Note that DFH\$IWUP utility program cannot resolve any in-doubt resources updated using LU6.1 protocols.)
- Operator action that has prevented CICS from resolving DFHZN2101 messages.

DFH\$IWUP is an exit program to the CICS journal utility program, DFHJUP. DFH\$IWUP scans the journal for unit-of-work IDs as they appear in the DFHZN210x messages generated to the CSMT log, and outputs a report of all journal records for tasks relating to the unit-of-work IDs. For each journal record found, the report lists, by task:

- The user data records, including the resource name and record key if appropriate, that were affected by in-doubt tasks.

This helps you to investigate the messages that report loss of integrity, enabling you to quantify the effect of the failure.

- The resource type, and type of request.

This enables you to differentiate spurious messages from significant ones.  
(Spurious warning messages of in-doubt problems are issued by CICS regions that have not updated any resources.)

You can copy the CSMT log and edit it to produce your own list of unit-of-work IDs to be investigated.

You should run the DFH\$IWUP utility program against copies of the system logs for all CICS regions involved in the units of work that are in-doubt.

**Note:** You cannot use the DFH\$IWUP utility program to resynchronize resources.

You can create your own DFH\$IWUP utility program. The source code is supplied in member DFH\$IWUP of the VSE/ESA sublibrary, PRD1.BASE.

For more information about the DFH\$IWUP utility program, see the *CICS Operations and Utilities Guide*.

---

## Using the DFH\$IWUP utility program

The following summarize the steps to using the DFH\$IWUP utility program:

1. To create your own DFH\$IWUP utility program, customize the code supplied in member DFH\$IWUP of the VSE/ESA sublibrary, PRD1.BASE.
2. Assemble and link-edit the DFH\$IWUP program into your CICS environment.
3. Submit JCL to run the DFH\$IWUP utility program.

---

## Chapter 27. Miscellaneous changes to CICS

This chapter describes other changes in CICS, including the following:

- “Recovery of VSAM files”
- “Recovery from DTB failures”
- “Support for forward recovery utilities” on page 144
- “Improvements to MRO” on page 144
- “Enhancements to transaction management” on page 145

---

### Recovery of VSAM files

CICS enhances the availability and the integrity of VSAM data sets by:

- Allowing you to automatically archive CICS journal data sets
- Ensuring data set integrity in the event of a dynamic transaction backout (DTB) failure, without having to abend CICS.
- Simplifying CICS restart procedures.

For further information on this subject, see the *CICS Recovery and Restart Guide*.

---

### Recovery from DTB failures

Dynamic transaction backout (DTB) and CICS emergency restart back out uncommitted changes to data sets by reapplying logged before-images using calls to file control.

If file control returns a bad response code on any of these calls, a backout failure has occurred. In previous releases of CICS, to recover from such a failure, the XDBFERR global user exit was usually taken, and you were advised to abend CICS in this exit. You then had to determine the cause of the failure and take the necessary steps to recover from the failure before restarting CICS.

You no longer have to abend CICS. When backout failures are detected during DTB (or during emergency restart), CICS:

- Sets the backout status field in the CICS base cluster block (one for each base cluster) to “failed”
- Stores a backout-failed log record on the system log, to enable a backout utility to start and stop its scan of the log in the correct places, and to locate the relevant before-images
- Sets a backout-failed status record in the CICS global catalog, DFHGCD
- Closes all file control table entries against the base cluster, to prevent further updates on the damaged data set.

Backout failure control code also:

- Tells you when all required FCT entries have been closed
- Maintains the backout-failed status across all restarts
- Provides new CEMT and EXEC CICS commands to inquire on the backout status, and to reset the status after batch backout has been completed.

The user exit is still taken at the point of failure, but a CICS abend is no longer necessary to preserve data integrity.

You can use a batch backout utility to back out any uncommitted changes caused by failures during DTB or emergency restart. To recover a data set, run the utility against the damaged data set, using archived system journal data sets to back out the uncommitted changes.

For further information on this subject, see the *CICS Recovery and Restart Guide*.

---

## Support for forward recovery utilities

There are new options to control the logging of REDO records, and to control the journal to which the REDO records are written. These options are available only with resource definition online (RDO).

A basic requirement of any forward recovery utility is a log containing all the after images (often referred to as REDO records).

In CICS/VSE Version 2, user journaling allows a request for a REDO record for a VSAM file by specifying the appropriate JREQ options on the DFHFCT macro for a file. However, the journaling of these REDO records takes place before the VSAM request has been issued. If the request fails, a change is journaled, even though it did not actually take place.

CICS file control is changed so that all REDO records are journaled *after* a VSAM request has completed. In addition, you can maintain a journal of all REDO records (using the FWDRECOVLOG parameter on the RDO DEFINE FILE command), keeping records written for forward recovery separate from those written for user journaling. All UNDO records for backout recovery continue to be written to the system log.

For further information on this subject, see the *CICS Recovery and Restart Guide*.

---

## Improvements to MRO

MRO is enhanced to support:

- Queueing algorithm improvements for improved performance
- Duplicate logon names for EXCI support
- ESM support for MRO link security

MRO communication between systems continues to use the existing VSE/ESA XPCC interfaces, and supports communication between CICS TS Release 1 and CICS for VSE/ESA Version 2.3

For further information on this subject, see the *CICS Intercommunication Guide*.

---

## Enhancements to transaction management

CICS introduces a number of changes to transaction management:

- The number of transaction classes is now unlimited. The old limit of 10 transaction classes (TCLASS) is removed, and you can define each transaction class with an 8-character name. The 10 transaction classes supported in CICS/VSE Version 2 continue to be supported. Transaction classes are represented in by TRANCLASS resource definitions.
- Set a runaway time limit at transaction definition level. This will override the system default set by the ICVR system initialization parameter.
- System definition is simplified, with the removal of the AMXT and CMXT system initialization parameters.
- A new transaction manager, with global user exits to give much greater flexibility and control at the point of transaction attach.
- A new transaction definition option, STORAGECLEAR, indicates whether task lifetime storage for a transaction is to be cleared or not upon release. Users can use this function to prevent other tasks from accidentally viewing confidential or sensitive data that was being stored by the task in task lifetime storage.

For further information on this subject, see the:

- *CICS Resource Definition Guide*
- *CICS System Definition Guide*
- *CICS-Supplied Transactions* manual
- *CICS System Programming Reference* manual



---

## Chapter 28. Changes for system and resource definition

This chapter describes changes for system and resource definition.

---

### Changes to system definition

A number of functions are withdrawn. See also the “Changes to CICS externals” sections in this book for changes to system and resource definition, resulting from specific enhancements to CICS.

- **System initialization overlays (SIMODS)**

The SIMODS system initialization parameter is now obsolete.

- **Removal of documentation about the programmable interface to CEMT**

Users of earlier releases of CICS may be familiar with the old programmable interface to the master terminal program (DFHEMTA). Its use is still supported, but the documentation is available only with earlier releases of CICS/VSE.

The method of passing requests and receiving replies is unchanged. Provided the request does not use any abbreviations of commands, the same request can be invoked on new releases of CICS. The return codes for a new release of CICS are a superset of the return codes on earlier releases.

***You are recommended to use the SPI for master terminal functions.***

- **SYSGEN (DFHSG) macros**

The modules shown in Table 23 cannot be generated. Full-function versions of the modules are supplied.

*Table 23 (Page 1 of 2). Modules that can no longer be generated*

<b>DFHSG PROGRAM=</b>	<b>Module name</b>	<b>Module description</b>
BMS	DFHTPR	BMS page retrieval
CSA	DFHCSA	Common systems area
CSO	DFHACP	Abnormal condition
CSO	DFHTACP	Terminal abnormal condition
CSS	DFHTRAP	FE global trap exit program
DBP	DFHDBP	Dynamic transaction backout
ICP	DFHICP	Interval control program
ISC	DFHISP	Intercommunication program
ISC	DFHXFP	ISC transformer program
JCP	DFHJCP	Journal control
KPP	DFHRUP	Recovery utility program
SCP	DFHSCP	Storage control
SCP	DFHSCR	Storage control recovery
SRP	DFHSRP	System recovery
TBP	DFHDLBP	DL/I backout program

Table 23 (Page 2 of 2). Modules that can no longer be generated

DFHSG PROGRAM=	Module name	Module description
TCP	DFHCPY	VTAM 3270 print function support
TCP	DFHEXI	VTAM 3270 print function support
TCP	DFHGMM	VTAM Good Morning message program
TCP	DFHPRK	VTAM 3270 print function support
TCP	DFHP3270	3270 print function support
TCP	DFHRKB	VTAM 3270 print function support
TCP	DFHTCP	Terminal control program
TCP	DFHZCA	VTAM terminal control program module
TCP	DFHZCB	VTAM terminal control program module
TCP	DFHZCC	VTAM terminal control program module
TCP	DFHZCP	Common terminal control program module
TCP	DFHZCW	VTAM terminal control program module
TCP	DFHZCX	Common terminal control program module
TCP	DFHZCY	VTAM terminal control program module
TCP	DFHZCZ	VTAM terminal control program module
TCP	DFHZNAC	Node abnormal condition program
TCP	DFHZRLG	Response logging program
TCP	DFHZRSP	Resend program

Furthermore, the removal of the system generation feature means that far fewer modules are source-serviced. The source for modules that were source-serviced in previous releases (and which are now object-serviced) is no longer supplied.

---

## Changes to resource definition

Resource definition tables defined using macros are, at most, source-compatible between releases. However, source compatibility is not guaranteed.

## Obsolete function

A number of functions are withdrawn. See also the “Changes to CICS externals” sections in the rest of this book for changes to resource definition, resulting from specific enhancements to CICS.

- **Application load table (ALT)**

The program management component is re-engineered to such an extent that the virtual storage benefits previously offered by the ALT are superseded.

- **Nucleus load table (NLT)**

The NLT is made obsolete by the internal restructuring of CICS; there is no longer any need to optimize the working set.

- **Sign-on table (SNT)**



The SNT is obsolete because of the removal of CICS internal security. VSE/ESA Version 2 Release 4 continues to offer the VSE/ESA Interactive User Interface (IUI) security facilities through its Basic Security Manager (BSM), rather than through CICS. Note that resource-level security is not supported by the BSM.

- **Options on the system recovery table (SRT)**

Both the ROUTINE and PROGRAM options are obsolete. Both VSE cancel codes and OS/390 abend codes can be specified. User recovery program logic has to be provided using the new DFHSRP global user exit, XSRAB.

- **Ability to install transaction and program resource definitions using suffixed PCTs and PPTs**

These resources are installed only by using the CSD. This makes the use of a CSD mandatory.

However, the DFHPCT and DFHPPT table macros are retained for the sole purpose of migrating existing PCT and PPT definitions to the CSD.

- **Support for the existing DFHPCT and DFHPPT TYPE=GROUP definitions for functional grouping of CICS facilities**

This function is superseded by the RDO group concept and supplied definitions. For example, the DFHPCT TYPE=GROUP, FN=ISC entry is superseded by the RDO group, DFHISC.

- **Ability to install VTAM terminal resource definitions using a suffixed TCT**

These resources are installed only by using the CSD. However, unlike PCT and PPT resource definitions, particular TCT definitions are not eligible for online definition, for example, sequential devices. Table definition is retained for these resources only. You can still use the DFHTCT TYPE=GROUP, GROUP=*groupname* definition to assign a set of resources to a specific RDO group.

## The file control table (FCT)

File control resources such as VSAM files and data tables are supported online. Although the FCT continues to be supported for VSAM files, data tables, and DAM files, you are strongly recommended to migrate your VSAM resource definitions to RDO, to take advantage of improvements in availability.

### Statement of intent

It is intended that CICS TS Release 1 will be the last release to support FCT definition of VSAM files.

## The program list table (PLT)

The DFHDELIM option applies to PLTP1 (startup) as well as to PLTSD (shutdown).

---

## Using the changed system and resource definitions

- 1 Review your existing tables and startup job streams, using the sample SIT and job streams provided.

There are many changes to the system initialization table in this release. Use the sample SITs and job streams as a base, documented in the VSE/ESA Version 2 Release 4 library.

- 2 If you used the SYSGEN feature to apply installation modifications to modules that are withdrawn (listed in Table 23 on page 147) review these and implement the required functions through formal customization methods, such as global user exits.
- 3 If you have locally modified the source code of CICS management modules in the past, review these modifications and implement the functions that they provide by using the global user exit XPI.
- 4 If you use system initialization overlays or the SIMODS system initialization parameter, convert your implementations to run during the second stage of PLTPI processing.
- 5 Consider using the SPI for master terminal functions. The old programmable interface to the master terminal program (DFHEMTA) is still supported but the documentation provided for earlier releases.
- 6 Consider migrating your VSAM resource definitions to RDO:
  - Specify MIGRATE=YES on the DFHFCT TYPE=INITIAL macro
  - Assemble and catalog the tables
  - Use the DFHCSDUP utility to migrate the resource definitions to the CSD
  - Assemble the FCT using the MIGRATE=COMPLETE option. This produces an FCT load module.
  - Assign a set of resources to a specific RDO group with the following definition: DFHFCT TYPE=GROUP, GROUP=*groupname*
- 7 Migrate existing PCT and PPT definitions to the CSD:
  - Modify existing PCT and PPT definitions to include the appropriate DFHPCT TYPE=GROUP, GROUP=*groupname* definitions
  - Assemble and catalog the tables.
  - Use the DFHCSDUP utility to migrate the definitions to the CSD.
- 8 If you currently use the sign-on table (SNT) explicitly, ensure you have the services of an external security manager to sign on. This could be the BSM provided in VSE/ESA Version 2 Release 4, or a vendor-supplied ESM.
- 9 Replace any user recovery code implemented by the ROUTINE and PROGRAM options on the system recovery table (SRT) by the new DFHSRP global user exit, XSRAB.
- 10 Migrate terminal resource definitions to the CSD:

**Note:** Particular TCT definitions, such as sequential devices, are not eligible for online definition. Table definition is retained only for these resources.

- Specify MIGRATE=YES on the DFHTCT TYPE=INITIAL macro
- Assemble and catalog the tables
- Use the DFHCSDUP utility to migrate the resource definitions to the CSD
- Assemble the TCT using the MIGRATE=COMPLETE option. This produces an TCT load module.
- Assign a set of resources to a specific RDO group with the following definition: DFHFCT TYPE=GROUP, GROUP=*groupname*

**11** Upgrade your CSD. The following procedure applies whether or not you are using the VSE/ESA Interactive User Interface (IUI).

- Use the CSD created during the base installation of VSE/ESA and CICS as the foundation for your new CSD.
- From your **existing** CSD, use the DFHCSDUP COPY command to move your own resource groups to the new CSD.
- Use the DFHCSDUP APPEND command to move and extend LSTs.
- Migrate all unsupported table definitions to the CSD using the DFHCSDUP MIGRATE facilities.
- Upgrade your existing resource definitions and check that these definitions work as expected. Then, add definitions for the new functions.
- Perform all the above operations using facilities (CEDA and DFHCSDUP) provided with CICS TS Release 1.



---

## Chapter 29. Creating and installing resources using EXEC CICS CREATE

This chapter describes the EXEC CICS CREATE command for directly creating and installing resources in a running CICS region. It contains the following topics:

- “Overview of the EXEC CICS CREATE command”
- “Benefits of the EXEC CICS CREATE command” on page 154
- “Changes to CICS externals” on page 154
- “Security” on page 156
- “Coexistence” on page 156
- “Examples of the CREATE command” on page 156

---

### Overview of the EXEC CICS CREATE command

The EXEC CICS CREATE command allows you to add CICS resources into a CICS region without reference to the CICS system definition (CSD) file. It also allows you to write applications for administering the running CICS system.

The EXEC CICS CREATE command can be used in a PLT program to install CICS resources during a CICS cold start. This could potentially improve the performance of a cold start by reducing the number of resources read from the CSD file by GRPLIST processing.

The EXEC CICS CREATE command for non-CSD resources allows you to develop a whole range of systems management applications that were not previously possible. The following could be considered:

- Customization of resource definition processing by developing your own interfaces and using EXEC CICS CREATE to install the resources you define.
- Introducing a form of autoinstall for those resources for which CICS does not provide it. An EXEC CICS CREATE command can be used to install automatically such resources just before their first use, or when processing a NOTFND response.
- Installing resources in remote CICS regions, by using the EXEC CICS CREATE command in programs executed by distributed program link. However, note that each command performs an implicit syncpoint, which is not allowed for DPL programs unless they are linked to with the SYNCONRETURN option.
- Improving the performance of cold start by selectively adding, in a PLT program, only those resources needed for a particular run of CICS.
- Making certain resources available only at certain times, by scheduling timed transactions that execute EXEC CICS CREATE and EXEC CICS DISCARD commands. This could be especially useful for removing files from CICS regions for batch updates, and then reinstalling them after the batch update is complete.

---

## Benefits of the EXEC CICS CREATE command

Although the resource definition online (RDO) function of CICS has provided significant productivity improvement over earlier macro-based forms of resource definition, its use is becoming constrained by the need to include all the CICS definitions in the CICS CSD file. Although sharing of the CSD file between CICS regions is possible, it is sometimes impracticable, especially when large numbers of CICS regions are in use.

There is an increasing need for a single point of configuration, installation, and distribution, by which the resources for all CICS regions are administered from a single point. The EXEC CICS CREATE command for non-CSD resources fulfils the resource installation requirement. The CREATE function, by enabling the installation of resource definitions, makes it possible for IBM CICSplex System Manager (CICSplex SM), or a similar system management product, to provide configuration and installation of CICS resources from a management focal point.

---

## Changes to CICS externals

The EXEC CICS CREATE command for non-CSD resources causes the following changes to CICS externals:

- “Changes to the system programming interface (SPI)”
- “Changes to problem determination” on page 156

## Changes to the system programming interface (SPI)

CICS introduces the following new SPI commands:

- CREATE CONNECTION
- CREATE FILE
- CREATE LSRPOOL
- CREATE MAPSET
- CREATE PARTITIONSET
- CREATE PARTNER
- CREATE PROFILE
- CREATE PROGRAM
- CREATE SESSIONS
- CREATE TERMINAL
- CREATE TRANCLASS
- CREATE TRANSACTION
- CREATE TYPETERM

These commands all include the following:

- Support of the RESP, RESP2, NOEDF and NOHANDLE options
- Support of command and resource security checking
- Support by the command interpreter (CECI)
- Support by the execution diagnostic facility (CEDF)
- Support by the CICS translator

However, the ATTRIBUTES option in these commands are **not** parsed by the CICS translator. The attributes are analyzed at execution time instead, and an INVREQ condition with an appropriate RESP2 value is raised if the attributes are invalid. The syntax of the ATTRIBUTES string is identical to that for the

operands of the DEFINE command that can be coded for the CSD update utility, DFHCSDUP, as described in the *CICS Resource Definition Guide*.

The commands are available for execution any time during the final phase of CICS initialization. This means they can be used in programs specified in the second section of the program load table for postinitialization (PLTPI) as well as during CICS execution.

The function of each CREATE command is to add a resource to the current CICS region, but without any reference to data stored on the CSD. If the resource already exists, it is replaced if possible. An implicit syncpoint is performed if the resource is successfully added.

## Syntax

The syntax of all CREATE commands is common:

```
EXEC CICS CREATE resource(data-value)
        ATTRLEN(data-value) ATTRIBUTES(data-area)
```

## Options

The options for all the CREATE commands are:

### resource(name)

The type and name of the resource to be installed in the CICS region.

### ATTRIBUTES(data-area)

A data-area containing a list of attributes to be assigned to the resource name when it is added. These attributes must be coded as a single character string using the syntax shown in the *CICS System Programming Reference* manual. For information on specifying these attributes, see the description of the DEFINE command for each resource in the *CICS Resource Definition Guide*. This character string is not interpreted until the CREATE resource command is executed.

### ATTRLEN(data-value)

The halfword data-value containing the length of the data area supplied in the ATTRIBUTES option. This can be any value from 0 through 32767.

## Conditions

The conditions for all the CREATE commands are:

<b>Condition</b>	<b>RESP2 value</b>	<b>Meaning</b>
ILLOGIC	2	The command cannot be executed because an earlier connection or terminal pool definition has not yet been completed.
INVREQ		The attributes specified in ATTRIBUTES are invalid.
INVREQ	200	The command was executed in a program that has been linked to from a remote system that has not specified the SYNCONRETURN option, or has been linked to locally and is defined with EXECUTIONSET=DPLSUBSET.
LENGERR	1	The length specified in ATTRLEN is negative.
NOTAUTH	100	The user associated with the issuing task is not authorized to use this command.
NOTAUTH	101	The user associated with the issuing task is not authorized to create this particular connection.

<i>Condition</i>	<i>RESP2 value</i>	<i>Meaning</i>
NOTAUTH	102	The user of the transaction issuing the CREATE command is not an authorized surrogate of the userid specified in the command.

## Changes to problem determination

No new messages are added to support EXEC CICS CREATE command for non-CSD resources, but existing RDO messages, which were previously only displayed on the screen of the CEDA transaction user, are now recorded on the CSMT transient data destination.

The EXEC CICS CREATE command also causes the back-translated CREATE command to be written to the CSDL transient data destination, for auditing the resources added to a CICS region.

### Trace

There are no new trace points in support of EXEC CICS CREATE command for non-CSD resources. However, the existing trace point X'00E1' is now also issued by DFHEIP upon entry and exit from the EXEC CICS CREATE command.

---

## Security

The EXEC CICS CREATE command is subject to command security, under control of the XCMD system initialization parameter, and also to resource security, under control of the appropriate *Xname* system initialization parameter for the resource being added. (See Chapter 7, "CICS security" on page 39 for information about command and resource security.) To distinguish the authority to create new CICS objects, rather than to just to INQUIRE or SET the attributes of existing objects, an ESM access level of ALTER is required to create an object. The authority to DISCARD CICS resources is also ALTER. This is in line with the usual ESM policy of requiring a higher authority to create and destroy objects than simply to update them.

Because ALTER is a higher authority than either READ or UPDATE, this implies that any user who is authorized to CREATE new objects is also authorized to INQUIRE and SET them.

---

## Coexistence

Exploitation of the EXEC CICS CREATE command to create resources that are not defined in any CSD could make those resource definitions incompatible with the CSD definitions used in CICS regions at earlier releases, even if the CSD is shared with those regions.

---

## Examples of the CREATE command

The following is an illustration of the command syntax for the CREATE PROGRAM command in a COBOL program:

```
EXEC CICS CREATE PROGRAM(PROGRAM-NAME)
      ATTRIBUTES(PROGRAM-ATTRIBUTES)
      ATTRLEN(PROGRAM-ATTRLEN)
      END-EXEC.
```



The following is an illustration of the COBOL working storage declarations for the PROGRAM, ATTRIBUTES, and ATTRLEN of the above CREATE command:

```
01 PROGRAM-NAME      PICTURE X(8).
01 PROGRAM-ATTRLEN   PICTURE S9(4) COMPUTATIONAL.
01 PROGRAM-ATTRIBUTES.
  05 FILLER          PICTURE X(9) VALUE 'LANGUAGE(''.
  05 LANGUAGE        PICTURE X(9).
  05 FILLER          PICTURE X(2) VALUE ') ' '.
  05 FILLER          PICTURE X(9) VALUE 'RESIDENT(''.
  05 RESIDENT        PICTURE X(3).
  05 FILLER          PICTURE X(2) VALUE ') ' '.
  05 FILLER          PICTURE X(11) VALUE 'USESVCOPY(''.
  05 USELPCOPY       PICTURE X(3).
  05 FILLER          PICTURE X(2) VALUE ') ' '.
  05 FILLER          PICTURE X(7) VALUE 'STATUS(''.
  05 STATUS          PICTURE X(8).
  05 FILLER          PICTURE X(2) VALUE ') ' '.
  05 FILLER          PICTURE X(5) VALUE 'CEDF(''.
  05 CEDF            PICTURE X(3).
  05 FILLER          PICTURE X(2) VALUE ') ' '.
  05 FILLER          PICTURE X(13) VALUE 'DATALOCATION(''.
  05 DATALOCATION    PICTURE X(5).
  05 FILLER          PICTURE X(2) VALUE ') ' '.
  05 FILLER          PICTURE X(8) VALUE 'EXECKEY(''.
  05 EXECKEY        PICTURE X(4).
  05 FILLER          PICTURE X(2) VALUE ') ' '.
  05 FILLER          PICTURE X(13) VALUE 'EXECUTIONSET(''.
  05 EXECUTIONSET    PICTURE X(9).
  05 FILLER          PICTURE X(2) VALUE ') ' '.
```



## Appendix A. System initialization parameters

This appendix covers:

- “New system initialization parameters”
- “Changed system initialization parameters” on page 167
- “Obsolete system initialization parameters” on page 169

### New system initialization parameters

Table 24 summarizes the new system initialization parameters.

<i>Table 24 (Page 1 of 8). New system initialization parameters</i>	
Parameter	Description
AIEXIT={ <b>DFHZATDX</b>  name}	Indicates the autoinstall module to be used when autoinstalling VTAM terminals and APPC connections and sessions.
AILDELAY={ <b>0</b>  hhmmss}	Indicates the delay period to elapse after a session between CICS and an autoinstalled terminal ends and the terminal entry is deleted.
AIQMAX={ <b>100</b>  number}	Indicates the maximum number of VTAM terminals and APPC connections that can be queued concurrently for autoinstall.
AIRDELAY={ <b>700</b>  hhmmss}	Indicates the delay period to elapse after emergency restart before autoinstall entries that are not in session are deleted.
AUXTR={ <b>OFF</b>  ON}	Sets the auxiliary trace destination on or off and controls whether CICS system trace, user trace, or exception trace entries are written to the auxiliary trace data set.
AUXTRSW={ <b>NO</b>  ALL NEXT}	Sets the trace autoswitch facility on or off.
CDSASZ={ <b>0K</b>  number}	Specifies the size of the CICS DSA (CDSA). The default size is 0, indicating that the DSA size can change dynamically. A non-zero value indicates that the DSA size is fixed.
CHKSTRM={ <b>CURRENT</b>  NONE}	Sets terminal storage-violation checking on or off. Can be used as an alternative to the CICS-supplied transaction, CSFE.
CHKSTSK={ <b>NONE</b>  ALL CURRENT}	Sets task storage-violation checking on or off at start up. Can be used as an alternative to the CICS-supplied transaction, CSFE.
CMDPROT={ <b>YES</b>  NO}	Allows or inhibits validation of start addresses of storage referenced as output parameters on EXEC CICS commands.
CMDSEC={ <b>ASIS</b>  ALWAYS}	Indicates whether you want CICS to honor the CMDSEC option specified on a transaction's resource definition.

Table 24 (Page 2 of 8). New system initialization parameters

Parameter	Description
CONFDATA={ <b>SHOW</b>  HIDETC}	Indicates whether CICS is to hide user data that might otherwise appear in CICS trace entries or in dumps that contain the RAIA. CONFDATA applies to initial input data received on a VTAM RECEIVE ANY operation, on an MRO link, FEPI screens, and RPLAREAs.
CONFTEXT={ <b>NO</b>  YES}	Indicates whether CICS is to prevent VTAM from tracing user data.
CSDACC={ <b>READWRITE</b>  READONLY}	Indicates the type of CSD access permitted in this CICS region. Effective on a cold start only.
CSDBUFND=number	Indicates the number of buffers to be used for CSD data. Effective on a cold start only and can only be used with the CSDLRNO=NONE parameter.
CSDBUFNI=number	Indicates the number of buffers to be used for the CSD index. Effective on a cold start only and can only be used with the CSDLRNO=NONE parameter.
CSDFRLOG=number	Identifies the journal to be used for forward recovery of CSD. Effective on a cold start only. On a warm or emergency restart, file resource definitions are recovered from the global catalog. CSDFRLOG can only be used if the CSDRECOV=ALL parameter is also coded.
CSDJID={ <b>NO</b>  number}	Identifies the journal to be used for automatic journaling of file requests against the CSD. Effective on a cold start only. On a warm or emergency restart, file resource definitions are recovered from the global catalog.
CSDLRNO={ <b>1</b>  number NO NONE}	Indicates whether the CSD is to be associated with an LSRpool. Effective on a cold start only. On a warm or emergency restart, file resource definitions are recovered from the global catalog.
CSDRECOV={ <b>NONE</b>  ALL BACKOUTONLY}	Indicates whether the CSD is a recoverable file. Effective on a cold start only. On a warm or emergency restart, file resource definitions are recovered from the global catalog.
CSDSTRNO={ <b>2</b>  number}	Indicates the number of concurrent requests that can be processed against the CSD. Effective on a cold start only. On a warm or emergency restart, file resource definitions are recovered from the global catalog.
CWAKEY={ <b>USER</b>  CICS}	Indicates the storage key for the CWA if CICS storage protection is active (STGPROT=YES).
DFLTUSER={ <b>CICSUSER</b>  userid}	Associates the ESM userid with the security attributes to be used for all terminal users who are not explicitly signed on.
DISMACP={ <b>YES</b>  NO}	Disables any transaction that terminates abnormally with an ASRDabend. Is ineffective if an ASRDabend is handled by an activeabend exit.

Table 24 (Page 3 of 8). New system initialization parameters

Parameter	Description
DSALIM={ <b>5M</b>  number}	Specifies the upper limit of the total amount of storage within which CICS can allocate the individual DSAs that reside below 16MB.
DSHIPIDL={ <b>020000</b>  hhmmss}	Indicates the minimum time that an <i>inactive</i> shipped terminal definition must remain installed in this region. When the timeout delete mechanism is invoked, only those shipped definitions that have been inactive for longer than the specified time are deleted.
DSHIPINT={ <b>120000</b>  0 hhmmss}	Indicates the interval between invocations of the timeout delete mechanism. The timeout delete mechanism removes any shipped terminal definitions that have not been used for longer than the time specified.
DTRTRAN={ <b>CTRX</b>  name}	Identifies the transaction definition required for dynamic transaction routing. CTRX is the name of the new transaction available to define a dynamic transaction routing definition. The default sample, CTRX, is provided in the CSD group, DFHISC.
DUMPDS={ <b>AUTO</b>  A B}	Specifies the transaction dump data set to be opened during CICS initialization.
DUMPSW={ <b>NO</b>  NEXT}	Indicates whether CICS is to switch automatically to the next dump data set when the first becomes full.
ECDSASZE={ <b>OK</b>  number}	Specifies the size of the ECDSA. The default size is 0 indicating that the DSA size can change dynamically. A non-zero value indicates that the DSA size is fixed.
EDSALIM={ <b>20M</b>  number}	Specifies the upper limit of the total amount of storage within which CICS can allocate individual extended DSAs above 16MB.
EODI={ <b>EO</b>  xx}	Indicates the end-of-data indicator for input from sequential terminals.
ERDSASZE={ <b>OK</b>  number}	Specifies the size of the ERDSA. The default size is 0 indicating that the DSA size can change dynamically. A non-zero value indicates that the DSA size is fixed.
ESDSASZE={ <b>OK</b>  number}	Specifies the size of the ESDSA. The default size is 0 indicating that the DSA size can change dynamically. A non-zero value indicates that the DSA size is fixed.
ESMEXITS={ <b>NOINSTLN</b>  INSTLN}	Specifies whether installation data is to be passed via the RACROUTE interface to the ESM manager for use in exits written for the ESM.
EUDSASZE={ <b>OK</b>  number}	Specifies the size of the EUDSA. The default size is 0 indicating that the DSA size can change dynamically. A non-zero value indicates that the DSA size is fixed.
FEPI={ <b>NO</b>  YES}	Indicates whether FEPI is required. FEPI=YES causes CICS to start the FEPI transaction, CSZI.
FSSTAFF={ <b>NO</b>  YES}	Specifies whether function-shipping START affinity support is required in an AOR.
GNTRAN={ <b>CESF</b>  transaction-id}	Identifies the transaction CICS is to invoke when a user's terminal-timeout period expires.

Table 24 (Page 4 of 8). New system initialization parameters

Parameter	Description
INITPARM=(pgname_1='parmstring_1', ... ,pgname_n= 'parmstring_n')	Passes parameters to applications that use the ASSIGN INITPARM command. The area giving access to the parameters is specified by the ASSIGN INITPARM command.
INTTR={ <b>ON</b>  OFF}	Indicates whether the internal CICS trace destination is to be activated at system initialization and controls whether CICS system trace, user trace, or exception trace entries are written to the internal trace table.
MN={ <b>OFF</b>  ON}	Indicates whether CICS monitoring is to be activated at initialization.
MNCONV={ <b>NO</b>  YES}	Specifies whether conversational tasks are to have separate performance class records produced for each pair of terminal control I/O requests. Any clock active at the time such a performance class record is produced is stopped immediately before the record is written (this includes user-defined clocks).
MNEXC={ <b>OFF</b>  ON}	Indicates whether the monitoring exception class is to be activated during system initialization. The monitoring status class is recorded in the CICS global catalog for use during warm and emergency restarts.
MNFREQ={ <b>0</b>  hhmmss}	Specifies the interval for which CICS automatically produces transaction performance class records for any long-running transaction. The monitoring frequency value is recorded in the CICS global catalog for use during warm or emergency restarts.
MNPER={ <b>OFF</b>  ON}	Indicates whether the monitoring performance class is to be activated during CICS initialization. The monitoring performance class status is recorded in the CICS global catalog for use during warm or emergency restarts.
MNSYNC={ <b>NO</b>  YES}	Specifies whether a transaction performance class record should be produced when a transaction takes an implicit or explicit syncpoint. No action is taken for syncpoint rollbacks. The monitoring syncpoint status is recorded in the CICS global catalog for use during warm or emergency restarts.
MNTIME={ <b>GMT</b>  LOCAL}	Specifies whether the time stamp fields in performance class monitoring data will be returned to an application using the EXEC CICS STATISTICS MONITOR( <i>taskno</i> ) command in GMT or local time. The monitoring time value is recorded in the CICS global catalog for use during warm and emergency restarts.
MROFSE={ <b>NO</b>  YES}	Specifies whether you want to extend the lifetime of the long-running mirror task to the end of the user task.
MSGCASE={ <b>MIXED</b>  UPPER}	Indicates whether the message domain is to display messages in upper or lowercase.

Table 24 (Page 5 of 8). New system initialization parameters

Parameter	Description
NEWSIT={ <b>NO</b>  YES}	Specifies whether CICS is to load the specified SIT and enforce the use of all system initialization parameters (modified by any system initialization parameters provided via PARM, SYSIPT, or the system console).
PARMERR={ <b>INTERACT</b>  IGNORE ABEND}	Indicates the action CICS should take if incorrect system initialization overrides are detected during initialization. It only affects subsequent SIT overrides.
PGAICTLG={ <b>MODIFY</b>  NONE ALL}	Indicates whether autoinstalled program definitions are to be cataloged. When CICS is running, you can catalog autoinstalled programs dynamically using an EXEC CICS SET SYSTEM or CEMT SET SYSTEM command.
PGAEXIT={ <b>DFHPGADX</b>  NAME}	Identifies the autoinstall exit program. When CICS is running, you can catalog autoinstalled programs dynamically using an EXEC CICS SET SYSTEM or CEMT SET SYSTEM command.
PGAIPGM={ <b>INACTIVE</b>  ACTIVE}	Indicates the state of the program autoinstall function at initialization. When CICS is running, you can catalog autoinstalled programs dynamically using an EXEC CICS SET SYSTEM or CEMT SET SYSTEM command.
PLPISEC={ <b>NONE</b>  CMDSEC RESSEC ALL}	Indicates whether or not CICS is to perform command security checking for PLT programs during CICS initialization.
PLTPIUSR=userid	Identifies the user ID CICS is to use for security checking for PLT programs that run during CICS initialization. All PLT programs run under the authority of the specified user ID, which must be authorized to all the resources referenced by the programs, as defined by the PLTPISEC parameter.
PRTYAGE={ <b>32768</b>  value}	Sets the priority aging algorithm for incrementing the priority of a task in milliseconds.
PRVMOD={name (name,name...name)}	Identifies the modules not to be used from the SVA.
PSDINT={ <b>0</b>  hhmmss}	Specifies how long VTAM is to hold sessions in a recovery-pending state if CICS fails.
PVDELAY={ <b>30</b>  number}	Specifies how long entries can remain in signed-on-from lists for connections for which persistent verification is specified in a CONNECTION resource definition.
RDSASZE={ <b>0K</b>  number}	Specifies the size of the read-only DSA (RDSA). The default size is 0 indicating that the DSA size can change dynamically. A non-zero value indicates that the DSA size is fixed.
RENTPGM={ <b>PROTECT</b>  NOPROTECT}	Specifies whether you want CICS to allocate the RDSA and ERDSA from read-only key-0 protected storage.
RESP={ <b>FME</b>  RRN}	Specifies the type of request CICS terminal control receives from logical units (function management end (FME) or reached recovery node (RRN)).

Table 24 (Page 6 of 8). New system initialization parameters

Parameter	Description
RESSEC={ <b>ASIS</b>  ALWAYS}	Specifies whether or not you want CICS to honor the RESSEC option specified on a transaction's resource definition.
RUWAPool={ <b>NO</b>  YES}	specifies the option for allocating a storage pool the first time an LE program runs in a task.
SDSASZE={ <b>OK</b>  number}	Specifies the size of the SDSA. The default size is 0 indicating that the DSA size can change dynamically. A non-zero value indicates that the DSA size is fixed.
SEC={ <b>YES</b>  NO}	Indicates the level of external security required. YES for full security or NO for none.
SECPRFX={ <b>NO</b>  YES}	Indicates whether or not CICS should prefix the resource names in any authorization requests to a full-function ESM, with a prefix corresponding to the ESM user ID for the CICS region. The prefix to be used is obtained from DFHIRP.
SNSCOPE={ <b>NONE</b>  CICS VSEIMAGE }	Specifies whether or not a user ID can be signed on to CICS more than once, within the scope of a single CICS region or a single VSE image.
SPCTR={{ <b>(1,2</b>  1,[,2],[,3])  ALL OFF}	Sets the level of tracing for all CICS components used by a transaction, a terminal, or both, selected for special tracing.
SPCTRxx={{ <b>(1,2</b>  1,[,2],[,3])  ALL OFF}	Sets the level of tracing for a particular CICS component used by a transaction, a terminal, or both, selected for special tracing.
STATRCD= <b>OFF</b>  ON	Sets the statistics recording status at initialization. Status is recorded in the CICS global catalog for use during warm or emergency restarts. Statistics collected are written to a DMF data set.
STGPROT={ <b>NO</b>  YES}	Activates storage protection in the CICS region. If STGPROT=NO is specified, storage protection is inactive and CICS runs in a single storage key.
STGRCVY={ <b>NO</b>  YES}	Indicates whether CICS should attempt to recover from a storage violation. Either way, CICS continues unless you specify that CICS should terminate (in the dump table).
STNTR={ <b>1</b>  (1 [,2],[,3]) ALL OFF}	Indicates the level of standard tracing required for CICS as a whole. Up to 3 levels of tracing can be selected, but some CICS components do not have trace points at all these levels.
STNTRxx={ <b>1</b>  (1 [,2],[,3]) ALL OFF}	Indicates the level of standard tracing required for a particular CICS component. The component is identified by the value of xx in the keyword. One STNTR parameter is coded for each component that is to be traced.
SYDUMAX={ <b>999</b>  number}	Indicates the number of system dumps that can be taken for each dump table entry. If this number is exceeded, subsequent dumps are suppressed.



Table 24 (Page 7 of 8). New system initialization parameters

Parameter	Description
SYSTR={ <b>ON</b>  OFF}	Controls the master system trace flag, and obtains entries of CICS system activity. Entries are written to all active trace destinations.
TCSACTN={ <b>NONE</b>  UNBIND}	Indicates the action terminal control should take if the terminal control shutdown wait threshold expires. Applies to VTAM terminals (including LU Type 6.2 single-session APPC terminals) but not to VTAM intersystem connections (LU Type 6.1 and LU Type 6.2 parallel connections). Use with the TCSWAIT parameter.
TCSWAIT={ <b>4</b>  number NO NONE 0}	Specifies the CICS terminal control shutdown threshold. The wait threshold is the time (during shutdown) that CICS terminal control allows to pass before it considers terminal shutdown to be hung.
TCTUAKEY={ <b>USER</b>  CICS}	Specifies the storage key for the TCTUAs if storage protection is active. TCTUAKEY is ignored if storage protection is inactive.
TCTUALOC={ <b>BELOW</b>  ANY}	Indicates where TCTUAs should be stored. Above 16MB is the preferred location.
TRDUMAX={ <b>999</b>  number}	Specifies the number of transaction dumps that may be taken for each dump table entry. If this number is exceeded, subsequent dumps are suppressed.
TRTABSZ={ <b>16</b>  number-of-kilobytes}	Identifies the size of the internal trace table. The CICS trace table is allocated in virtual storage above 16MB <b>before</b> the ECDSA and the EUDSA.
TRTRANSZ={ <b>40</b>  number-of-kilobytes}	Indicates the size of the transaction dump trace table.
TRTRANTY={ <b>TRAN</b>  all}	Identifies the trace entries to be copied from the internal trace table to the transaction dump trace table.
TSMGSET={ <b>4</b>  number}	Identifies the number of entries for which dynamic storage is allocated for storing pointers to records put into the temporary storage message set.
UDSASZE={ <b>0K</b>  number}	Specifies the primary extent size of the UDSA.
USERTR={ <b>ON</b>  OFF}	Sets the master trace flag on or off. If the trace flag is off, the user trace facility is disabled and CICS ignores EXEC CICS ENTER TRACENUM commands.
USRDELAY={ <b>30</b>  number}	Specifies the maximum time that an eligible user ID and its attributes are to be retained in the user table if the user ID is unused.
VTPREFIX={\ <b>\</b>  character}	Specifies the first character to be used for the terminal identifiers (termids) of autoinstalled virtual terminals.
XAPPC={ <b>NO</b>  YES}	Specifies whether ESM security can be used when establishing APPC sessions.
XCMD={ <b>NO</b>  name YES}	Specifies whether CICS is to perform command security checking. Checking is only performed if SEC=YES is also specified, and the CMDSEC=YES option is specified on the TRANSACTION resource definition.

Table 24 (Page 8 of 8). New system initialization parameters

Parameter	Description
XRF <del>SOFF</del> ={ <b>NOFORCE</b>  FORCE}	Specifies whether all users signed on to the active CICS region are to remain signed on following a takeover. Applicable only if XRF=YES is also coded.
XRFSTME={ <u>5</u>  decimal-value}	Specifies the timeout delay interval for users who are still signed on when an XRF takeover occurs.
XUSER={ <b>NO</b>  YES}	Specifies whether or not CICS is to perform surrogate user checks.

## Changed system initialization parameters

Information relating to the following system initialization parameters is changed as a result of the addition of new options, or because of new functions.

Table 25 (Page 1 of 2). Changed system initialization parameters

Parameter	Function
APPLID={ <b>DBDCCICS</b>   applid (generic-applid,specific-applid)}	Generic and specific applids cannot be the same
# BMS={({MINIMUM STANDARD  <b>FULL</b> }[ ,COLD] # [, <b>UNALIGN</b>  ALIGN]} [, <b>DDS</b>  NODDS]) # # # #	You cannot define 3270 display units and printers as DEVICE(SCSPRINT) on the RDO TYPETERM definition. Full or standard BMS is only needed if XRF is active, and MESSAGE for RECOVNOTIFY is specified on any TYPETERM definitions. BMS=FULL is needed for the report controller.
DATFORM={ <b>MMDDYY</b>  DDMMYY YYMMDD}	CICS maintains the external date display in the form 0CYDDDD, where C=0 represents years 19xx, C=1 for years 20xx, and so on. YY=year of the century; and DDD=day of the year. CICS converts the date to the format you specify.
DBP={1\$ 2\$ xx YES}	You can no longer code DBP=NO.
DUMP={ <b>YES</b>  NO}	SDUMPs are produced if DUMP=YES is specified (unless suppressed by system dump table options or system defaults). If DUMP=NO is specified, SDUMPs are suppressed.
DUMPDS={ <b>AUTO</b>  A B}	AUTO opens the dump data set for all types of start.
FCT={ <b>YES</b>  xx NO}	Effective on a cold start only.
GMTRAN={ <b>WELCOME TO CICS</b>  'text'}	Change to default text.
GRPLIST={ <b>DFHLIST</b>  name (name[,name2] [,name3][,name4])}	Up to four lists of resource definition groups can be named. If more than one group list is specified, the later group lists are concatenated into the first group list. Any duplicate SITs in later lists override those in earlier lists.
ICVTSD={ <b>5000</b>  number}	The range is now 0 through 5000 milliseconds.
JSTATUS=RESET	JSTATUS only resets the journal status of data sets defined on disk (JTYPE=DISK1 DISK2), with the JOUROPT=AUTOARCH option <b>not</b> coded. JSTATUS does not apply if automatic archiving is active.
MXT={ <b>5</b>  number}	The range is now 1 through 999. MXT refers only to user tasks and does not include system tasks.
PRINT={ <b>NO</b>  YES PA1 PA2 PA3}	When YES, PA1, PA2, or PA3 is specified, transaction CSPP is initiated which in turn invokes program DFHP3270. Transaction and programs are now defined in the CSD group, DFHHARDC. For 3270 and LUTYPE2 LUs, the resources defined in group DFHVTAMP are required.
SRT={ <b>YES</b>  NO xx}	CICS now issues ESTAEX macros to intercept program checks.

Table 25 (Page 2 of 2). Changed system initialization parameters

Parameter	Function
START={{ <b>AUTO</b>  COLD STANDBY LOGTERM}},[ALL]}	Use the new LOGTERM option as an alternative to a full emergency restart when the previous run terminates in an uncontrolled shutdown. LOGTERM is only available if the CICS system log is defined on disk data and if XRF=NO is coded.
SUFFIX=xx	SIT=xx can now be coded either in PARM or SYSIPT.
SYSIDNT={{ <b>CICS</b>  name}}	SYSIDNT can now only be updated on a cold start.
TBEXITS={{[name1]},[name2]},[name3]},[name4]}	The name option is no longer valid. If TBEXITS are not needed, you can either omit the parameter altogether or code it as TBEXITS=(,,).
TCP={{ <b>YES</b>  NO}}	Suffix numbers are no longer valid.
TD={{({3} decimal-value-1)} ({{3} decimal-value-2})}}	The range for decimal-value-1 is now 1 through 32767. More buffers may be allocated than are specified, depending on the CI size defined in the intrapartition data set.
# TS={{[COLD]},{0{3} decimal-value-1}}, # {3 decimal-value-2}} # #	The range for decimal-value-1 is now 3 through 32767. More buffers may be allocated than are specified, depending on the CI size defined in the auxiliary temporary storage data set.
VTPREFIX={\ character}	Specifies the first parameter to be used for the termids of autoinstalled virtual terminals. Virtual terminals are used by the External Presentation Interface (EPI) and terminal emulator function of the CICS Client products.
<b>Note:</b> Checking is performed for XDCT, XFCT, XJCT, XPCT, XPPT, and XTST only if YES is specified for the SEC system initialization parameter, and RESSEC(YES) is specified on the appropriate resource definition. These parameters can be coded in the SIT, PARM, and SYSIPT only.	
XDCT={{ <b>NO</b>  name YES}}	Specifies whether CICS is to perform transient data resource security checking.
XFCT={{ <b>NO</b>  name YES}}	Specifies CICS is to perform file resource security checking, and, optionally, the ESM resource class name in which the file resource security profiles have been defined.
XJCT={{ <b>NO</b>  name YES}}	Specifies whether CICS is to perform journal resource security checking, and, optionally, the ESM resource class name in which the journal resource security profiles have been defined.
XPCT={{ <b>NO</b>  name YES}}	Specifies whether CICS is to perform started transaction resource security checking, and, optionally, the ESM resource class name in which the started task security profiles have been defined.
XPPT={{ <b>NO</b>  name YES}}	Specifies whether CICS is to perform program resource security checking, and, optionally, the ESM resource class name in which the program resource security profiles are defined.
XTST={{ <b>NO</b>  name YES}}	Specifies whether CICS is to perform temporary storage security checking, and, optionally, the ESM resource class name in which the temporary storage security profiles are defined.

## Obsolete system initialization parameters

The following SIT parameters are obsolete;

<i>Table 26. Obsolete system initialization parameters</i>	
<b>Parameter</b>	<b>Reason for removal</b>
ABDUMP	Removed because of the restructuring of dumping facilities.
ALEXIT	Removed because of restructured PLT processing.
ALT	Removed because the application load table is obsolete.
AMXT	New dispatcher algorithms, and the new transaction class resource definitions remove the need to limit the number of active tasks
AUTINST	Replaced by individual terminal autoinstall parameters AIXIT, AILDELAY, AIRDELAY, and AIQMAX.
BFP	Removed because the built-in-functions program is obsolete.
CMXT	Superseded by the RDO TRANCLASS resource definition.
COBOL2	Removed because the VS COBOL II environment is automatically enabled.
DIP	Removed because DFHDIP is now always loaded as part of the CICS nucleus.
DTB	Removed because all dynamic log records are now spilled to main storage allocations in the extended read-only DSA (ERDSA).
EXEC	Removed because a command-level environment is mandatory
EXITS	Removed because the user-exit interface is now always initialized.
EXTSEC	Removed because CICS internal security support is obsolete, and the use of a security manager is compulsory.
FERS	Removed because the Facility Recognition System is obsolete.
ICVS	Removed because the stall purge mechanism is superseded.
LESTG	Removed because of improved DSA management above 16MB.
MONITOR	Replaced by the new MN, MNCONV, MNEXC, MNFREQ, MNPER, MNSYNC, and MNTIME system initialization parameters for monitoring.
NLT	Removed because the nucleus load table is obsolete.
OPNDLIM	Removed because enhancements to VTAM make it unnecessary.
PCDUMP	Removed because of the restructuring of dump facilities.
PCT	Removed because all transaction and profile resources must be defined using RDO.
PGSIZE	Removed because the CICS virtual page size is fixed at 4096 bytes.
PLI	Removed because DOS PL/I is no longer supported.
PPT	Removed because all program resources must be defined using RDO.
SCS	Removed because storage cushion sizes are determined automatically.
SIMODS	Removed because system initialization overlays are obsolete.
SVD	Superseded by the dump table.
TRACE	Replaced by the new tracing system initialization parameters, TRTABSZ, TRTRANSZ, TRTRANTY, and USERTR.
XRFTRSZ	Ignored because the XRF trace table now resides above 16MB and has a fixed size of 64KB.
ZCP	Removed because ZCP programs cannot be suffixed.



# Bibliography

## CICS Transaction Server for VSE/ESA Release 1 library

<b>Evaluation and planning</b>	
<i>Release Guide</i>	GC33-1645
<i>Migration Guide</i>	GC33-1646
<i>Report Controller Planning Guide</i>	SC33-1941
<b>General</b>	
<i>Master Index</i>	SC33-1648
<i>Trace Entries</i>	SC34-5556
<i>User's Handbook</i>	SX33-6101
<i>Glossary (softcopy only)</i>	GC33-1649
<b>Administration</b>	
<i>System Definition Guide</i>	SC33-1651
<i>Customization Guide</i>	SC33-1652
<i>Resource Definition Guide</i>	SC33-1653
<i>Operations and Utilities Guide</i>	SC33-1654
<i>CICS-Supplied Transactions</i>	SC33-1655
<b>Programming</b>	
<i>Application Programming Guide</i>	SC33-1657
<i>Application Programming Reference</i>	SC33-1658
<i>Sample Applications Guide</i>	SC33-1713
<i>Application Migration Aid Guide</i>	SC33-1943
<i>System Programming Reference</i>	SC33-1659
<i>Distributed Transaction Programming Guide</i>	SC33-1661
<i>Front End Programming Interface User's Guide</i>	SC33-1662
<b>Diagnosis</b>	
<i>Problem Determination Guide</i>	GC33-1663
<i>Messages and Codes Vol 3 (softcopy only)</i>	SC33-6799
<i>Diagnosis Reference</i>	LY33-6085
<i>Data Areas</i>	LY33-6086
<i>Supplementary Data Areas</i>	LY33-6087
<b>Communication</b>	
<i>Intercommunication Guide</i>	SC33-1665
<i>CICS Family: Interproduct Communication</i>	SC33-0824
<i>CICS Family: Communicating from CICS on System/390</i>	SC33-1697
<b>Special topics</b>	
<i>Recovery and Restart Guide</i>	SC33-1666
<i>Performance Guide</i>	SC33-1667
<i>Shared Data Tables Guide</i>	SC33-1668
<i>Security Guide</i>	SC33-1942
<i>External CICS Interface</i>	SC33-1669
<i>XRF Guide</i>	SC33-1671
<i>Report Controller User's Guide</i>	SC33-1940
<b>CICS Clients</b>	
<i>CICS Clients: Administration</i>	SC33-1792
<i>CICS Universal Clients Version 3 for OS/2: Administration</i>	SC34-5450
<i>CICS Universal Clients Version 3 for Windows: Administration</i>	SC34-5449
<i>CICS Universal Clients Version 3 for AIX: Administration</i>	SC34-5348
<i>CICS Universal Clients Version 3 for Solaris: Administration</i>	SC34-5451
<i>CICS Family: OO programming in C++ for CICS Clients</i>	SC33-1923
<i>CICS Family: OO programming in BASIC for CICS Clients</i>	SC33-1671
<i>CICS Family: Client/Server Programming</i>	SC33-1435
<i>CICS Transaction Gateway Version 3: Administration</i>	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
Administration 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

---

## 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.

Licenseses 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:

AFP	AS/400	C/370
CICS	CICS/ESA	CICS/VSE
CICS/VM	CICS/400	CICSplex
DB2	DFSORT	Enterprise Systems Architecture/390
ES/9000	ESA/390	IBM
IMS	IMS/ESA	Language Environment
MVS/ESA	OS/2	OS/390
SQL/DS	System/36	System/370
System/38	System/390	VSE/ESA
VTAM	3090	

---

# Index

## A

- access methods 8
- Advanced Function (AFD) printers 7
- APPC (LU6.2) session security 46
- APPC persistent verification
  - changes to CICS externals 62
  - overview 61
  - requirements 61
  - using 61
- application programming
  - changes to 65
  - DFHEICAL macro 65
  - EXCI 77
  - language restrictions 65
  - macro-level 66
  - obsoleted function 65
  - references to control blocks 67
  - roadmap to improvements 63
  - summary of improvements to 2
- autoinstall
  - advantages of 113
  - cataloging 113
  - changes to CICS externals 115
  - mapsets 114
  - model definitions 114
  - using 115
- automatic journal archiving
  - benefits of 117
  - DFHJACD data set 117
  - DFHJACDU utility 118
  - DFHXJCC user-replaceable module 117
  - DFHXJCO user-replaceable module 117
  - using 119
- availability
  - roadmap for improvements 11
  - storage enhancements 21

## B

- benefits
  - of EXEC CICS CREATE command 154
- BSM (basic security manager) 39

## C

- cancel start requests
  - overview 83
  - using cancel function 83
- CDSA 22
- CEDA programmable interface 139
- CEMT enhancements for statistics 129

- central functions component 5
- changes to SPI 154
  - EXEC CICS CREATE command 154
- CICS
  - control blocks, access to 65
  - internal macros 67
  - list of domains 14
  - macro invocations 67
  - new statistics function 132
  - nucleus 18
  - running systems in parallel 68
  - system autoinstall 113
- CICS default user, security 45
- CICS internal code
  - accessing domains 13
  - CICS nucleus 18
  - list of CICS domains 14
  - partition layout 18
  - restructuring of 13
  - virtual storage constraint relief 17
  - VTAM terminal control modules 17
- CICS key 23
- CICS spooler interface, API enhancements 7
- CICS state data
  - changes to CICS externals 105
  - overview of access to 105
- CICS system definition (CSD)
  - in RDO 136
  - sharing between CICS versions 137
- command security checking 53
- common work area (CWA) 160
- CPI-C
  - using 69
- CREATE command 154
- CSSM transient data queue 129
- CSSN transient data queue 129
- CSTT transaction removed 129
- CWA (common work area) 160

## D

- data collection, in monitoring 121
- database products 9
- default user, security 45
- devices
  - DFH\$IWUP utility program 141
  - DFH0STAT sample programs 132
  - DFHCSDUP utility, invoking 136
  - DFHDELIM option, PLT changes 149
  - DFHEICAL macro 65
  - DFHJACD data set 117

- DFHJACDU journaling utility 118
- DFHMCT control table 122
- DFHMCT macro changes
  - TYPE=EMP 123
  - TYPE=FINAL 123
  - TYPE=INITIAL 122
  - TYPE=RECORD 123
- DFHSG macros 147
- DFHSTUP, changes to 132
- DFHXJCC user-replaceable module 117
- DFHXJCO user-replaceable module 117
- DFLTUSER system initialization parameter 45
- DLBL statements removed 129
- DMF
  - data sets for monitoring 121
  - use in collecting statistics 129
- DSALIM, system initialization parameter 22
- dual-purpose resource definition, extensions to 137
- dump, in monitoring 124
- dynamic storage area (DSA)
  - CDSA 23
  - definition of types 22
  - ECDSA 23
  - EUDSA 23
  - overview of 21
  - Specifying values for 22
  - UDSA 23
- dynamic transaction routing (DTR) 111

## E

- ECDSA 22
- EDSALIM, system initialization parameter 22
- ERDSA 22
- ESDSA 22
- EUDSA 22
- EXCI
  - CALL interface 78
  - changes to CICS externals 79
  - EXEC CICS LINK 78
  - overview 77
  - using 78
- EXEC CICS commands in statistics 131
- EXEC CICS CREATE command 154
  - CONNECTION 154
  - FILE 154
  - for non-CSD resources 153
  - JOURNALMODEL 154
  - LSRPOOL 154
  - MAPSET 154
  - PARTITIONSET 154
  - PARTNER 154
  - PROFILE 154
  - PROGRAM 154
  - SESSIONS 154
  - TDQUEUE 154

- EXEC CICS CREATE command (*continued*)
  - TERMINAL 154
  - TRANCLASS 154
  - TRANSACTION 154
  - TYPETERM 154
- exit programming interface (XPI) 101
- exit-handling changes 92
- extended storage usage 21
- external security
  - DFLTUSER system initialization parameter 45

## F

- file control table (FCT), changed 149
- forward recovery utilities, support for 144
- front end programming interface (FEPI)
  - changes to CICS externals 73
  - using 71

## H

- hardware prerequisites 5

## I

- in-doubt resources
  - DFH\$IWUP utility program 141
  - resolving 141
- Interactive Computing and Control facility (ICCF) 5
- internal macros 67
- intersystem communication
  - requirements 10
- intersystem session queue
  - overview 35
  - using 36
  - XZIQUE global user exit 35

## L

- Language Environment for VSE/ESA 9
- languages supported 9
- link security user ID 47

## M

- macro-level application programs 66
- mapsets 114
- model definitions in autoinstall 114
- monitoring
  - changes to CICS externals 125
  - CICS internal code changes 122
  - controlling 124
  - exception class records 122
  - improvements to 121
  - obsolete function in 121
  - performance class records 121
  - system initialization parameters, new 123



monitoring (*continued*)  
using dump 124  
using new functions in 125  
using trace 124

MRO  
requirements 10  
security 47

## N

new PLT phase 92  
non-CSD resources  
EXEC CICS CREATE command 153

## O

operating system requirements 5

## P

partition layout 18  
persistent sessions in VTAM 29  
prerequisites  
software and hardware 5  
preset terminal security 44  
processors, supported 6  
program list table (PLT), changes 149  
programming languages  
restrictions in 65  
programming languages supported by CICS TS  
Release 1 9  
protection of storage 24

## R

RDSA 22  
recovery from DTB failures 143  
recovery of VSAM files 143  
REDO records 144  
report controller, support of RSL values 58  
requirements, hardware 5  
resource definition  
changes. 148  
file control table (FCT) 149  
obsolete function 148  
program list table (PLT), changes 149  
resource definition online (RDO)  
CEDA programmable interface 139  
CSD 136  
DFHCSDUP utility 136  
dual-purpose resource definitions 137  
improvements to 135  
installing single resources 139  
removing installed definitions 138  
sharing CSD between CICS versions 137  
resource definitions, removing 138

resource security checking 48

## S

SCS, system initialization parameter, obsolete 22  
SDSA 22  
security  
APPC (LU6.2) session security 46  
basic security manager, BSM 39  
browse security, report controller 58  
category 1 transactions 41  
CICS default user 45  
command security checking 53  
console operators, effects on 56  
DFLTUSER system initialization parameter 45  
link user ID, default 47  
multiregion (MRO) security 47  
non-terminal security 40  
performance 52  
PLT programs at startup, non-terminal security 42  
preset terminal security 44  
printer security, report controller 58  
report security, report controller 57  
resource definition changes 40  
resource security 48  
resource security for DPL commands 52  
RESSEC keyword, new 52  
RESSEC system initialization parameter 52  
security domains 39  
sign-on, changes to 44  
signon in remote regions 47  
started transactions, non-terminal security 41  
surrogate user checking 42  
system initialization parameters 56  
terminal user security 43  
transaction categories 41  
transaction security, changes 40  
trigger-level transactions, non-terminal security 42  
types of security 39  
user domain 40  
vendor products 40  
XS domain 40  
sequential terminal support 7  
shared data tables  
description of 85  
using 86  
sign-on, changes to 44  
single node persistent sessions 29  
statistics  
changes to 129  
changes to CICS externals 133  
changes to data collection 130  
CSSM and CSSN withdrawn 129  
CSTT transaction removed 129  
DFH0STAT sample programs 132  
DFHSTUP changes 132

- statistics (*continued*)
  - DLBL statements removed 129
  - end-of-day 130
  - interval 130
  - new EXEC CICS commands 131
  - new XSTOUT parameters 132
  - obsolete function 129
  - requested 130
  - STATRCD system initialization parameter 131
  - unsolicited 130
  - using new function 132
- STATRCD system initialization parameter 131
- storage
  - CDSA 22, 23
  - changes to externals 26
  - CICS key 23
  - ECDSA 22, 23
  - enhancements 21
  - ERDSA 22
  - ESDSA 22
  - EUDSA 22, 23
  - keys 23
  - RDSA 22
  - SDSA 22
  - UDSA 22, 23
  - user key 23
  - using protection 24
- storage requirements 22
- supported devices 6
- supported languages 9
- surrogate user checking, security 42
- SYSGEN macros 147
- system definition
  - changes to 147
  - SYSGEN macros 147
- system initialization
  - changes to parameters 159
  - new monitoring parameters 123
  - STATRCD statistics parameter 131
- system management
  - dynamic transaction routing 111
  - roadmap to improvements 109
- system programming
  - global user exits 91
  - roadmap to improvements 89
  - XPI 101
- system types supported 7

## T

- terminal control modules 17
- trace 156
- trace, in monitoring 124
- transaction management, enhancements to 145
- transaction security 40

## U

- UDSA 22
- user key 23

## V

- virtual storage constraint relief 17, 21
- VSAM
  - recovery of files 143
- VSE/POWER, prerequisites 5
- VTAM persistent sessions
  - changes to externals 32
  - clean-up and recovery 30
  - comparison with XRF 31
  - overview of 29
  - predatory takeover 31
  - releases supported 8
  - sessions not retained 31
  - terminal control modules 17
  - unbinding sessions 30
  - using 32

## X

- XPI
  - overview 101
- XPI (exit programming interface) 101
- XRF, compared to VTAM 31
- XRFRSZ SIT parameter, obsolete 59
- XSTOUT, new parameters for 132
- XZIQUE global user exit 35

---

## **Sending your comments to IBM**

**CICS® Transaction Server for VSE/ESA™**

### **Release Guide**

#### **GC33-1645-00**

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.

GC33-1645-00



*Spine information:*



CICS TS for VSE/ESA

Release Guide

*Release 1*