

CICS Transaction Server for z/OS
5.6

Installing CICS



Note

Before using this information and the product it supports, read the information in [Product Legal Notices](#).

This edition applies to the IBM® CICS® Transaction Server for z/OS®, Version 5 Release 6 (product number 5655-Y305655-BTA) and to all subsequent releases and modifications until otherwise indicated in new editions.

© **Copyright International Business Machines Corporation 1974, 2023.**

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

Contents

About this PDF.....	vii
Chapter 1. Planning for installation.....	1
Program Directories.....	2
Prerequisites for installing CICS TS.....	2
CICS Explorer planning.....	3
CICSplex SM setup checklist and worksheets.....	3
MVS installation and setup checklist.....	4
System worksheet.....	5
CMAS worksheet.....	6
MAS worksheet.....	7
WUI server worksheet.....	8
IVP planning worksheet.....	9
Chapter 2. CICS Explorer installation.....	11
Configuring the FTP server for CICS Explorer.....	11
Chapter 3. Installing CICS TS Developer Trial.....	13
Chapter 4. Installing CICS TS VUE (Value Unit Edition).....	15
Chapter 5. Installing CICS TS.....	17
Installing the CICS TS activation module.....	17
Installing the CICS TS base component from CBPDO.....	18
Installing the CICS TS base component from ServerPac.....	19
Installing CICS TS using DFHISTAR.....	19
Checking the output from the installation jobs.....	22
Editing the DFHISTAR job.....	22
Creating RACF profiles for the CICS Transaction Server data sets.....	45
Running the DFHISTAR job.....	46
Running the installation jobs.....	46
Checking the output from the installation jobs.....	52
Postinstallation activities.....	52
Chapter 6. Activating CICS Transaction Server.....	59
Chapter 7. Setting up the MVS environment for CICS.....	61
Authorizing the CICS and CICSplex SM libraries.....	61
Authorizing CICS regions to access MVS resources.....	62
Protecting CICS load module data sets.....	62
Authorizing access to a CICS region z/OS Communications Server ACB.....	63
Authorizing the region user ID to access category 1 transactions.....	64
Authorizing access to an SMSVSAM server.....	64
Authorizing access to MVS log streams.....	64
Authorizing access to z/OS UNIX System Services.....	66
Activating RACF resource classes.....	69
Defining the default CICS user ID to RACF.....	69
Installing CICS-required modules in the MVS linklist.....	69
CICS- and CICSplex SM-supplied modules required in the MVS linklist.....	69

CICSplex SM API linklist modules.....	72
Installing the REXX function package.....	72
Modules of other MVS products in the MVS linklist.....	73
Defining CICS as an MVS subsystem.....	73
Coexistence with automation programs and other CICS releases.....	74
Noting IEASYSxx values for CICSplex SM.....	74
Specifying each CMAS correctly in IEASYSxx.....	75
Coding the IEFSSNaa MVS subsystem initialization member.....	76
The SYS1.PARMLIB(BPXPRMxx) parameters.....	77
The console message-handling facility.....	79
EXCI pipe allocation.....	83
Installing the CICS SVCs.....	84
Using more than one version of the CICS Type 3 SVC.....	86
Installing newest release CICS Type 3 SVC and DFHIRP modules supplied through maintenance.....	87
MRO between different CICS releases with a changed SVC number.....	87
Selecting the high-performance option.....	88
Defining EYU9XENF as a TSO/E authorized command.....	89
Defining CICS regions as applications to SNA.....	89
Defining specific APPL definitions and APPL parameters to SNA.....	90
SNA version and release level indicator.....	91
Defining cross-domain services when using SNA.....	91
SNA LU alias facility.....	92
Defining SNA requirements for CICSplex SM.....	95
Installing CICS modules in the MVS link pack area.....	98
Preparing to install CICS modules in the MVS link pack area.....	98
Space requirements for CICS modules in the MVS link pack area.....	111
Defining the CICS LPA library to your MVS.....	112
Installing CICS modules in the LPA.....	112
Controlling the use of modules from the MVS link pack area.....	113
Sample DFHCSDUP job to specify USELPACOPY(YES).....	117
Installing CICSplex SM modules in the MVS link pack area.....	117
Space requirements for CICSplex SM modules.....	118
Installing CICSplex SM modules into the LPA.....	118
Applying maintenance to LPA modules.....	119
CICSplex SM modules eligible for the MVS link pack area.....	119
Defining CICS IPCS exit control data to MVS.....	120
The DFHIPCSP CICS exit control data.....	120
Preparing to use the CICSplex SM IPCS tools	120
Updating BLSCECT.....	121
Updating library allocations.....	121
MVS Program properties table entries.....	121
Configuring MVS workload management for CICS systems.....	122
Matching CICS performance parameters to service policies.....	122
Implementing MVS automatic restart management.....	123
MVS cross-system MRO definitions.....	123
MVS ASREXIT - SYMREC Authorization Exit.....	124
Setting up VSAM RLS support.....	125
Defining the main coupling facility lock structure.....	125
Defining coupling facility cache structures and cache sets.....	126
Preparing for RLS access.....	126
Defining sharing control data sets.....	127
Authorizing CICS user IDs for VSAM RLS support.....	128
Adding new parameters to SYS1.PARMLIB(IGDSMSxx).....	128
Establishing new procedures for VSAM RLS support.....	128
Activating the coupling facility structures.....	129
Console messages.....	129
Defining the logger environment for CICS.....	129

Requirements planning and checklist for the logger environment.....	130
Setting up the environment for CICS log manager.....	133
Coupling facility or DASD-only?.....	135
Coupling facility log streams.....	136
DASD-only log streams.....	150
Analyzing system logger data (SMF Type 88 records).....	154
Managing auxiliary storage.....	155
Unicode data conversion by z/OS.....	156
Applying service to CICS Transaction Server for z/OS.....	156
Software maintenance strategy for CICS TS and its embedded components.....	157
Load library secondary extents.....	158
The CICS TS-supplied SMP/E procedure.....	158
APARs and PTFs.....	159
Mapping CICS releases from APARs and PTFs to CICS external versions.....	159
Applying service to CICS modules.....	163
Applying service to CICSplex SM.....	164
CICS Transaction Server for z/OS-supplied SMP/E procedure.....	164
Applying PTFs to CICSplexes.....	164
Applying service to CICSplex SM modules.....	165
Chapter 8. Getting ready to run CICS.....	167
Tailoring the CICS-supplied skeleton jobs.....	167
The CICS installation libraries.....	167
Running DFHISTAR.....	168
CICSplex SM postinstallation members.....	168
Jobs for creating the CICS data sets.....	170
Data set naming conventions.....	170
DFHCOMDS job for common data sets.....	171
DFHDEFDS job for CICS region data sets.....	172
DFHCMACI job for creating the messages data set.....	173
Defining the sample applications data sets.....	173
CICS resource definitions for CICSplex SM.....	176
Adding CICS support for programming languages.....	178
Installing Language Environment support.....	178
Enabling TCP/IP in a CICS region.....	181
Using TCP/IP in a CICS region.....	182
Installing MRO support.....	182
Installing the modules DFHIRP and DFHCSVC in the LPA.....	183
Installation requirements for XCF/MRO.....	183
Defining z/OS images as systems in an XCF sysplex.....	185
Enabling MRO for CICS startup.....	185
Adding communications support between systems.....	185
Verifying Java components checklist.....	191
Defining DL/I support.....	192
Program specification blocks (PDIR).....	192
Adding remote DL/I support.....	193
Enabling REXX for CICS.....	194
Chapter 9. Renewing a Developer Trial license.....	195
Chapter 10. Verifying the CICS installation.....	197
Preparing to run the IVPs.....	197
Creating and defining activities for the IVP jobs.....	197
Reviewing and defining IVP security.....	198
Specifying system initialization parameters for the IVP jobs.....	200
Resources for the CICS messages facility, CMAC.....	200
The CICS startup procedure, DFHSTART.....	201

Verify batch job, DFHIVPBT.....	203
Verify interactive job, DFHIVPOL.....	204
Verifying shared data tables support.....	213
Example verification of shared data tables.....	214
Verifying the CICS-DBCTL interface.....	218
The IMS installation requirements for the DFHIVPDB job.....	218
The DFHIVPDB job steps.....	219
Testing the CICS Db2 environment.....	223
Chapter 11. Default CICS resource definitions for CICSplex SM.....	225
Notices.....	235
Index.....	241

About this PDF

This PDF describes how to install CICS Transaction Server for z/OS, Version 5 Release 6 . It is intended for system programmers who are responsible for installing and tailoring CICS and CICSplex® SM.

It describes:

- The content of CICS TS and its delivery methods
- Prerequisites and planning for installation
- How to install CICS TS and CICSplex SM and verify the installation, before you go on to any further configuration and tailoring.

These tasks are done in conjunction with the *Program Directory*.

For details of the terms and notation used, see [Conventions and terminology used in CICS documentation](#) in IBM Documentation.

Date of this PDF

This PDF was created on 2025-12-02 (Year-Month-Date).

Chapter 1. Planning for installation

The IBM packages that are available for installing CICS TS are a system replace method called ServerPac and the Custom-Built Product Delivery Option (CBPDO).

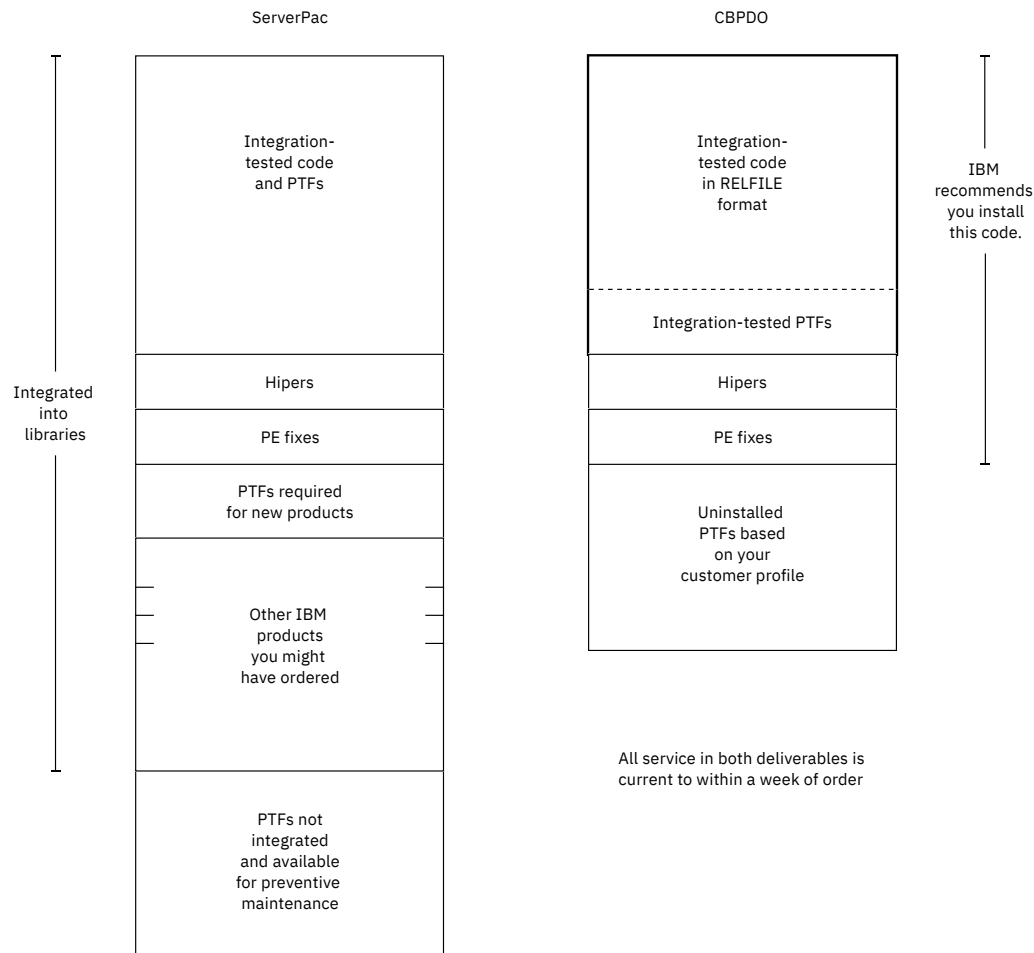


Figure 1. The content differences of ServerPac and CBPDO

ServerPac

The ServerPac can be installed through the z/OSMF Software Management task.

During the ServerPac installation, you configure your installation to map to your environment, such as data set naming, data set placement, catalog specification, and so on.

If you order ServerPac, you receive the following items:

- DVDs (ServerPac install only) or a downloadable package containing a complete, generated CICS TS system.
- Sample jobs to assist with the installation of the CICS TS product and service.

If you ordered the portable software instance of the ServerPac, you use the z/OSMF Software Management task to acquire the files, then customize and deploy them. See [Downloading a ServerPac in PSI format](#) for details.

CBPDO (Custom-Build Product Delivery Option)

CBPDO (Custom-Built Product Delivery Option) is an entitled software delivery package that consists of uninstalled products and unintegrated service. There is no dialog program to help you install it, as there is with ServerPac. You must use SMP/E to install.

If you order CBPDO, you can download one logically stacked SMP/E RELFILE package that contains all the base elements. Depending on your customer profile, you receive uninstalled service. You also receive the following items:

- Sample jobs to help you install CICS TS and service.
- The CBPDO Memo to Users Extension and program materials that describe your CBPDO order.

Installing

You can install CICS TS by using the standard IBM installation methods, or by using the CICS-supplied DFHISTAR jobs. For information about installing by using the DFHISTAR jobs, see [“Installing CICS TS using DFHISTAR”](#) on page 19.

Program Directories

The Program Directory is a document provided with CICS Transaction Server. It contains information about the materials and procedures for installing each component of CICS TS.

To install any of the editions of CICS Transaction Server for z/OS, you need the Program Directory for both components.

Table 1. Program Directories required for installation. Program Directories required for installation	
Edition	Program Directories required
CICS Transaction Server for z/OS - base	<ul style="list-style-type: none">• Program Directory: base for CICS TS for z/OS V5.6• Program Directory: activation module for CICS TS for z/OS V5.6 - base
CICS Transaction Server for z/OS - Developer Trial	<ul style="list-style-type: none">• Program Directory: base for CICS TS for z/OS V5.6• Program Directory: activation module for CICS TS for z/OS V5.6 - Developer Trial
CICS Transaction Server for z/OS - Value Unit Edition	<ul style="list-style-type: none">• Program Directory: base for CICS TS for z/OS V5.6• Program Directory: activation module for CICS TS for z/OS V5.6 - Value Unit Edition

Prerequisites for installing CICS TS

Whether you choose the CBPDO method or the ServerPac method, the hardware requirement is the same but the software requirements can vary.

For the latest and most complete information about the hardware and software requirements (required and optional) and the service that you must apply to install CICS TS, see [Detailed system requirements](#) and the supplied Program Directories (see [Detailed system requirements](#)).

For information about optional supported software and the service that you must apply to ensure that CICS TS installs correctly, see [Program Directories](#).

z/OS UNIX file system

CICS TS for z/OS, Version 5.6 requires the OMVS address space must be active in full-function mode, otherwise the installation of this function fails as the CICS TS installation includes z/OS UNIX files.

In the set of installation jobs, some initial jobs create the z/OS UNIX directories and files, based on the z/OS UNIX directory structure for CICS as illustrated in [Figure 2 on page 3](#). Run these jobs before any of the normal DFHINST n jobs. The user ID running these jobs requires superuser authority.

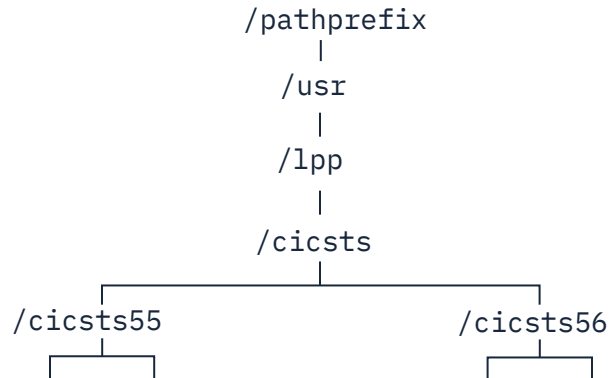


Figure 2. The z/OS UNIX directory structure for CICS

1. If you normally maintain additional SMP/E target zone libraries to apply service, you can also create additional directories at the /cicsts56 level to create the z/OS UNIX equivalent. See the DFHIHFSA job for more information.
2. The /pathprefix in [Figure 2 on page 3](#) is optional.

CICS Explorer planning

CICS Explorer® is a system management tool that is delivered as part of CICS Transaction Server. It provides an easy-to-use way to manage one or more CICS systems. It is based on an Eclipse platform and provides a view of some CICSplex SM functions in a CICS Transaction Server, and a platform for the integration of CICS tools.

See [Downloading and starting CICS Explorer in the CICS Explorer product documentation](#).

CICSplex SM setup checklist and worksheets

Use the aids provided to help you install and setup CICSplex SM.

Checklist

Use the checklist as a guide to your progress as you set up or revise the configuration of your CICSplex SM components. The checklist is for use with a CICS Transaction Server for z/OS system and all of the components you can install on it.

You perform some of the tasks on the MVS checklist only once for your CICSplex SM environment, and for other tasks perform them once for each component. See the 'Where to obtain information' column for a reference to information about how to perform each task.

The order of items in the checklist is a suggested order for performing the installation and setup steps. However, you might find that, particularly if you are modifying your CICSplex SM environment, a different order is more practical.

Worksheets

Use the worksheets as a record of the names and locations of components and data sets. Copy the worksheets as required.

The worksheets contain, in some cases, more than one line for a type of CICSplex SM component. You might have fewer or more than shown of that type of component.

MVS installation and setup checklist

A checklist is provided to help you to install and setup MVS.

Component			What you must do	Values to note	Where to obtain information
Components are indicated: CM =CMAS, M =MAS, W =WUI server					
CM	M	W	Make note of SYS1.PARMLIB(IEASYSxx) values for this MVS system	APF= CMD= LNK= LNKAUTH= LPA= MAXUSER= NSYSLX= PROG= RSVNONR= RSVSTRT= SMF= SYSNAME=	“Noting IEASYSxx values for CICSplex SM” on page 74
CM			Update number of linkage indexes in IEASYSxx	NSYSLX value	“Specifying each CMAS correctly in IEASYSxx” on page 75
CM	M	W	Update IEAAPFxx or PROGxx to authorize SEYUAUTH	IEAAPFxx or PROGxx member Library name	“Authorizing the CICS and CICSplex SM libraries” on page 61
CM			Verify SEYULINK is authorized	LNKAUTH= value Library name	“Authorizing the CICS and CICSplex SM libraries” on page 61
CM			Update linklist with SEYULINK	LNKLSTxx member Library name	“CICS- and CICSplex SM-supplied modules required in the MVS linklist” on page 69
	M	W	Update LPA list with index.SEYULPA Optional library. Can be populated below.	LPALSTxx member Library name	“Installing CICSplex SM modules into the LPA” on page 118
CM	M	W	Use your ESM to protect CICSplex SM libraries	As required by your ESM	See Creating profiles for the CICSplex SM data sets and Defining the CICSplex SM started tasks
CM			Define security for the CMAS startup procedures	Procedure names	Activating security parameters CICSplex SM
CM			Create z/OS Communications Server application definition for each CMAS	SYS1.VTAMLST major node member Application name(s)	“Step 1: Creating a z/OS Communications Server application definition for a CMAS” on page 96
CM			Define cross-domain resources for each CMAS	SYS1.VTAMLST members	“Step 2: Defining cross-domain resources for a CMAS” on page 97
CM	M	W	Edit DFHISTAR for postinstallation members	Edited member	“ CICSplex SM postinstallation members” on page 168
CM	M	W	Run edited DFHISTAR member to generate POST installation members.	sysproc.XEYUINST output library name	“ CICSplex SM postinstallation members” on page 168
	M	W	(Optional) Install LPA modules	Installed usermod name	“Installing CICSplex SM modules into the LPA” on page 118

Component			What you must do	Values to note	Where to obtain information
CM			Create CICSplex SM data repository	dsindex.EYUDREP. cmasname	Creating the CICSplex SM data repository
CM			Create CICSplex SM system parameter member for each CMAS	Modified EYUCMSOP parameter member(s)	Preparing to start a CMAS
	M		Edit CICSplex SM system parameter member for each MAS	Modified EYULMSOP parameter member(s)	Preparing to start a z/OS MAS
		W	Edit CICSplex SM system parameter member for each WUI	Modified EYUWUIOP parameter member(s)	Web User Interface server initialization parameters
CM			Edit CICS SIT parameters for each CMAS	Modified parameter member(s)	CMAS-related CICS system initialization parameters
	M		Edit CICS SIT parameters for each MAS	Modified parameter member(s)	z/OS MAS-related CICS system initialization parameters
		W	Edit CICS SIT parameters for each WUI	Modified parameter member(s)	Reviewing CICS system initialization parameters for the WUI
CM			Create the CICS data sets and the data repository for each CMAS	Modified EYUCMSDS member	Preparing to start a CMAS
CM			Install CMAS startup procedure (EYUCMASP sample procedure)	Installed procedure member	Preparing to start a CMAS
	CM		Create CMAS-to-CMAS links using the WUI or the EYU9XDBT utility	CMAS names Target APPLID Target CICS SYSID	CMAS to CMAS link definitions - CMTCMDEF
CM			Create CICSplex definition using the WUI or the EYU9XDBT utility	CICSplex name	CICSplex definitions - CPLEXDEF
	M		Create all MAS definitions using the WUI or the EYU9XDBT utility	MAS name(s)	Creating a CICS system definition
	M		Start the MAS	Message EYUXL0007I LMAS Phase II initialization complete	Preparing to start a z/OS MAS
	M		Shut down the MASs using the WUI CICS regions view - stops CICS	Message EYUXL0016I MAS shutdown complete	Stopping and restarting management of a CICS system

System worksheet

A worksheet is provided to help you to setup your CICSplex SM system.

System: _____

Subsystem ID: _____

SNA applid: _____

	Name:	SNA applid:	CICS-SYSID:
CMAS:	_____	_____	_____
WUI server:	_____	_____	_____
MAS:	_____	_____	_____
MAS:	_____	_____	_____
MAS:	_____	_____	_____
CMAS:	_____	_____	_____
WUI server:	_____	_____	_____
MAS:	_____	_____	_____
MAS:	_____	_____	_____
MAS:	_____	_____	_____
CMAS:	_____	_____	_____
WUI server:	_____	_____	_____
MAS:	_____	_____	_____
MAS:	_____	_____	_____
MAS:	_____	_____	_____

CMAS worksheet

A worksheet is provided to help you to setup your CMAS.

System: _____

CMAS name: _____

CICS sysid: _____

SNA applid: _____

SYS1.PARMLIB(IEASYSxx) values:

APF= _____

LNK= _____

LNKAUTH= _____

NSYLSX= _____

PROG= _____

DSN added to member IEAAPFxx or PROGxx: when LNKAUTH=APFTAB: (when LNKAUTH=LNKLST, no DSNhere)

DSN added to member LNKLSTxx:

SNA mode table node name:

SYS1.VTAMLST start list (ATCSTRxx):

6 CICS TS for z/OS: Installing CICS

SYS1.VTAMLST configuration list (ATCCONxx): _____

SYS1.VTAMLST applications member: _____

SYS1.VTAMLST cross-domain member: _____

SNA definitions; node names: _____

	Name	SNA applid	CICS sysid
MAS:	_____	_____	_____
MAS:	_____	_____	_____
MAS:	_____	_____	_____

Installation materials library: _____SDFHINST

Modified DFHISTAR (postinstallation) member: _____

DFHINST exec output library: _____XDFHINST

CICSplex SM data repository DSN: _____

CICS resource definition tables output DSN: _____

CICS CSD DSN: _____

Modified EYUCMSOP member: _____

CICS SIT parameters member: _____

Modified EYUCMSDS member: _____

CMAS startup procedure member: _____

Links to other CMASs:

CMAS name:	SNA applid:	CICS sysid:	Protocol:
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

Links to MASs:

MAS name:	SNA applid:	CICS sysid:	Protocol:
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____
_____	_____	_____	_____

MAS worksheet

A worksheet is provided to help you to set up your MAS.

MVS system: _____

CICSplex name: _____

CPSM name: _____ MAS CPSM name: _____

CICS sysid: _____ MAS CICS sysid: _____
 SNA applid: _____ MAS SNA applid: _____
 MAS type: _____ AOR, TOR

 DSN added to member IEAAPFxx or PROGxx: when _____ .SEYUAUTH
 LNKAUTH=APFTAB: (when LNKAUTH=LNKLST, no
 DSN here)
 DSN added to member LPALSTxx: _____ .SEYULPA
 SYS1.VTAMLST start list (ATCSTRxx): _____
 SYS1.VTAMLST configuration list (ATCCONxx): _____
 SYS1.VTAMLST applications member: _____
 Installation materials library: _____ .SDFHINST
 Modified DFHISTAR (postinstallation) member: _____
 DFHINST exec output library: _____ .XDFHINST
 LPA module (usermod) name: _____
 CICS resource definition tables output DSN: _____
 CICS CSD DSN: _____
 Modified EYULMSOP DSN (member): _____
 CICS SIT parameters DSN (member): _____
 Link from CMAS:

CMAS name:	SNA applid:	CICS sysid:	Protocol:
_____	_____	_____	_____

WUI server worksheet

A worksheet is provided to help you to setup your WUI server.

MVS system: _____
 CICSplex name: _____
 CPSM name: _____ MAS CPSM name: _____
 CICS sysid: _____ MAS CICS sysid: _____
 SNA applid: _____ MAS SNA applid: _____

 DSN added to member IEAAPFxx or PROGxx: when _____ .SEYUAUTH
 LNKAUTH=APFTAB: (when LNKAUTH=LNKLST, no dsn
 here)
 DSN added to member LPALSTxx: _____ .SEYULPA
 SYS1.VTAMLST start list (ATCSTRxx): _____
 SYS1.VTAMLST configuration list (ATCCONxx): _____
 SYS1.VTAMLST applications member: _____
 Installation materials library: _____ .SDFHINST
 Modified DFHISTAR (postinstallation) member: _____
 DFHINST exec output library: _____ .XDFHINST

LPA module (usermod) name: _____

CICS resource definition tables output dsn: _____

CICS CSD DSN: _____

Modified EYUWUIOP DSN (member): _____

CICS SIT parameters DSN (member): _____

Link from CMAS:

CMAS name: _____ SNA applid: _____ CICS sysid: _____ Protocol: _____

IVP planning worksheet

A worksheet is provided to help you to define your IVP.

Table 2. IVP planning worksheet			
Component	DFHISTAR parameters	Your value	DFHISTAR defaults
Load library high-level qualifier for CICS	TINDEX.CICS.XTRAQUAL		CICSTS56.CICS
Load library high-level qualifier for CICSplex SM	TINDEX.CPSM.XTRAQUAL		CICSTS56.CPSM
Attributes of the CICS TS system data sets	DSINFO		CICSTS56 CICS41 3390
Data set name of the SCEECICS library	SCEECICS		SYS1.SCEECICS
Data set name of the SCEERUN library	SCEERUN		SYS1.SCEERUN
Data set name of the SCEERUN2 library	SCEERUN2		SYS1.SCEERUN2
Data set name of the SCEESAMP library	SCEESAMP		SYS1.SCEESAMP
CMAS name	CMASNAME		CMAS01
CMAS sysid	CMSSYSID		CM01
CMAS applid	CMASNAME		CMAS01
CICSplex name	CSYSPLEX		CSYPLX01
MAS name	CSYSNAME		CSYS01
MAS sysid	CSYSYSID		CS01
Web User Interface CICSplex name	WUIPLEX		WUIPCM01
Web User Interface applid	WUINAME		WUINCM01
Web User Interface sysid	WUISYSID		WU01

<i>Table 2. IVP planning worksheet (continued)</i>			
Component	DFHISTAR parameters	Your value	DFHISTAR defaults
TCP/IP host name	TCPIPHST		XXXXXXXX. XXXXXXXX. XXXXXXXX. XXXXXXXX
TCP/IP port for Web User Interface	TCPIPPRT		12345
Timezone	TIMEZONE		B

Chapter 2. CICS Explorer installation

You can install CICS Explorer a number of ways and in a number of locations, depending on the requirements of your organization. After installation, you must configure your host for CICS Explorer to access z/OS resources.

CICS Explorer is based on IBM Explorer for z/OS (z/OS Explorer), which supports these connection types to access z/OS resources:

FTP

The z/OS FTP connection allows you to work with z/OS resources in the z/OS perspective provided by z/OS Explorer.

z/OSMF

The z/OS® Management Facility (z/OSMF) requires z/OSMF to be installed and enabled in your z/OS operating system. Using a z/OSMF connection provides richer function, including viewing job output for running jobs. In z/OS Explorer, the z/OSMF connection also makes an FTP connection, to handle file transfers.

RSE

The Remote System Explorer (RSE) connection allows you to use the RSE perspective in the Eclipse Workbench, where you can connect to and work with a variety of remote systems. The RSE connection requires the IBM Explorer for z/OS Host Component to be installed on your host (or the IBM Developer for z/OS Server). You can download it from [Mainframe DEV Downloads](#).

To use CICS Explorer with a z/OS FTP or z/OSMF connection, you must [configure the FTP server for CICS Explorer](#) to access z/OS resources.

To use CICS Explorer with an RSE connection, configure your RSE server instead. For instructions, see [rse.env](#), RSE configuration file in IBM Explorer for z/OS V3.1.0 product documentation.

For instructions on how to install the CICS Explorer client, see [Downloading and starting CICS Explorer in the CICS Explorer product documentation](#).

Configuring the FTP server for CICS Explorer

If your CICS Explorer uses a z/OS FTP connection to access data sets, z/OS hierarchical file systems such as zFS, and the JES spool, you must configure the z/OS® Communications Server FTP server for the connection to work. For CICS Explorer to successfully obtain the required spool files, the FTP.DATA configuration file for the Server must specify JESINTERFACELEVEL 2.

About this task

This task describes how to check the JESINTERFACELEVEL in the FTP server.

Procedure

1. Start an FTP client and run a **quote stat** command.
2. Check the output to determine the JESINTERFACELEVEL. For example:

```
230 userid is logged on. Working directory is "/u/<userid>".
ftp> quote stat
211-Server FTP talking to host 10.10.10.10, port 1087
211-User: USERID Working directory: /u/<userid>
211-The control connection has transferred 1360 bytes
....
211-Truncated records will not be treated as an error
211-JESLRECL is 80
211-JESRECFM is Fixed
211-JESINTERFACELEVEL is 2
211-Server site variable JESTRAILINGBLANKS is set to TRUE
....
211 *** end of status ***
```

```
ftp> quit  
221 Quit command received. Goodbye.
```

3. Change the FTP.DATA configuration file for the Server to specify JESINTERFACELEVEL 2.

What to do next

After you configure the FTP server, you must configure the CICS Explorer connections to your CICS or CICSplex SM systems. See [Connecting CICS Explorer to CICS systems in the CICS Explorer product documentation](#).

Chapter 3. Installing CICS TS Developer Trial

You use the same methods to install CICS TS Developer Trial as you do CICS Transaction Server.

Make sure that you are familiar with the restrictions of CICS TS Developer Trial, described in [Developer Trial and Value Unit Edition](#).

In summary, you do the following:

1. Download CICS Developer Trial from [Shopz](#). This comes as the CICS TS base component and the Developer Trial activation module. Make sure that you have the [Program Directories](#) for both the base component and the Developer Trial activation module.
2. Using SMP/E, install both the base component and the activation module. For details, see [Installing CICS TS](#) and [Installing the CICS TS activation module in Installing](#).
3. Include the SDFHDEV activation library in the STEPLIB of your CICS JCL.
4. Restart your region. In the console view, message [DFHTI0200](#) confirms that you are running Developer Trial and shows the expiry date. This message is issued daily at midnight until the expiry date is reached.

To extend your use of CICS TS Developer Trial at the end of the usage period, see [Renewing a Developer Trial license](#).

You can upgrade from Developer Trial to a full version of CICS as the Monthly License Charge (MLC) base edition of CICS TS, **subject to your purchased entitlement**, without having to reinstall. For instructions, see [Upgrading from CICS TS Developer Trial](#).

Chapter 4. Installing CICS TS VUE (Value Unit Edition)

You use the same methods to install CICS TS VUE as you do CICS Transaction Server. There are slightly different steps to go through depending on whether you are installing CICS TS VUE on its own or you are moving your existing CICS TS MLC (Monthly License Charge) installation to VUE.

Moving an existing CICS TS MLC installation to VUE

In this scenario, do the following:

1. Download CICS VUE from [Shopz](#). This comes as the CICS TS base component and the VUE activation module. Make sure that you have the [Program Directories](#) for both the base component and the VUE activation module.
2. Using SMP/E, install only the activation module on top of your existing base product. You do not need to install the CICS TS base component. There is no license acceptance panel during this installation. For details, see [Installing the CICS TS activation module in Installing](#).
3. Modify your CICS JCL to replace the CICS TS MLC activation library (SDFHLIC) with the new SDFHVUE activation library in the STEPLIB. The SDFHVUE library must be APF-authorized. For details, see [Authorizing the CICS and CICSplex SM libraries](#). If you use coupling facility data table servers, temporary storage servers, region status servers, or named counter servers, you must also add the SDFHVUE library to the STEPLIB of the JCL for each of these servers.
4. Restart your region. In the console view, message [DFHTI0103](#) confirms that you are running VUE.

Installing CICS TS VUE

In this scenario, do the following:

1. Download CICS VUE from [Shopz](#). This comes as the CICS TS base component and the VUE activation module. Make sure that you have the [Program Directories](#) for both the base component and the VUE activation module.
2. Using SMP/E, install both the base component and the activation module. There is no license acceptance panel during this installation. For details, see [Installing CICS TS](#) and [Installing the CICS TS activation module in Installing](#).
3. Include the SDFHVUE activation library in the STEPLIB of your CICS JCL. The SDFHVUE library must be APF-authorized. For details, see [Authorizing the CICS and CICSplex SM libraries](#). If you use coupling facility data table servers, temporary storage servers, region status servers, or named counter servers, you must also add the SDFHVUE library to the STEPLIB of the JCL for each of these servers.
4. Restart your region. In the console view, message [DFHTI0103](#) confirms that you are running VUE.

Chapter 5. Installing CICS TS

You must install two components for any of the three editions of CICS Transaction Server for z/OS: the base component and the activation module that is specific to each edition.

CICS TS has two components:

- CICS TS for z/OS V5.6 - base component
- An activation module specific to the offering

These components can be installed in any order but they must both be installed before you can run the installation verification procedures, described in [Verifying the CICS installation](#).

Installing the CICS TS activation module

You install an activation module for the CICS offering that you require. The activation module and the base component can be installed in any order.

Before you begin

Create an SMP/E environment for the components in your CICS Transaction Server for z/OS, Version 5 Release 6 offering. This environment can be shared between the CICS TS for z/OS V5.6 - base component and the activation module.

If you are using more than one offering, the activation modules for all the offerings can share the same SMP/E environment.

Make sure that you have access to the Program Directory for the activation module for the edition that you are installing. See [“Program Directories”](#) on page 2 for details.

About this task

This task describes the steps that are required to install the activation module.

Procedure

1. Install the activation module by using the instructions in the program directory for the activation module component that you require.
2. Authorize the data set.

Results

You installed data sets containing the activation module called `CICSTS56.HLQ.suffix`. Where *suffix* is the activate module suffix, which is one of the following:

- SDFHLIC for CICS TS for z/OS V5.6 - activation module
- SDFHDEV for CICS TS for z/OS Developer Trial V5.6 - activation module
- SDFHVUE for CICS TS for z/OS Value Unit Edition V5.6 - activation module

What to do next

If you previously installed CICS TS for z/OS Developer Trial Version 5.6 and are upgrading to another edition of the product, follow the instructions in [Upgrading from CICS TS Developer Trial](#).

If this is the first version of CICS TS for z/OS, Version 5.6 that you installed, you must install the base component, as described in [Chapter 5, “Installing CICS TS,”](#) on page 17.

Installing the CICS TS base component from CBPDO

You install all the elements from the CICS TS CBPDO using a single installation process.

The CBPDO *Memo to Users Extension* contains information about the CBPDO you ordered and the features and service it includes. It also contains CBPDO installation information.

Read the *Memo to Users Extension* thoroughly before starting any of the installation tasks.

The Program Directory gives a sample IEBCOPY job that you can customize to copy RELFILE(2) from the CICS TS CBPDO tape.

1. Modify the **LABEL=3** parameter to reference the file number of RELFILE(2) as supplied on the CBPDO tape.
2. When you have copied RELFILE(2) to DASD, you generate a single set of installation jobs using the CICS TS job generator, DFHISTAR.
3. This generates the following set of customized installation jobs, based on the parameters you specify to DFHISTAR:
 - DFHIHFS0, DFHIHFS1, DFHIHFSA, and DFHISMKD, the z/OS UNIX-related jobs
 - DFHINST1 through DFHINST6

Using the SMP/E RECEIVE, APPLY, and ACCEPT commands

The process for using the SMP/E RECEIVE, APPLY, and ACCEPT commands is explained in the Program Directory for CICS Transaction Server.

Using DFHISTAR process

DFHISTAR provides default qualifiers for the data sets into which you install CICS TS. The DFHISTAR job provides several parameters to enable you customize your settings.

CICS TS comprises a number of elements that are installed in a single process by the DFHINST n jobs. To ensure the element libraries are easily identified, DFHISTAR adds an element qualifier to the data set names. Using the default high-level qualifier CICSTS56, the names generated by DFHISTAR take the following form:

CICS

CICSTS56.CICS.ddname

CICSplex SM

CICSTS56.CPSM.ddname

REXX for CICS

CICSTS56.REXX.ddname

To enable you to customize the z/OS UNIX-related jobs, the DFHISTAR job provides the following parameters:

HFS0DSN

The data set name of the file system to be mounted at directory /pathprefix/usr/lpp/cicsts. Pathprefix is variable, and optional. The other parts of this directory name structure are fixed. The default data set name is OMVS.USR.LPP.CICSTS.

The DFHIHFS0 job uses this parameter.

HFS1DSN

The data set name of the file system to be mounted at directory /pathprefix/usr/lpp/cicsts/ussdir, where ussdir is a variable you specify in the ussdir parameter in DFHISTAR. If you omit the **ussdir** parameter it defaults to the value of the **TINDEX** parameter in lowercase, which in turn defaults to CICSTS56, so if both defaults are taken, the full directory path resolves to /pathprefix/usr/lpp/cicsts/cicsts56.

The default data set name is OMVS.USR.LPP.CICSTS.CICSTS56.

The DFHIHFS1 job uses this parameter.

HFSADSN

The data set name of the file system to be mounted at directory /pathprefix/usr/lpp/cicsts/ussdira, where *ussdira* is a variable you also specify in DFHISTAR. If you omit the **ussdira** parameter, it defaults to the value (in lowercase) of *aindex*, which, in turn, defaults to CICSTS56.A. The default data set name is OMVS.USR.LPP.CICSTS.CICSTS56.A.

The DFHIHFSA job uses this parameter.

“Editing the DFHISTAR job” on page 22 describes all the parameters that you can specify to customize the installation jobs, and explains the jobs that are generated.

Installing the CICS TS base component from ServerPac

The ServerPac distribution of CICS TS can be installed through z/OSMF. You receive the package, customize the installation information (for example, you can define the names of the SMP/E zones into which you want to install CICS TS, define values for the installation variables used in skeleton batch jobs, and define catalog data set names and the aliases associated with them, among many other options), then deploy the installation.

Installing with z/OSMF

z/OSMF is a driving system requirement (it must be up and running on at least one of your systems and you have the necessary authority to use it). Find information about preparing z/OSMF, downloading the ServerPac, configuring and deploying the installation in [ServerPac Installation using z/OSMF](#).

Installing CICS TS using DFHISTAR

Use the DFHISTAR job for installing CICS TS.

Table 3. CICS Transaction Server installation checklist			
Step	Done?	Description	See topic
1		Check that you have received the material that you ordered.	N/A
2		Check that you have all the installation prerequisites.	See the Program Directory
3		Copy RELFILE(2) from the distribution tape. Record your name for the TDFHINST library: _____ ----	See the Program Directory
4		Edit the DFHISTAR job in the TDFHINST library.	“Editing the DFHISTAR job” on page 22
4.1		Specify the CICS Transaction Server temporary installation libraries.	“Specifying the CICS Transaction Server temporary installation libraries” on page 26
4.2		Specify the JOB parameters for installation jobs.	“Specifying the JOB parameters for installation jobs” on page 27

Table 3. CICS Transaction Server installation checklist (continued)

Step	Done?	Description	See topic
4.3		Specify the scope of the installation.	“Specifying the scope of the installation” on page 27
4.4		Specify the type of JES to be used.	“Specifying the type of JES to be used” on page 27
4.5		Specify the utilities to be used.	“Specifying the utilities” on page 27
4.6		Specify the prefix of CICS Transaction Server jobs.	“Specifying the prefix of CICS Transaction Server jobs” on page 28
4.8		Specify the prefix of the CSSLIB library.	“Specifying the data set name of the CSSLIB library” on page 40
4.9		Specify the prefix of the SISLOAD library.	“Specifying the library data set names” on page 40
4.10		Specify the indexes of CICS Transaction Server data sets.	“Specifying the indexes of CICS Transaction Server data sets” on page 28
4.11		Specify block sizes.	“Specifying block sizes” on page 30
4.13		Specify the disk unit for work data sets.	“Specifying the disk unit for work data sets” on page 31
4.14		Specify disk volumes.	“Specifying disk volumes” on page 31
4.15		Allocate the space for CICS Transaction Server disk volumes.	“Allocating space for CICS Transaction Server disk volumes” on page 33
4.16		Specify SMP/E zone attributes.	“Specifying SMP/E zone attributes” on page 35
4.17		Specify attributes of the temporary SMP/E work data sets.	“Specifying attributes of the temporary SMP/E work data sets” on page 34
4.18		Specify attributes of the permanent SMP/E data sets.	“Specifying attributes of the permanent SMP/E data sets” on page 35
4.19		Specify the high-level qualifiers for SMP/E data sets.	“Specifying the high-level qualifiers for SMP/E data sets” on page 38

<i>Table 3. CICS Transaction Server installation checklist (continued)</i>			
Step	Done?	Description	See topic
4.20		Specify the z/OS UNIX directories and data sets.	“Specifying the CICS TS z/OS UNIX directories and data sets” on page 29
4.21		Specify the distribution tape device type.	“Specifying the distribution tape device type” on page 38
4.22		Specify attributes of the CICS Transaction Server system data sets.	“Specifying attributes of the CICS Transaction Server system data sets” on page 38
4.23		Specify attributes specific to CICSplex SM (optional).	“Specifying attributes specific to CICSplex SM” on page 43
4.24		Specify attributes of any additional target libraries.	“Specifying attributes of any additional target libraries” on page 38
5		Create RACF® profiles for the CICS Transaction Server data sets.	“Creating RACF profiles for the CICS Transaction Server data sets” on page 45
6		Run the DFHISTAR job.	“Running the DFHISTAR job” on page 46
6.1		Check the output from the DFHISTAR job.	“Checking the output from the DFHISTAR job” on page 46
8		Run the installation jobs.	“Running the installation jobs” on page 46
8.1		Check the output from the installation jobs.	“Checking the output from the installation jobs” on page 22
9		Copy the CICS Transaction Server procedures into a procedure library.	“Copying the CICS Transaction Server procedures into a procedure library” on page 52
10		Create any extra sets of CICS Transaction Server target libraries (optional).	“Creating extra sets of CICS Transaction Server target libraries (optional)” on page 55
11		Load the Japanese national language feature (optional).	“Loading other optional features” on page 57

This topic covers the following subjects:

- [“Editing the DFHISTAR job” on page 22](#)
- [“Creating RACF profiles for the CICS Transaction Server data sets” on page 45](#)

- [“Running the DFHISTAR job” on page 46](#)
- [“Running the installation jobs” on page 46](#)
- [“Checking the output from the installation jobs” on page 22](#)
- [“Postinstallation activities” on page 52](#)

Checking the output from the installation jobs

When you have successfully run all of the installation jobs, CICS Transaction Server is loaded.

You now have CICS Transaction Server installed on your DASD. Back up the volume on which CICS Transaction Server resides. If any errors occur during customization later, you do not have to re-run the installation jobs.

Editing the DFHISTAR job

You can edit the DFHISTAR job to assign values to installation parameters for your environment.

The DFHISTAR job is in the TDFHINST library when you copy RELFILE(2) from the distribution tape. You can either edit the DFHISTAR job directly or copy the DFHISTAR job to preserve the IBM supplied values and edit your copy.

Remember that the product is installed using cataloged data sets.

To help you look up details about a particular parameter, Table 4 on page 22 lists the DFHISTAR parameters, in alphabetical order, with their predefined values. The parameters in the DFHISTAR job itself are listed in associated groups. The table also indicates whether a parameter is relevant to CICS, or CICSplex SM, or both.

The default values of the parameters are the same as the IBM supplied values. You can use the IBM supplied values, define your own values, or let the CICS Transaction Server installation process determine default values for you.

Enter your values for parameters of the DFHISTAR job in lowercase. Except for the z/OS UNIX parameters **USSDIR**, **USSDIRA**, and **JAVADIR**, the values are translated into uppercase when you run the DFHISTAR job.

Table 4. Alphabetical list of parameters for the DFHISTAR job				
Parameter	CICS	CICSplex SM	Supplied value	Refer to
ACTIVATE	X	X	SDFHLIC	“Specifying the data set name of the activation module” on page 29
ADDTVOL	X		CICS56 SYSALLDA	“Specifying attributes of any additional target libraries” on page 38
AINDEX	X		CICSTS56.A	“Specifying attributes of any additional target libraries” on page 38
ALINDEX	X		SYS1.CICSTS56.A	“Specifying attributes of any additional target libraries” on page 38
ASMPLTS	X		CICSTS56.A.SMPLTS	“Specifying attributes of any additional target libraries” on page 38
ASMPMTS	X		CICSTS56.A.SMPMTS	“Specifying attributes of any additional target libraries” on page 38
ASMPSCDS	X		CICSTS56.A.SMPSCDS	“Specifying attributes of any additional target libraries” on page 38
ASMPSTS	X		CICSTS56.A.SMPSTS	“Specifying attributes of any additional target libraries” on page 38
AZONE	X		AZONE	“Specifying attributes of any additional target libraries” on page 38

Table 4. Alphabetical list of parameters for the DFHISTAR job (continued)				
Parameter	CICS	CICSplex SM	Supplied value	Refer to
AZONECSI	X		CICSTS56.A.AZONE	“Specifying attributes of any additional target libraries” on page 38
AZONELOG	X		CICSTS56.A.AZONE.SMPLOG	“Specifying attributes of any additional target libraries” on page 38
BLKFB80	X		0	“Specifying block sizes” on page 30
BLKISPF			3200	This parameter is now obsolete.
BLKU	X	X	32760	“Specifying block sizes” on page 30
CMACVOL	X		CICS56	“Specifying disk volumes” on page 31
CMASNAME		X	CMAS01	“Specifying attributes specific to CICSplex SM” on page 43
CMCIPT		X	12346	“Specifying attributes specific to CICSplex SM” on page 43
CMSSYSID		X	CM01	“Specifying attributes specific to CICSplex SM” on page 43
CSSLIB	X		SYS1.CSSLIB	“Specifying the data set name of the CSSLIB library” on page 40
CSYSYSID		X	CS01	“Specifying attributes specific to CICSplex SM” on page 43
CSYSNAME		X	CSYS01	“Specifying attributes specific to CICSplex SM” on page 43
CSYSPLEX		X	CSYPLX01	“Specifying attributes specific to CICSplex SM” on page 43
DEFVOL	X	X	CICS56 SYSALLDA	“Specifying disk volumes” on page 31
DINDEX	X	X	CICSTS56	“Specifying the indexes of CICS Transaction Server data sets” on page 28
DISTVOL	X	X	CICS56 SYSALLDA	“Specifying disk volumes” on page 31
DOWNLOAD			CICSTS56.DOWNLOAD	“Specifying the DFHISTAR DOWNLOAD parameter” on page 45
DSINFO	X	X	CICSTS56 SYSALLDA	“Specifying attributes of the CICS Transaction Server system data sets” on page 38
DZONE	X		DZONE	“Specifying SMP/E zone attributes” on page 35
DZONECSI	X		CICSTS56.DZONE NEW CICSTS56 SYSALLDA	“Specifying SMP/E zone attributes” on page 35
DZONELOG	X		CICSTS56.DZONE.SMPLOG NEW	“Specifying SMP/E zone attributes” on page 35
GINDEX	X	X	CICSTS56	“Specifying the indexes of CICS Transaction Server data sets” on page 28
GZONE	X		NEW CICSOPT	“Specifying SMP/E zone attributes” on page 35
GZONECSI	X	X	CICSTS56.GZONE NEW CICSTS56 SYSALLDA	“Specifying SMP/E zone attributes” on page 35
GZONELOG	X		CICSTS56.GZONE.SMPLOG NEW	“Specifying SMP/E zone attributes” on page 35
HFS0DSN	X		OMVS.USR.LPP.CICSTS	“Specifying the CICS TS z/OS UNIX directories and data sets” on page 29

Table 4. Alphabetical list of parameters for the DFHISTAR job (continued)				
Parameter	CICS	CICSplex SM	Supplied value	Refer to
HFS1DSN	X		OMVS.USR.LPP.CICSTS.CICSTS56	“Specifying the CICS TS z/OS UNIX directories and data sets” on page 29
HFSADSN	X		OMVS.USR.LPP.CICSTS.CICSTS56.A	“Specifying the CICS TS z/OS UNIX directories and data sets” on page 29
JAVADIR	X		java/J8.0_64	“Specifying the installation directory for Java support” on page 30
JES	X		JES2	“Specifying the type of JES to be used” on page 27
JOB	X	X	(No valid value)	“Specifying the JOB parameters for installation jobs” on page 27
LIB	X	X	CICSTS56.XDFHINST	“Specifying the CICS Transaction Server temporary installation libraries” on page 26
LINDEX	X	X	SYS1.CICSTS56	“Specifying the indexes of CICS Transaction Server data sets” on page 28
LOGGER-INFO	X		001 500 4096 64000 2048 MVSX XXXXXXXX XXXXXXXX	“Specifying log stream and log stream structure attributes” on page 43
MODNAME	X		DFHCSVC	“Installing the CICS SVCs” on page 84
NEWDREP-PREFIX		X	.	“Specifying attributes specific to CICSplex SM” on page 43
OLDDREP		X	.	“Specifying attributes specific to CICSplex SM” on page 43
OPTVOL	X		CICS56 SYSALLDA	“Specifying disk volumes” on page 31
PATHPREFIX	X		Null string	“Specifying the CICS TS z/OS UNIX directories and data sets” on page 29
PREFIX	X		DFH	“Specifying the prefix of CICS Transaction Server jobs” on page 28
SCEECICS	X	X	SYS1.SCEECICS	“Specifying the data set names of the SCEECICS and SCEERUN libraries” on page 41
SCEECPP	X		SYS1.SCEECPP	“Specifying the data set name of the SCEECPP library” on page 41
SCEELIB	X		SYS1.SCEELIB	“Specifying the data set names of the SCEERUN2 library” on page 41
SCEEBND2	X		SYS1.SCEEBND2	“Specifying the data set name of the SCEEBND2 library” on page 40
SCEERUN	X	X	SYS1.SCEERUN	“Specifying the data set names of the SCEECICS and SCEERUN libraries” on page 41
SCEERUN2	X	X	SYS1.SCEERUN2	“Specifying the data set names of the SCEERUN2 library” on page 41
SCEELKED	X		SYS1.SCEELKED	“Specifying the data set name of the SCEELKED library” on page 40
SCEELKEX	X		SYS1.SCEELKEX	“Specifying the data set name of the SCEELKEX library” on page 40
SCEE OBJ	X		SYS1.SCEE OBJ	“Specifying the data set name of the SCEE OBJ library” on page 41
SCEESAMP	X	X	SYS1.SCEESAMP	“Specifying the data set name of the SCEESAMP library” on page 42

Table 4. Alphabetical list of parameters for the DFHISTAR job (continued)

Parameter	CICS	CICSplex x SM	Supplied value	Refer to
SCSFMOD0	X		SYS1.SCSFMOD0	“Specifying the data set names of the SCSFMOD0 and SIXMEXP libraries” on page 43
SCLBSID	X		SYS1.SCLBSID	“Specifying the data set name of the SCLBSID library” on page 41
SCOPE	X	X	ALL	“Specifying the scope of the installation” on page 27
SCSQLOAD	X		SYS1.SCSQLOAD	“Specifying the data set name of the SCSQLOAD library” on page 42
SCSQANLE	X		SYS1.SCSQANLE	“Specifying the data set name of the SCSQANLE library” on page 42
SCSQCICS	X		SYS1.SCSQCICS	“Specifying the data set name of the SCSQCICS library” on page 42
SCSQAUTH	X		SYS1.SCSQAUTH	“Specifying the data set name of the SCSQAUTH library” on page 42
SDSNLOAD	X		SYS1.SDSNLOAD	“Specifying the data set name of the SDSNLOAD library” on page 42
SEZACMTX	X		SYS1.SEZACMTX	“Specifying the data set names of the SEZARPCL and SEZACMTX libraries” on page 41
SEZARPCL	X		SYS1.SEZARPCL	“Specifying the data set names of the SEZARPCL and SEZACMTX libraries” on page 41
SISPLOAD	X		SYS1.SISPLOAD	“Specifying the library data set names” on page 40
SIXMEXP	X		SYS1.SIXMEXP	“Specifying the data set names of the SCSFMOD0 and SIXMEXP libraries” on page 43
SMPLTS	X		CICSTS56.SMPLTS	“Specifying attributes of the permanent SMP/E data sets” on page 35
SMPMTS	X		CICSTS56.SMPMTS	“Specifying attributes of the permanent SMP/E data sets” on page 35
SMPPTS	X		CICSTS56.SMPPTS	“Specifying attributes of the permanent SMP/E data sets” on page 35
SMPSCDS	X		CICSTS56.SMPSCDS	“Specifying attributes of the permanent SMP/E data sets” on page 35
SMPSTS	X		CICSTS56.SMPSTS	“Specifying attributes of the permanent SMP/E data sets” on page 35
SMPVOL	X	X	CICS56 SYSALLDA	“Specifying disk volumes” on page 31
SMPWORK	X	X	SYSALLDA	“Specifying attributes of the temporary SMP/E work data sets” on page 34
SMS	X		NO	“Specifying the SMS option for DASD allocations” on page 31
SVCNUMB	X		216	“Installing the CICS SVCs” on page 84
TAPEUNIT	X		3480	“Specifying the distribution tape device type” on page 38
TARGVOL	X	X	CICS56 SYSALLDA	“Specifying disk volumes” on page 31
TCPIPHST		X	XXXXXXXX.XXXXXXXXXX.XXXXXX XX.XXXXXXXXXX	“Specifying attributes specific to CICSplex SM” on page 43

Table 4. Alphabetical list of parameters for the DFHISTAR job (continued)				
Parameter	CICS	CICSplex SM	Supplied value	Refer to
TCPIPPRT		X	12345	“Specifying attributes specific to CICSplex SM” on page 43
TEMPLIB	X	X	CICSTS56.TDFHINST	“Specifying the CICS Transaction Server temporary installation libraries” on page 26
TIMEZONE		X	B	“Specifying attributes specific to CICSplex SM” on page 43
TINDEX	X	X	CICSTS56	“Specifying the indexes of CICS Transaction Server data sets” on page 28
TZONE	X	X	TZONE	“Specifying SMP/E zone attributes” on page 35
TZONECSI	X		CICSTS56.TZONE NEW CICSTS56 SYSALLDA	“Specifying SMP/E zone attributes” on page 35
TZONELOG	X		CICSTS56.TZONE.SMPLOG NEW	“Specifying SMP/E zone attributes” on page 35
USSDIR	X		.	“Specifying the CICS TS z/OS UNIX directories and data sets” on page 29
USSDIRA	X		.	“Specifying attributes of any additional target libraries” on page 38
UTILITIES	X	X	ASMA90 IEWL GIMSMP IEBCOPY	“Specifying the utilities” on page 27
WORKUNIT	X	X	SYSALLDA	“Specifying the disk unit for work data sets” on page 31
WUI		X	YES	“Specifying attributes specific to CICSplex SM” on page 43
WUINAME		X	WUINCM01	“Specifying attributes specific to CICSplex SM” on page 43
WUIPLEX		X	WUIPCM01	“Specifying attributes specific to CICSplex SM” on page 43
WUISYSID		X	WU01	“Specifying attributes specific to CICSplex SM” on page 43
XTRAQUAL	X		...	“Specifying extra qualifiers” on page 29

Specifying the CICS Transaction Server temporary installation libraries

Specify your data set names for the two temporary libraries that are used to install CICS Transaction Server. If you do not want to use the default names, record your values for the TEMPLIB and LIB parameters.

TEMPLIB library_name

Specifies the name of the temporary installation library that contains the skeleton installation jobs. Specify the name of the data set into which you copied RELFILE(2) from the distribution tape, described in *Copy RELFILE(2) from the Distribution Tape* in the Program Directory.

Also specify this name on the SYSPROC DD statement of the DFHISTAR job.

LIB library_name

Specifies the name of the installation output library to which the jobs generated by the DFHISTAR job are added.

Specifying the JOB parameters for installation jobs

Decide which parameters you want to use on the JOB statements of the CICS Transaction Server installation jobs and specify them on the JOB parameter.

JOB accounting_information

Specifies the JOB statement and accounting information that you want substituted into the jobs generated by the DFHISTAR job. Here is an example:

```
JOB //XXXXXXX JOB 1,userid,MSGCLASS=A,MSGLEVEL=(1,1),
JOB //          CLASS=A,NOTIFY=userid
JOB /*JOBPARM SYSAFF=node1
JOB /*ROUTE PRINT node2.userid
```

1. Do not change XXXXXXXX given in the sample JOB statement in the DFHISTAR job because it is the 8-character job name that is substituted by the DFHISTAR job. For example, for the installation job DFHIVPBT, the DFHISTAR job changes XXXXXXXX to DFHIVPBT.
2. Normal JCL rules for coding JOB statements apply to the JOB parameter.
3. Delete or comment out extra lines of the JOB statement that you do not require.
4. Normal JCL rules apply when coding the JOB statement; for example, all lines except the last line must end in a comma.

Specifying the scope of the installation

Specify the scope of the CICS Transaction Server installation on the **SCOPE** parameter.

SCOPE ALL|BASE|POST

Specifies whether you want to generate all the CICS Transaction Server installation and postinstallation jobs, or only the postinstallation jobs. When you install CICS Transaction Server from the distribution tape, specify the default, SCOPE ALL. Code the other options, if necessary, during postinstallation tasks.

ALL

Specifies that you want to generate all the CICS Transaction Server installation jobs and all the postinstallation jobs.

BASE

Specifies that you want to generate only the installation jobs (DFHINST1 through DFHINST6, DFHIHFS0, DFHIHFS1, and DFHISMKD) that you use to install CICS Transaction Server from the distribution tape.

POST

Specifies that you want to generate only the postinstallation jobs that you can use to create the CICS Transaction Server data sets, and run the IVPs.

Specifying the type of JES to be used

Specify the type of job entry subsystem (JES) that you use to install CICS Transaction Server on the JES parameter. The DFHISTAR job generates jobs with statements suitable for JES2 or JES3.

JES JES2|2|JES3|3

Specifies the release of JES that you are using. If you are using JES2, specify *JES2* or 2. If you are using JES3, specify *JES3* or 3.

Specifying the utilities

Specify the utilities to install CICS Transaction Server on the UTILITIES parameter.

UTILITIES *asmprog binder smpeprog copyutil*

Specifies the names of utility programs to be used when installing CICS Transaction Server elements and programs that it uses.

asmprog

The program name of the assembler. Specify *ASMA90* for High Level Assembler/MVS and VM and VSE, which is required.

binder

The program name of the z/OS binder. Ensure that program IEWL references the z/OS program management binder.

smpeprog

The program name of the SMP/E program. The IBM-supplied name is GIMSMP.

copyutil

The program name of the data set copy utility program. The IBM-supplied name is IEBCOPY.

The High Level Assembler either must be in the LINKLST concatenation or you must add a STEPLIB DD statement that points to the library containing the High Level Assembler in any jobs that invoke SMP/E.

Specifying the prefix of CICS Transaction Server jobs

Specify the 1- to 6-character prefix to be added to the jobs generated by the DFHISTAR job. This prefix overwrites the first characters of the job name.

For example, PREFIX USERID changes the job name DFHINST1 to USERIDT1.

PREFIX prefix

The 1- to 6-character prefix to be added to the CICS jobs generated by the DFHISTAR job.

Specifying the indexes of CICS Transaction Server data sets

Optionally specify the high-level indexes (high-level qualifiers) for the CICS Transaction Server distribution, target, and SMP/E libraries allocated by the installation process. If you do not specify these parameters, they will take default values.

GINDEX library_prefix

Assigns a high-level index to the CICS Transaction Server SMP/E global libraries allocated by the installation process.

The *library_prefix* value must not be longer than 26 characters and the leading character must be alphabetic. If you specify more than one level of index, the names must be separated by a period; for example, GINDEX CICSTS56.TEST.

TINDEX library_prefix

Assigns a high-level index to the CICS Transaction Server SMP/E target libraries (except for the SDFHLINK, SDFHLPA, SEYULINK, and SEYULPA target libraries) allocated by the installation process.

1. The LINDEX parameter defines the high-level index for the SDFHLINK and SDFHLPA libraries.
2. The *dsindex* operand of the DSINFO parameter defines the high-level index for the data sets created by the DFHCOMDS and DFHDEFDS jobs.

The *library_prefix* value must not be longer than 26 characters and the leading character must be alphabetic. If you specify more than one level of index, the names must be separated by a period; for example, TINDEX CICSTS56.TEST.

DINDEX library_prefix

Assigns a high-level index to the CICS Transaction Server SMP/E distribution libraries allocated by the installation process.

The *library_prefix* value must not be longer than 26 characters and the leading character must be alphabetic. If you specify more than one level of index, the names must be separated by a period; for example, DINDEX CICSTS56.TEST.

LINDEX library_prefix

Assigns a high-level index to the SDFHLPA, SDFHLINK, SEYULINK, and SEYULPA libraries allocated by the installation process. Define the *library_prefix* value in the MVS Main Catalog.

The *library_prefix* value must not be longer than 26 characters and the leading character must be alphabetic. If you specify more than one level of index, the names must be separated by a period; for example, LINDE_X SYS1.CICSTS53.TEST.

Specifying the data set name of the activation module

Specify the data set name of the activation module that is used during the installation process.

ACTIVATE *dsname*

Specifies the data set qualifier for the activation module that is required to run your version of CICS. DFHSTART is updated with the activation module data set.

The value that you specify must correspond with the version of CICS that you are installing.

SDFHLIC

Specifies the licensed edition (the default value).

SDFHDEV

Specifies the Developer Trial.

SDFHVUE

Specifies the Value Unit Edition.

Specifying extra qualifiers

Specify extra qualifiers that can optionally be inserted into the data set name of the 'target', 'distribution', and 'additional' zone data sets respectively, and inserted before the last data set qualifier. For example, XTRAQUAL JDOE . . changes the name of the target zone libraries to the values set by TINDEX.CICSTS56.CICS.JDOE.SDFHLOAD.

XTRAQUAL . . .

Three qualifiers to be used by 'target', 'distribution', and 'additional' zone data sets. If a qualifier is not required, specify a period (.).

Specifying the CICS TS z/OS UNIX directories and data sets

The DFHISTAR job has parameters that enable you to customize the UNIX system services z/OS UNIX directories.

PATHPREFIX

The name of an optional prefix to the CICS TS directory /usr/lpp/cicsts. For example:

```
/example/usr/lpp/cicsts
```

USSDIR

The name of the CICS TS directory, under /*pathprefix*/usr/lpp/cicsts.

The full path is therefore /*pathprefix*/usr/lpp/cicsts/*ussdir*.

USSDIR is a name that you can specify. The default value is the value of the TINDEX parameter in lowercase.

The default path is:

```
/pathprefix/usr/lpp/cicsts/cicsts56
```

The name of the UNIX System Services directory after the root directory (/usr/lpp) is always /*cicsts*.

USSDIRA *dsname*

The name of the UNIX System Services directory for the Additional Target zone, under /*pathprefix*/usr/lpp/cicsts.

The full path is therefore /*pathprefix*/usr/lpp/cicsts/*ussdira*.

USSDIRA is a name that you can specify. The default value is the value of the AINDEX parameter in lowercase.

The default path is:

```
/pathprefix/usr/lpp/cicsts/cicsts56
```

The name of the UNIX System Services directory after the root directory (/usr/lpp) is always /cicsts.

HFS0DSN

The data set name of the file system to be mounted at directory */pathprefix/usr/lpp/cicsts*.

These directory names are fixed. The default is data set name OMVS.USR.LPP.CICSTS .

See [“Running the DFHIHFS0 job” on page 47](#) for details of the job that uses this parameter.

HFS1DSN

The data set name of the file system to be mounted at directory */pathprefix/usr/lpp/cicsts/ussdir*, where *ussdir* is the name of the directory specified on the ussdir parameter in the DFHISTAR job. The default is data set name OMVS.USR.LPP.CICSTS.CICSTS56 .

See [“Running the DFHIHFS1 job” on page 48](#) for details of the job that uses this parameter.

HFSADSN

The data set name of the z/OS UNIX equivalent of the SMP/E "additional target zone", to be mounted at directory */pathprefix/usr/lpp/cicsts/ussdira*, where *ussdira* is the name of the directory specified on the ussdira parameter in the DFHISTAR job. The default data set name is OMVS.USR.LPP.CICSTS.CICSTS56.A .

See [“Creating extra sets of CICS Transaction Server target libraries \(optional\)” on page 55](#) for details of the job that uses this parameter.

Specifying the installation directory for Java support

Specify the directory where the IBM 64-bit SDK for z/OS, Java Technology Edition is installed on z/OS UNIX. Support for the 31-bit version of the SDK is withdrawn, so you must upgrade to use the 64-bit version.

The **JAVADIR** parameter specifies the location of the SDK on z/OS UNIX. The supplied value *java/J8.0_64* points to the default installation directory for Version 8 of the IBM 64-bit SDK for z/OS, Java Technology Edition. This value of this parameter is appended to */pathprefix/usr/lpp/*, giving a full path name of */pathprefix/usr/lpp/javadir*.

The installation directory is used in the CICS-supplied sample JVM profiles.

Specifying block sizes

Specify the block sizes to be used when allocating data sets during installation on the BLKFB80 and BLKU parameters.

BLKFB80 {0|blocksize}

The block size to be used when allocating data sets that have a fixed block record format and record length of 80 bytes.

The IBM-supplied value in DFHISTAR is 0. Leave this value specified as 0 to allow z/OS to determine the optimum block size for you.

BLKU {32760|blocksize}

The block size to be used when allocating data sets that have an undefined record length.

Specifying the disk unit for work data sets

Specify the UNIT parameter for the disk or disks on which work data sets are stored on the WORKUNIT parameter.

WORKUNIT disktype

A unit identifier.

Specifying the SMS option for DASD allocations

Specify the extent to which you want to leave the allocation of CICS TS installation data sets to SMS. You can either let SMS handle all DASD allocation, or you can use volume parameters to control the allocations that do not have to be managed by SMS.

Some CICS TS data sets are installed in PDSE data sets. These are the following data sets:

- The SMP/E SMPLTS data set.
- The distribution libraries ADFHMOD and ADFHMOD2.
- The target libraries, SDFHLOAD, SDFHAUTH, and SDFHWSLD.

SMS YES | NO

If you specify SMS YES, the VOLUME parameter is omitted from the generated installation jobs and all data set allocations are handled by SMS.

If you specify SMS NO, the VOLUME parameter is included on the generated installation jobs, and is recognized according to your SMS configuration.

The VOLUME parameters used are those specified on the ADDTVOL, DEFVOL, DISTVOL, CMACVOL, OPTVOL, SMPVOL, and TARGVOL parameters.

Specifying disk volumes

To make the best use of your disk space, you can specify your own disk volumes and device types to be used to install CICS Transaction Server.

If you intend to install CICS Transaction Server into disk space managed by the storage management subsystem (SMS) component of MVS/DFP, you do not have to specify your own disk volumes; SMS can determine device assignment. In this case, proceed to “Specifying SMP/E zone attributes” on page 35. For further information about installing system-managed storage and about planning for and migrating storage to an SMS-managed environment, see [z/OS DFSMSdfp Storage Administration](#)

You can specify your own disk details using the following parameters:

DEFVOL volume disktype

Defines the default disk on which the contents of the disk volumes CMACVOL, DISTVOL, OPTVOL, SMPVOL, and TARGVOL resides if the appropriate parameter is not coded in the DFHISTAR job. For example, if you do not code the DISTVOL parameter, the CICS Transaction Server distribution libraries resides on the disk defined by DEFVOL.

volume

Is one of the following entries:

- The volume serial identifier, in the range 1 through 6 characters, of the default volume.
- A period (.) if all volumes other than CMACVOL and SMPVOL that are not specifically defined by the appropriate parameter of the DFHISTAR job are put onto any available volume. The CMACVOL and SMPVOL volumes are put onto the same volume as the library specified by the TEMPLIB parameter.

disktype

Is the UNIT parameter of the volume.

If you omit the DEFVOL parameter, all volumes that are not specifically defined by the appropriate parameter of the DFHISTAR job are put onto the same volume as the library specified by the TEMPLIB parameter.

DISTVOL volume disktype

Defines the disk on which the CICS Transaction Server distribution libraries resides.

volume

Is one of the following entries:

- The volume serial identifier, in the range 1 through 6 characters, of the volume on which the distribution libraries resides.
- A period (.) if the CICS Transaction Server libraries are to be put onto any available volume.

disktype

Is the UNIT parameter of the volume.

If you omit the DISTVOL parameter, the distribution libraries are put on the volume specified by the DEFVOL parameter. If the DEFVOL parameter is omitted, or if a period (.) is specified for its *volume* operand, the distribution libraries are put onto any available volume.

TARGVOL volume disktype

Specifies details of the disk containing the CICS Transaction Server target libraries.

volume

Is one of the following entries:

- The volume serial identifier, in the range 1 through 6 characters, of the volume on which the CICS Transaction Server target libraries are to reside.
- A period (.) if the CICS Transaction Server target libraries are to be put onto any available volume.

disktype

Is the UNIT parameter for the volume.

If you omit the TARGVOL parameter, the CICS Transaction Server target libraries are put onto the volume specified by the DEFVOL parameter. If the DEFVOL parameter is omitted, or if a period (.) is specified for its *volume* operand, the CICS Transaction Server target libraries are put onto any available volume.

SMPVOL volume disktype

Specifies the disk that contains the permanent, non-VSAM SMP/E data sets for CICS Transaction Server that are associated with global or distribution zones and are therefore unique.

volume

Is one of the following entries:

- The volume serial identifier, in the range 1 through 6 characters, of the volume on which the permanent non-VSAM SMP/E data sets are to reside.
- A period (.) if the permanent non-VSAM SMP/E data sets are to be put onto the same volume as the library specified by the TEMPLIB parameter.

disktype

Is the UNIT parameter for the volume.

If you omit the SMPVOL parameter, the permanent non-VSAM SMP/E data sets for CICS Transaction Server are put on the volume specified by the DEFVOL parameter. If the DEFVOL parameter is omitted, or if a period (.) is specified for its *volume* operand, the data sets are put onto the same volume as the library specified by the TEMPLIB parameter.

OPTVOL volume disktype

Specifies details of the disk onto which the optional source material is copied.

volume

Is one of the following entries:

- The volume serial identifier, in the range 1 through 6 characters, of the volume on which the optional source material is to reside.
- A period (.) if the optional source material is to be put on any available volume.

disktype

Is the UNIT parameter of the volume. This is required only if *volume* is specified.

If you omit the OPTVOL parameter, the optional source material is put on the volume specified by the DEFVOL parameter. If the DEFVOL parameter is omitted, or if a period (.) is specified for its *volume* operand, the optional source material is put onto any available volume.

CMACVOL volume

Defines the disk on which the VSAM KSDS, DFHMACD, resides. This data set is used for the CICS Transaction Server messages facility (CICS-supplied transaction CMAC).

volume

Is one of the following entries:

- The volume serial identifier, in the range 1 through 6 characters, of the volume on which the VSAM KSDS, DFHMACD, is to reside.
- A period (.) if the DFHMACD data set is to be put onto the same volume as the library specified by the TEMPLIB parameter.

If you omit the CMACVOL parameter, the DFHMACD data set is put onto the volume specified by the DEFVOL parameter. If the DEFVOL parameter is omitted, or if a period (.) is specified for its *volume* operand, the DFHMACD data set is put onto the same volume as the library specified by the TEMPLIB parameter.

When are these volumes used?

The reference table lists the DFHISTAR volume parameters and details of their use.

DFHISTAR volume parameter	Installing	Applying service	Customizing	Assembling resource tables	Running CICS Transaction Server
SMPVOL	Yes	Yes	Yes	Yes	
DISTVOL	Yes	Yes	Yes		
TARGVOL	Yes	Yes	Yes	Yes	Yes
DZONECSI ¹	Yes	Yes	Yes		
TZONECSI ¹	Yes	Yes	Yes	Yes	
GZONECSI ¹	Yes	Yes	Yes	Yes	

¹ The entries for xZONECSI parameters are also for the associated xZONE parameters.

You require the RELFILE data sets on SMPVOL during installation only.

You require SMPVOL, DISTVOL, TARGVOL, DZONE, TZONE, and GZONE when you apply service or customize your CICS Transaction Server programs. You require SMPVOL and GZONE when you apply service or customize your alternative libraries for use with the extended recovery facility.

You require SMPVOL, TARGVOL, TZONE, and GZONE when you assemble your CICS Transaction Server tables. You require SMPVOL and GZONE when you assemble CICS Transaction Server tables for the second (alternate) CICS Transaction Server region.

You require only TARGVOL to run CICS Transaction Server.

Allocating space for CICS Transaction Server disk volumes

Whether or not you use SMS-managed data sets, you still require enough disk space in which to create the CICS Transaction Server disk volumes.

The space required by the installation jobs on these volumes depends on the type of disk you intend to use. The number of tracks required on the different types of DASD are given in [Table 5 on page 34](#). The Program Directory describes the size of the CICS Transaction Server distribution and target libraries.

<i>Table 5. Number of tracks required for CICS Transaction Server</i>		
Identification	3380	3390
CICSTS56.TDFHINST	15	15
CICSTS56.XDFHINST	15	15
Relfile data sets on SMPVOL	4875	4500
SMP/E non-VSAM data sets on SMPVOL	390	375
DISTVOL	4680	4320
TARGVOL	11235	10848
DZONE	165	165
TZONE	165	165
GZONE	165	165
Total during installation	21705	19905
Total after installation	16830	15405

Allow up to 15% on the values in [Table 5 on page 34](#) for servicing requirements. Secondary allocations are 10% of the primary allocations.

If you intend to store other IBM software or your own application programs in these libraries, you must modify the generated jobs accordingly.

Specifying attributes of the temporary SMP/E work data sets

You must define the attributes of the temporary SMP/E work data sets, SMPWRK1, SMPWRK2, SMPWRK3, SMPWRK4, and SMPWRK6, for the following CICS Transaction Server jobs: DFHINSTJ, DFHINST6, DFHLPUMD, DFHSMPE, and EYULPMOD.

Define the attributes of those SMP/E data sets on the SMPWORK parameter:

The CICS Transaction Server jobs used to install CICS Transaction Server for z/OS have DD statements for the SMP/E data sets that they must know about.

SMPWORK disktype

The UNIT parameter for the disk that is to contain the temporary SMP/E work data sets, SMPWRK1, SMPWRK2, SMPWRK3, SMPWRK4, and SMPWRK6, required to install CICS Transaction Server.

Do not allocate the SMPWRK6 data set to Virtual I/O (VIO). If you specify a value for *disktype*, ensure that SMPWRK6 cannot be allocated to VIO.

If you specify a value for *disktype*, or omit the SMPWORK parameter altogether, //SMPWRKn DD statements are added to the following jobs generated by the DFHISTAR job:

- DFHINSTJ
- DFHLPUMD
- DFHSMPE

If you specify NO, a period (.), or a null string, CICS Transaction Server assumes that SMP/E knows about the temporary SMP/E work data sets. To define the attributes of the SMP/E work data sets, you must do one of the following tasks:

- Provide appropriate DDDEFS for the temporary SMP/E work data sets.
- Apply the SMP/E sample usermod (SMP0001) that contains superzap statements for updating the default attributes of the SMP/E data sets in the GIMMPDFT module.

The GIMMPDFT module, which is part of SMP/E, defines the default attributes of SMP/E data sets, and can be used to dynamically allocate data sets to be used by all zones. For more information about the entries in the GIMMPDFT module and the sample entry values in the usermod SMP0001, see [SMP/E for z/OS Reference](#).

Specifying attributes of the permanent SMP/E data sets

You can specify the attributes of the permanent SMP/E data sets using the SMPPTS, SMPMTS, SMPSTS, SMPSCDS and SMPLTS parameters.

Specify the attributes of the permanent SMP/E data sets on the following parameters:

SMPPTS dsname

Specifies the name of the SMP/E primary data set used to store temporarily PTF function SYSMODs or other fixes that are in RECEIVE or APPLY status; that is, PTF fixes that have not been rejected or accepted.

SMPMTS dsname

Specifies the name of the SMP/E macro temporary store (MTS) data set used to store updated versions of macros. Although required by SMP/E, CICS does not use the MTS data set.

SMPSTS dsname

Specifies the name of the SMP/E source temporary store (STS) data set used to store updated versions of source elements. Although required by SMP/E, CICS does not use the STS data set.

SMPSCDS dsname

Specifies the name of the SMP/E saved control data set (SCDS) used to store old target zone entries that have been modified by inline JCLIN processing in a SYSMOD.

SMPLTS dsname

Specifies the name of the link-edit temporary (LTS) data set used with the CALLLIBS function. This data set must always be an SMS-managed PDSE, whether or not Java is installed, unless you have installed the PTFs for z/OS UNIX and PDSE support on non-managed SMS volumes listed in Flash10007, which can be found by links from <https://www.ibm.com/support/techdocs>.

The CICS Transaction Server jobs that must know the attributes of the SMP/E data sets have DD statements for them.

Specifying SMP/E zone attributes

Specify the attributes of the SMP/E distribution zone, global zone, target zone, and any additional target zones.

The CICS Db2® attachment facility contains modules named with the DSN prefix. Therefore, to prevent existing Db2 modules with the same DSNxxxxx names from being overwritten, do not install CICS Transaction Server into the same target and distribution zones as Db2.

To specify SMP/E zone attributes, use the following parameters:

GZONELOG dsname NEW|OLD

Specifies details of the SMP/E log for the global zone CSI.

dsname

The name of the global zone log.

NEW|OLD

Specifies whether an existing global zone log is to be used. If you specify NEW, any existing global zone log with the specified *dsname* is deleted, and a new global zone log is allocated. If you specify OLD, an existing global zone log is used.

TZONELOG dsname NEW|OLD

Specifies details of the SMP/E log for the target zone CSI.

dsname

The name of the target zone log.

NEW|OLD

Specifies whether an existing target zone log is to be used. If you specify NEW, any existing target zone log with the specified *dsname* is deleted, and a new target zone log is allocated. If you specify OLD, an existing target zone log is used.

DZONELOG dsname NEW|OLD

Specifies details of the SMP/E log for the distribution zone CSI.

dsname

The name of the distribution zone log.

NEW|OLD

Specifies whether an existing distribution zone log is to be used. If you specify NEW, any existing distribution zone log with the specified *dsname* is deleted, and a new distribution zone log is allocated. If you specify OLD, an existing distribution zone log is used.

GZONECSI cluster NEW|OLD volume disktype

Specifies details of the global zone CSI.

cluster

The VSAM cluster name, without the qualifier '.CSI'.

NEW|OLD

Specifies whether an existing global zone CSI is to be used. If you specify NEW, any existing global zone CSI with the specified *cluster* name is deleted, and a new global zone CSI is allocated. If you specify OLD, an existing global zone CSI is used.

volume

Either the volume serial (volser) identifier for the volume on which the global zone CSI is to be allocated or a period (.) if the CSI is to be put on a volume determined by the CICS Transaction Server installation process.

disktype

The UNIT parameter for the volume.

TZONECSI cluster NEW|OLD volume disktype

Specifies details of the target zone CSI.

cluster

The VSAM cluster name, without the qualifier '.CSI'.

NEW|OLD

Specifies whether an existing target zone CSI is to be used. If you specify NEW, any existing target zone CSI with the specified *cluster* name is deleted, and a new target zone CSI is allocated. If you specify OLD, an existing target zone CSI is used.

volume

Either the volume serial (volser) identifier for the volume on which the target zone CSI is to be allocated or a period (.) if the CSI is to be put on a volume determined by the CICS Transaction Server installation process.

disktype

The UNIT parameter for the volume.

DZONECSI cluster NEW|OLD volume disktype

Specifies details of the distribution zone CSI.

cluster

The VSAM cluster name, without the qualifier '.CSI'.

NEW|OLD

Specifies whether an existing distribution zone CSI is to be used. If you specify NEW, any existing distribution zone CSI with the specified *cluster* name is deleted, and a new distribution zone CSI is allocated. If you specify OLD, an existing distribution zone CSI is used.

volume

Either the volume serial (volser) identifier for the volume on which the distribution zone CSI is to be allocated or a period (.) if the CSI is to be put on a volume determined by the CICS Transaction Server installation process.

disktype

The UNIT parameter for the volume.

GZONE NEW|OLD options

Specifies whether the global zone to be used already exists.

NEW|OLD

Specifies whether an existing global zone is to be used. The DFHISTAR job as supplied specifies NEW. Optionally, change this to OLD to use an existing global zone. If you specify OLD, CICS Transaction Server is installed into an existing SMP/E global zone.

Specify NEW if you want to preserve your existing releases of CICS Transaction Server in their current SMP/E zones and install the new release in its own zones.

If you specify OLD, the existing SMP/E zones are used and any existing release of the product is deleted.

If you specify OLD, but specify NEW for the GZONECSI parameter, both parameters are assigned the NEW disposition.

options

Specifies the name of the SMP/E options to be used on the SET BOUNDARY command.

TZONE zonenumber

Specifies the name of the target zone.

zonenumber

The name of the target zone to be used by SMP/E. This name must be unique to the target zone. It must not be longer than seven characters, and the leading character must be alphabetic.

DZONE zonenumber

Specifies the name of the distribution zone.

zonenumber

The name of the distribution zone to be used by SMP/E. This name must be unique in the global zone. It must not be longer than seven characters, and the leading character must be alphabetic.

Specifying SMP/E zone and zone log dispositions

As supplied, the DFHISTAR job assumes that you are going to install CICS Transaction Server into new target and distribution zones. However you can specify a new or old global zone, and new or old zone logs by the disposition option NEW|OLD on the associated parameters of the DFHISTAR job.

The disposition option NEW means that the DFHINST3 job deletes any existing zone or zone log with its name specified before redefining it. For example, if you specify the following parameter:

```
GZONELOG CICSTS56.GZONE.SMPLOG NEW
```

the DFHINST3 job deletes any existing SMP/E global zone log with the name CICSTS56.GZONE.SMPLOG before defining a new SMP/E global zone log with that name.

Further, if you specify different dispositions for a zone parameter and its associated zone log parameter, they are both given the default disposition NEW, to ensure that both a zone and its zone log have the same disposition.

If you intend to install CICS Transaction Server using one new CSI for all zones, you must specify the disposition NEW on all three CSI parameters of the DFHISTAR job. For example:

DZONE	DZONE
DZONECSI	CICSTS56.SMPZONE NEW CICSTS56 SYSALLDA
DZONELOG	CICSTS56.DZONE.SMPLOG NEW
GZONE	NEW CICSOPT
GZONECSI	CICSTS56.SMPZONE NEW CICSTS56 SYSALLDA

GZONELOG	CICSTS56.GZONE.SMPLOG NEW
TZONE	TZONE
TZONECSI	CICSTS56.SMPZONE NEW CICSTS56 SYSALLDA
TZONELOG	CICSTS56.TZONE.SMPLOG NEW

Specifying the high-level qualifiers for SMP/E data sets

For each different high-level qualifier that you have specified for SMP/E zone CSIs, logs, and other SMP/E data sets, you must create an ALIAS definition in the main catalog before the data sets can be used.

Specifying the distribution tape device type

Specify the type of device that will load the CICS Transaction Server distribution tape on the TAPEUNIT parameter.

TAPEUNIT devicetype

Specifies the device type to be used to read the distribution tape. Use 3480 for the 3480 tape cartridge, 3400-6 for the 6250 tape, or the unit names in use in your installation.

Specifying attributes of the CICS Transaction Server system data sets

Use the DSINFO parameter to specify the attributes of the CICS Transaction Server system data sets, which are created when you run the postinstallation jobs, DFHCOMDS, DFHDEFDS, EYUCMSDS, EYUWUIDS, and EYUCSYDS

For more information about the postinstallation jobs DFHCOMDS and DFHDEFDS, see [“DFHCOMDS job for common data sets” on page 171](#).

DSINFO dsindex volume disktype qualifier

Defines the following attributes of CICS TS system data sets:

dsindex

Assigns a high-level index to all the data sets defined by the jobs, DFHCOMDS, DFHDEFDS, EYUCMSDS, EYUWUIDS, and EYUCSYDS.

The leading character of *dsindex* must be alphabetic. *dsindex* can have one or two levels of index, but each level must be no longer than eight characters. If you specify more than one level of index, the names must be separated by a period, for example, CICSTS56.CICSHTC1.

volume

The volume identifier of the volume.

disktype

The UNIT parameter for the volume.

qualifier

A partial qualifier added to the index for the data sets created by the jobs DFHCOMDS, DFHDEFDS, EYUCMSDS, EYUWUIDS, and EYUCSYDS. You can specify a partial qualifier of up to four alphanumeric characters; these characters are appended to the characters CICS to make the qualifier. If you specify a period (.), no qualifier is used.

Specifying attributes of any additional target libraries

If you want to create extra copies of the CICS Transaction Server target libraries, specify the attributes of those libraries on the listed parameters.

AINDEX library_prefix

Assigns a high-level index to the additional set of CICS target libraries copied by a version of the DFHINSTA job.

Note:

1. The high-level index for the additional SDFHLINK and SDFHLPA libraries is defined by the ALINDEX parameter.
2. The high-level index for the data sets created by the jobs DFHCOMDS and DFHDEFDS is defined by the *dsindex* operand of the DSINFO parameter.

The AINDEX value must be unique; for example, it must be different from the INDEX value. It must not be longer than 26 characters, and the leading character must be alphabetic. If you specify more than one level of index, the names must be separated by a period; for example, AINDEX CICSTS56.A.TEST.

ALINDEX library_prefix

Assigns a high-level index to the additional SDFHLPA and SDFHLINK libraries allocated by running a version of the DFHINSTA job.

The *library_prefix* value must not be longer than 26 characters and the leading character must be alphabetic. If you specify more than one level of index, the names must be separated by a period; for example, ALINDEX SYS1.CICSTS56.A.TEST.

AZONELOG dsname

Specifies details of the SMP/E log for the additional target zone CSI.

dsname

The name of the additional target zone log to be used by SMP/E.

AZONECSI cluster

Specifies details of the additional target zone CSI. The CSI data set is created on the volume and unit specified by the ADDTVOL parameter.

cluster

The VSAM cluster name, without the qualifier **.CSI**.

AZONE zonename

Specifies the name of the additional target zone, to be used for the set of CICS Transaction Server target libraries copied by a version of the DFHINSTA job.

zonename

The name of the additional target zone to be used by SMP/E. This name must be unique to the target zone. It must not be longer than seven characters and the leading character must be alphabetic.

ASMPSCDS dsname

Specifies the name of the additional zone SMP/E SCDS data set.

dsname

The name of the additional zone SMP/E SCDS data set.

ASMPMTS dsname

Specifies the name of the additional zone SMP/E MTS data set.

dsname

The name of the additional zone SMP/E MTS data set.

ASMPSTS dsname

Specifies the name of the additional zone SMP/E STS data set.

dsname

The name of the additional zone SMP/E STS data set.

ASMP LTS dsname

Specifies the name of the additional zone SMP/E LTS data set.

dsname

The name of the additional zone SMP/E LTS data set.

ADDTVOL volume disktype

Specifies the volume and unit type to contain all the additional zone data sets.

volume

The volume serial identifier of the volume.

disktype

The UNIT parameter for the volume.

Specifying the library data set names

Specify the data set names for the SISPLoad, CSSLIB, SCEELKED, SCEELIB, SCEEBND2, SCEELKEX, SCEEOBJ, SCEECPP, SCLBSID, SEZARPCL, SEZACMTX, SCEECICS, SCEERUN, SCEERUN2, SCEESAMP, SDSNLOAD, SCSQLOAD, SCSQANLE, SCSQCICS, SCSQAUTH, and SCSFMODE0 libraries.

Specifying the data set names of the SISPLoad library

Specify the full data set name, up to 44 characters, of the library that contains ISPLINK; SISPLoad for ISPF Version 4 and above, or ISPLoad for ISPF version 3 and below. For example, SISPLoad SYS1.USERID.SISPLoad changes the SISPLoad library name to SYS1.USERID.SISPLoad. This library is accessed, as read-only, during the installation of CICS Transaction Server.

SISPLoad dsname

Up to 44 characters.

Specifying the data set name of the CSSLIB library

Specify the full data set name, up to 44 characters, of the CSSLIB library. For example, CSSLIB SYS1.USERID.CSSLIB changes the CSSLIB library name to SYS1.USERID.CSSLIB. This library is accessed, as read-only, during the installation of CICS Transaction Server.

CSSLIB dsname

Up to 44 characters.

Specifying the data set name of the SCEELKED library

Specify the full data set name, up to 44 characters, of the SCEELKED library. For example, SCEELKED SYS1.USERID.SCEELKED changes the SCEELKED library name to SYS1.USERID.SCEELKED. This library is accessed, as read-only, during the installation of CICS Transaction Server.

SCEELKED dsname

Up to 44 characters.

Specifying the data set names of the SCEELIB library

Specify the full data set names, up to 44 characters, of the SCEELIB library. For example, SCEELIB SYS1.USERID.SCEELIB changes the SCEELIB library name to SYS1.USERID.SCEELIB. The library is accessed, as read-only, during the installation of CICS Transaction Server.

SCEELIB dsname

Up to 44 characters.

Specifying the data set name of the SCEEBND2 library

Specify the full data set name, up to 44 characters, of the SCEEBND2 library. For example, SCEEBND2 SYS1.USERID.SCEEBND2 changes the SCEEBND2 library to SYS1.USERID.SCEEBND2. This library is accessed, as read-only, during the installation of CICS Transaction Server.

SCEEBND2 dsname

Up to 44 characters.

Specifying the data set name of the SCEELKEX library

Specify the full data set name, up to 44 characters, of the SCEELKEX library. For example, SCEELKEX SYS1.USERID.SCEELKEX changes the SCEELKEX library name to SYS1.USERID.SCEELKEX. This library is accessed, as read-only, during the installation of CICS Transaction Server.

SCEELKEX dsname

Up to 44 characters.

Specifying the data set name of the SCEEOBJ library

Specify the full data set name, up to 44 characters, of the SCEEOBJ library. For example, SCEEOBJ SYS1.USERID.SCEEOBJ changes the SCEEOBJ library name to SYS1.USERID.SCEEOBJ. This library is accessed, as read-only, during the installation of CICS Transaction Server.

SCEEOBJ dsname

Up to 44 characters.

Specifying the data set name of the SCEECPP library

Specify the full data set name, up to 44 characters, of the SCEECPP library. For example, SCEECPP SYS1.USERID.SCEECPP changes the SCEECPP library name to SYS1.USERID.SCEECPP. This library is accessed, as read-only, during the installation of CICS Transaction Server.

SCEECPP dsname

Up to 44 characters.

Specifying the data set name of the SCLBSID library

Specify the full data set name, up to 44 characters, of the SCLBSID library. For example, SCLBSID SYS1.USERID.SCLBSID changes the SCLBSID library name to SYS1.USERID.SCLBSID. This library is accessed, as read-only, during the installation of CICS Transaction Server.

SCLBSID dsname

Up to 44 characters.

Specifying the data set names of the SEZARPCL and SEZACMTX libraries

Specify the full data set names, up to 44 characters, of the SEZARPCL and SEZACMTX libraries. For example, SEZARPCL SYS1.USERID.SEZARPCL changes the SEZARPCL library to SYS1.USERID.SEZARPCL and SEZACMTX SYS1.USERID.SEZACMTX changes the SEZACMTX library name to SYS1.USERID.SEZACMTX. These libraries are accessed, as read-only, during the installation of CICS Transaction Server.

SEZARPCL dsname

Up to 44 characters.

SEZACMTX dsname

Up to 44 characters.

Specifying the data set names of the SCEECICS and SCEERUN libraries

Specify the full data set names, up to 44 characters, of the SCEECICS & SCEERUN libraries. For example, SCEECICS SYS1.USERID.SCEECICS changes the SCEECICS library to SYS1.USERID.SCEECICS and SCEERUN SYS1.USERID.SCEERUN changes the SCEERUN library name to SYS1.USERID.SCEERUN. These libraries are accessed, as read-only, during the installation of CICS Transaction Server.

SCEECICS dsname

Up to 44 characters.

SCEERUN dsname

Up to 44 characters.

Specifying the data set names of the SCEERUN2 library

Specify the full data set names, up to 44 characters, of the SCEERUN2 library. For example, SCEERUN2 SYS1.USERID.SCEERUN2 changes the SCEERUN2 library to SYS1.USERID.SCEERUN2. This library is accessed, as read-only, during the installation of CICS Transaction Server.

SCEERUN2 dsname

Up to 44 characters.

Specifying the data set name of the SCEESAMP library

Specify the full data set name, up to 44 characters, of the SCEESAMP library. For example, SCEESAMP SYS1.USERID.SCEESAMP changes the SCEESAMP library name to SYS1.USERID.SCEESAMP. This library is accessed, as read-only, during the installation of CICS Transaction Server.

SCEESAMP dsname

Up to 44 characters.

Specifying the data set name of the SDSNLOAD library

Specify the full data set name, up to 44 characters, of the Db2 SDSNLOAD library. For example, SDSNLOAD SYS1.USERID.SDSNLOAD changes the SDSNLOAD library name to SYS1.USERID.SDSNLOAD. This library is accessed, as read-only, during the installation of CICS Transaction Server.

The REXX for CICS element contains some modules that are link-edited against the Db2 load library, SDSNLOAD. If you do not have Db2 installed, the DFHINST6 job fails because SDSNLOAD cannot be allocated to the job. To avoid this problem, define a dummy SDSNLOAD data set, with LRECL=0 and RECFM=U, and specify the name of this empty data set on the SDSNLOAD parameter.

SDSNLOAD dsname

Up to 44 characters.

Specifying the data set name of the SCSQLOAD library

Specify the full data set name, up to 44 characters, of the SCSQLOAD library. For example, SCSQLOAD SYS1.USERID.SCSQLOAD changes the SCSQLOAD library name to SYS1.USERID.SCSQLOAD. This library is accessed, as read-only, during the installation of CICS Transaction Server.

SCSQLOAD dsname

Up to 44 characters.

Specifying the data set name of the SCSQANLE library

Specify the full data set name, up to 44 characters, of the SCSQANLE library. For example, SCSQANLE SYS1.USERID.SCSQANLE changes the SCSQANLE library name to SYS1.USERID.SCSQANLE. This library is accessed, as read-only, during the installation of CICS Transaction Server.

SCSQANLE dsname

Up to 44 characters.

Specifying the data set name of the SCSQCICS library

If you want to run the IBM MQ for z/OS sample programs, specify the full data set name, up to 44 characters, of the SCSQCICS library. For example, SCSQCICS SYS1.USERID.SCSQCICS changes the SCSQCICS library name to SYS1.USERID.SCSQCICS. This library is accessed, as read-only, during the installation of CICS Transaction Server.

SCSQCICS dsname

Up to 44 characters.

Specifying the data set name of the SCSQAUTH library

Specify the full data set name, up to 44 characters, of the SCSQAUTH library. For example, SCSQAUTH SYS1.USERID.SCSQAUTH changes the SCSQAUTH library name to SYS1.USERID.SCSQAUTH. This library is accessed, as read-only, during the installation of CICS Transaction Server.

SCSQAUTH dsname

Up to 44 characters.

Specifying the data set names of the SCSFMODE and SIXMEXP libraries

Specify the full data set names, up to 44 characters, of the SCSFMODE and SIXMEXP libraries. For example, SCSFMODE0 SYS1.SCSFMODE0 changes the SCSFMODE0 library to SYS1.SCSFMODE0. These libraries are accessed, as read-only, during the installation of CICS Transaction Server.

SCSFMODE dsname

Up to 44 characters.

SIXMEXP dsname

Up to 44 characters.

Specifying log stream and log stream structure attributes

Specify attributes of the CICS Transaction Server log streams and the coupling facility structures that you use when you run the postinstallation jobs DFHILG1, DFHILG2, DFHILG3 and DFHILG4, on the LOGGER-INFO parameter.

For more information about these postinstallation jobs, see [Defining the logger environment for CICS](#).

LOGGER-INFO strsfx logsz shuntsz jnsz gensz sysname loghlq logmodel

Defines the following attributes of CICS Transaction Server system data sets:

strsfx

The last part of the coupling facility structure names, can be any three characters allowed in a structure name. The default is 001. It is used in DFHILG1, DFHILG2, DFHILG3, and DFHILG4.

logsz

The average buffer size for system log streams in the LOG_DFHLG_strsfx structure. The default is 500. It is used in DFHILG1.

shuntsz

The average buffer size for shunted system log streams in the LOG_DFHSHUNT_strsfx structure. The default is 4096. It is used in DFHILG1.

jnsz

The average buffer size for unforced user journal log streams in the LOG_USERJRNL_strsfx structure. The default is 64000. It is used in DFHILG1.

gensz

The average buffer size for forced user journal log streams and forward recovery log streams in the LOG_GENERAL_strsfx structure. The default is 2048. It is used in DFHILG1.

sysname

The MVS system name used to create model log streams for DFHLOG and DFHSHUNT. The default is MVSX. It is used in DFHILG2 and DFHILG5.

loghlq

The first qualifier of the model name for general logs and DFHLGLOG. It is used in DFHILG3, DFHILG4, DFHILG6, and DFHILG7.

logmodel

The second qualifier of the model name for general logs. It is used in DFHILG3 and DFHILG6.

Specifying attributes specific to CICSplex SM

Use these DFHISTAR attributes to customize the postinstallation JCL for CICSplex SM. This JCL is used in the installation verification procedure for CICSplex SM.

All attributes specific to CICSplex SM have defaults. If you do not want to use CICSplex SM, you can run DFHISTAR without providing overrides for any of the following CICSplex SM specific parameters:

CMASNAME value

Specifies the 1- to 8-character name to be allocated to a CMAS. The name can contain alphabetic, national, and numeric characters. However, the first character must be alphabetic or national. The default is CMAS01.

The name of a CMAS must be unique in the CICSplex SM environment. It must not be the same as the name of another CMAS, a CICSplex, a CICS system, or a CICS system group.

CMCIPIRT value

Specifies the numeric identifier allocated to the TCP/IP port number for the CICS management client interface (CMCI) on the WUI server. The identifier can contain numeric characters only, in the range 1 to 65535. The default is 12346. The port number must be unique and used by only one WUI Server. The CICSplex SM WUI Server does not support port sharing.

CMSSYSID value

Specifies the 4-character system identifier of the CMAS. This identifier can contain alphabetic, national, and numeric characters. It must match the SYSIDNT system initialization parameter for the CMAS. The default is CM01.

CSYSNAME name

Specifies the 1- to 8-character name to be allocated to a MAS. The name can contain alphabetic, national, and numeric characters. However, the first character must be alphabetic or national. The default is CSYS01.

The name of a MAS must be unique in the CICSplex SM environment. It must not be the same as the name of another MAS, a CICSplex, a CICS system, or a CICS system group.

CSYSPLEX value

Specifies the 1- to 8-character name to be allocated to a CICSplex of managed systems. This identifier can contain alphabetic, national, and numeric characters. The default is CSYPLX01.

The name of a CICSplex must be unique in the CICSplex SM environment. It must not be the same as the name of another CICSplex, a CICS system, or a CICS system group.

CSYSYSID value

Specifies the 1- to 4-character system identifier for the managed CICS system. This identifier can contain alphabetic, national, and numeric characters. The default is CS01.

OLDDREP dsname

Specifies an existing data repository that is being used by a previous release of CICSplex SM. The records in the existing data repository are migrated to a new data repository for CICS TS for z/OS, Version 5.6 . The existing data repository is not modified. If you do not specify this parameter, a new data repository is created.

dsname

The name of the existing data repository.

The new CICS TS for z/OS, Version 5.6 data repository has the name

dsinfo.EYUDREP.masname

where:

dsinfo

Is the index specified with the DSINFO parameter.

masname

Is the name specified with the CMASNAME parameter.

NEWDREP-PREFIX dsname

Specifies a prefix for a new data repository that is to be used by CICSplex SM.

dsname

The new CICS TS for z/OS, Version 5.6 data repository has the name:

newdrep-prefix.masname

where:

masname

Is the name specified with the CMASNAME parameter.

If a period (the default value) is used, then the name of the new CICS TS for z/OS, Version 5.6 data repository has the name:

dsinfo.EYUDREP.masname

where:

dsinfo

Is the *dsindex* attribute of that which is specified by the DSINFO parameter.

masname

Is the CMAS name specified by the CMASNAME parameter.

TCPIPHST

Specifies the TCP/IP host name for the WUI server. The default is
XXXXXXXX.XXXXXXXXXX.XXXXXXXXXX.XXXXXXXXXX.

TCPIPPRT

Specifies the numeric identifier allocated to the TCP/IP port number for the WUI server. The identifier can contain numeric characters only, in the range 1 to 65535. The default is 12345. The port number must be unique and used by only one WUI Server. The CICSplex SM WUI Server does not support port sharing.

TIMEZONE code

Specifies the time zone assigned to the data repository. This code must be a single alphabetic character in the range B through Z. The default is B.

WUI value

Specifies whether to create a WUI CICSplex. This parameter is ignored if the OLDDREP parameter is specified. The default is YES if OLDDREP is not specified.

YES

Create a WUI CICSplex.

NO

Do not create a WUI CICSplex.

WUINAME name

Specifies the 1- to 8-character name allocated to a WUI. The name can contain alphabetic, national, and numeric characters. However, the first character must be alphabetic or national. The default is WUINCM01.

WUIPLEX name

Specifies the 1- to 8-character name allocated to a WUI CICSplex. The name can contain alphabetic, national, and numeric characters. However, the first character must be alphabetic or national. This parameter is ignored if the OLDDREP parameter is specified. The default is created from the characters WUIP, followed by the CMSSYSID. For example, using the default CMSSYSID, CM01, the default WUIPLEX name is WUIPCM01.

WUISYSID name

Specifies the 1- to 4-character name allocated to a WUI system identifier. The name can contain alphabetic, national, and numeric characters. However, the first character must be alphabetic or national. The default value is WU01.

Specifying the DFHISTAR DOWNLOAD parameter

The DOWNLOAD parameter is populated into the DFHINSTE job that is created when running DFHISTAR.

Creating RACF profiles for the CICS Transaction Server data sets

Your Security Administrator creates appropriate RACF profiles for the CICS Transaction Server data sets.

At this stage, you require authority to access only the data set qualifiers specified on the TEMPLIB and LIB parameters of the DFHISTAR job. DFHISTAR uses a temporary sequential data set, with the high-level qualifier specified on the INDEX parameter, to resolve the parameters to be substituted into the jobs

being tailored. However, consider coordinating the access authority for all the CICS Transaction Server data sets at the same time.

RACF profiles are discussed in detail in the [RACF profiles](#).

Running the DFHISTAR job

Edit and save the DFHISTAR job with the values of installation parameters for your CICS Transaction Server environment. When you are ready to tailor the skeleton jobs, submit the DFHISTAR job.

The highest expected return code is 0.

When the DFHISTAR job has run, the Program Directory shows, in tabular form, the non-DFHISTAR jobs that have been copied and are tailored to your CICS Transaction Server environment, and are added to the library that you specified on the LIB parameter of the DFHISTAR job. By default, this library is the CICSTS56.XDFHINST library. If necessary, the DFHISTAR job creates the library specified on the LIB parameter.

Checking the output from the DFHISTAR job

Check the output from the DFHISTAR job and, if required, edit and submit the DFHISTAR job again.

The DFHISTAR job produces a job log and, if necessary, an error code:

- The output job log lists the values that were used for the parameters of the DFHISTAR job.
- If any error occurs when running the DFHISTAR job, a warning code of 4 or an error code of 12 is returned. For error code 4, the skeleton jobs are tailored and added to the CICSTS56.XDFHINST library. For error code 12, the skeleton jobs are not tailored or copied. To resolve the cause of either error code, examine the output job log and, if necessary, edit and submit the DFHISTAR job again.

You can run the DFHISTAR job any number of times to alter the attributes of the jobs that it creates.

When running the DFHISTAR job after the first time, you can select specific jobs to be created by using the SCOPE or SELECT parameter:

SCOPE ALL|BASE|POST

Specifies whether you want to generate all the CICS Transaction Server installation and postinstallation jobs, or only the postinstallation jobs. When you install CICS Transaction Server from the distribution tape, specify the default, SCOPE ALL. Code the other options, if necessary, during postinstallation tasks.

ALL

Specifies that you want to generate all the CICS Transaction Server installation jobs and all the postinstallation jobs.

BASE

Specifies that you want to generate only the installation jobs (DFHINST1 through DFHINST6, DFHIHFS0, DFHIHFS1, and DFHISMKD) that you use to install CICS Transaction Server from the distribution tape.

POST

Specifies that you want to generate only the postinstallation jobs that you can use to create the CICS Transaction Server data sets, and run the IVPs.

Running the installation jobs

Check that you are ready to run the installation jobs and then submit the jobs in sequence.

Before you run the installation jobs, read and run the included README file. Ensure the following settings are in place:

- The MVS image was IPLed with OMVS in full-function mode.
- The user ID under which you are running the jobs has superuser authority.

After you have run the DFHISTAR job to create the installation jobs, submit those jobs in sequence to install CICS Transaction Server. The following topics describe the CICS Transaction Server installation jobs, and give guidance on how to use them.

The CICS Transaction Server jobs are in the CICSTS56.XDFHINST library as a result of running the DFHISTAR job, which you copied from the distribution tape, as described in *Copy RELFILE(2) from the Distribution Tape* in the Program Directory.

Run these jobs one at a time. Before you run a job, read the information about it, starting with [“Running the DFHIHFS0 job”](#) on page 47.

After you have run a job, check its output before proceeding to the next job. If a job ends abnormally, find out why it failed by looking at the job log, which lists the error messages produced on each run. Correct the error and then proceed as advised in the job description. Do not attempt to run the next job until the previous job has run successfully.

Important: If you are installing electronically, rather than from a distribution tape, you should run the DFHINSTE job instead of DFHINST5. If you attempt to run DFHINST5 it will return an error.

Running the DFHIHFS0 job

This job create a file system and cicsts directory.

- Creates the *cicsts* directory at */pathprefix/usr/lpp/*.
- Mounts the file system at directory */pathprefix/usr/lpp/cicsts*.
- Creates the *dfhconfig* directory at */pathprefix/usr/lpp/cicsts/*.
- Creates empty file *featuretoggle.properties* in the *dfhconfig* directory.
- Owner=RWX
- Group=RWX
- Other=R-X

(In octal form: 775)

where:

R

Read

W

Write

X

Execute

-

No permission

Note:

1. You might not need to run DFHIHFS0 if you have installed an earlier release of CICS TS or if you are re-installing CICS TS 4.1, because CICS creates the */cicsts* directory that is common to all releases of CICS. If you do need to run DFHIHFS0, run this job once only.
2. You must grant RACF ALTER ACCESS to the OMVS data sets before running DFHIHFS0.
3. The */cicsts* directory contains only directories, each being a mount point.
4. CICS requires the MOUNT issued by DFHIHFS0 to access files stored in the file system, but the MOUNT command is lost when you re-IPL MVS. SDFHINST member DFHBPXP0 contains a MOUNT command for inclusion in a BPXPRMxx member of the SYS1.PARMLIB data set. The MOUNT command applies to the data set specified in the HFS0DSN parameter of the DFHISTAR job to be mounted at directory */pathprefix/usr/lpp/cicsts*. Copy this command into a BPXPRMxx member of the SYS1.PARMLIB data set to ensure the mount is restored when MVS is IPLed.
5. All steps of DFHIHFS0 must end with return code zero for the job to be successful.

Running the DFHIHFS1 job

This job unmounts and deletes the file system, before creating and mounting a new file system as specified in DFHISTAR.

1. Unmounts the file system at directory `/pathprefix/usr/lpp/cicsts/ussdir` to allow the job to be rerun, and if necessary forces return code zero.
2. Deletes from `/pathprefix/usr/lpp/cicsts` the directory defined by the `/ussdir` parameter of the DFHISTAR job. This allows the job to rerun and, if necessary, forces return code zero.
3. Deletes the file system specified in the HFS1DSN parameter of the DFHISTAR job to allow the job to rerun and, if necessary, forces return code zero.
4. Creates the file system specified in the HFS1DSN parameter of the DFHISTAR job.
5. Creates the `/ussdir` directory at `/pathprefix/usr/lpp/cicsts`, where `/ussdir` is the name of the directory specified on the `ussdir` parameter.
6. Mounts the file system at directory `/pathprefix/usr/lpp/cicsts/ussdir`.
7. Creates the `dfhconfig` directory at `/pathprefix/usr/lpp/cicsts/ussdir`.
8. Creates empty file `featuretoggle.properties` in the `dfhconfig` directory.
9. Changes the permission settings for the `/ussdir` directory and the `ussdir/dfhconfig` directory to 775.

All steps of DFHIHFS1 must end with return code zero for the job to be successful.

CICS requires the MOUNT issued by DFHIHFS1 to access files stored in the z/OS UNIX file system, but the MOUNT command is lost when you re-IPL MVS. SDFHINST member DFHBXP1 contains a MOUNT command for `/pathprefix/usr/lpp/cicsts/ussdir`, where `ussdir` is the name of the directory specified in the `ussdir` parameter in the DFHISTAR job. Copy this command into a BPXPRMxx member of the SYS1.PARMLIB data set to ensure the mount is restored when MVS is IPLed.

Running the DFHISMKD job

This job creates the UNIX System Services directories.

You must run this job before any of the other installation jobs.

The highest expected return code is 0.

Running the DFHINST1 job

This job allocates and catalogs CICS Transaction Server distribution and target libraries.

To ensure that you can rerun this job, it deletes and uncatalogs the data sets that are allocated in the second step of the job.

If the DFHINST1 job ends abnormally, examine the job log to determine the cause, correct the problem, and then rerun the job.

The highest expected return code is 0.

Running the DFHINST2 job

This job allocates the CICS Transaction Server RELFILE data sets. If you run the DFHINST2 job now, you ensure that enough space has been allocated to the RELFILE data sets to allow the DFHINST5 job to complete.

To ensure that you can rerun this job, it deletes and uncatalogs the data sets, if they exist, that it allocates later.

If the DFHINST2 job ends abnormally, examine the job log to determine the cause, correct the problem, and then rerun the job.

The highest expected return code is 0.

Running the DFHINST3 job

This job allocates the CICS Transaction Server SMP/E data sets.

Caution: If you intend to use an existing target or distribution zone that contains an earlier release of CICS Transaction Server elements, be aware that any earlier release of CICS is cleared before being replaced by CICS Transaction Server.

Before you run the DFHINST3 job, if you intend to install CICS Transaction Server using both existing and new CSIs, make sure any new CSIs have the same control interval size as the existing CSIs.

If your existing CSIs do not have a control interval size of 4096 bytes, edit the DFHINST3 job before running it to change the CONTROLINTERVALSIZE(4096) parameter on the commands used to create the VSAM data sets for the new CSIs, to specify the same control interval size as the existing CSIs.

For further information about allocating CSI data sets, see [SMP/E for z/OS Reference](#).

To ensure that you can rerun this job, it deletes and uncatalogs the data sets, if they exist, that it allocates later.

This job also sets up the global, target, and distribution zones, depending on the parameters that you specified to the DFHINST3 job:

1. If you specified NEW for GZONE, the global zone is deleted and redefined.
2. The distribution zone is deleted and redefined.
3. The target zone is deleted and redefined.
4. Member GIMZPOOL from SYS1.MACLIB is copied using the REPRO command into the zones redefined in the previous steps.
5. If you specified OLD for GZONE, the entries for the DZONE and TZONE names are removed from the global zone.

If the DFHINST3 job ends abnormally, examine the job log to find the cause, correct the problem, and then rerun the job.

The highest expected return code is 0.

Running the DFHINST4 job

This job primes the global zone, target zone, and distribution zone, which are the new SMP/E zones created in DFHINST3..

If you did not run the DFHINST2 job, increase the DSSPACE values in the DFHINST4 job before submitting it.

If the DFHINST4 job ends abnormally, examine the job log to determine the cause, correct the problem, and then repeat all jobs, beginning with DFHINST1. This correction avoids SMP/E space problems, and consequent X37 stops with an abend message, during reruns of these SMP/E jobs.

The highest expected return code is 0, if you install into new zones, and 8, if you are installing into existing zones.

Running the DFHINST5 job

This job uses the RECEIVE command to receive the CICS Transaction Server software from the distribution tape into the RELFILE data sets created by the DFHINST2 job. This installation job, apart from the initial IEBCOPY job, is the only job that requires the distribution tape to be mounted.

If the DFHINST5 job ends abnormally, examine the job log to determine the cause, correct the problem, and then repeat all jobs, beginning with DFHINST1. This correction avoids SMP/E space problems, and consequent X37 stops with an abend message, during reruns of these SMP/E jobs.

The highest expected return code is 0.

Important: If you are installing electronically, rather than from a distribution tape, you should run the DFHINSTE job instead of DFHINST5. If you attempt to run DFHINST5 it will return an error.

Running the DFHINSTE job

IMPORTANT: Run this job instead of the DFHINST5 job if you are installing electronically rather than from a distribution tape.

If the DFHINSTE job ends abnormally, examine the job log to determine the cause, correct the problem, and then repeat all jobs, beginning with DFHINST1. This correction avoids SMP/E space problems, and consequent X37 stops with an abend message, during reruns of these SMP/E jobs.

The highest expected return code is 0.

Running the DFHINST6 job

This job performs the SMP/E APPLY and ACCEPT functions that install CICS Transaction Server into the target and distribution libraries respectively.

To ensure that DFHINST6 runs successfully, perform the following steps:

- Ensure that the Db2 SDSNLOAD library is available for this job. Without SDSNLOAD, DFHINST6 fails; see [“Specifying the data set name of the SDSNLOAD library” on page 42](#).

If you run DFHINST6 with an empty SDSNLOAD library, the APPLY step completes with return code 4, and the return code from the binder is 8. The REXX for CICS modules, CICSQL and CICDB2, which interface with Db2, are stored in SCICLOAD without the required Db2 routines and are not usable.

- Run DFHINST6 on the same MVS image on which the z/OS UNIX file system is installed, unless you are set up for MVS sharing.

If you have modified the other installation jobs (for example, to use existing libraries and therefore existing target and distribution zones), consider splitting the DFHINST6 job to do APPLY CHECK, APPLY, ACCEPT CHECK, and ACCEPT functions as four separate jobs.

The DFHINST6 job is the longest running of all the installation jobs. It produces a large amount of printed output. The region size for the DFHINST6 job is currently set to 'REGION=0M', because this job requires more memory than the other installation jobs. Adjust your JES parameters (for example, with a JES2 / *JOBPARM LINES=99 statement) to avoid a system abend 722.

If successful, this job gives a return code of 4. See the "GIM23903W - LINK SUCCESSFUL . . ." message, listed in the report that is output by the apply job. DFHINST6 job issues messages GIM23903W and GIM23913W depending on the execution environment of the installer. Both messages are acceptable.

The binder produces IEW2454W messages during the APPLY stage for unresolved external references while some CICS Transaction Server load modules are being link-edited during installation, giving return code 4. You might also receive numerous IEW2646W and IEW2651W messages, which indicate conflicts with user-specified RMODE and AMODE modes respectively. You can ignore these IEWxxxx messages, which are produced for component object modules of executable CICS Transaction Server load modules.

Messages IEW2689W, IEW2470E, IEW2648E, and IEW2303E might be displayed. You can ignore them. When you have run the DFHINST6 job, the following SMP/E message is produced from the job:

```
GIM20502I  GIMSMP PROCESSING IS COMPLETE - THE HIGHEST RETURN CODE WAS 04 -
```

You can ignore this SMP/E message.

If any other SMP/E messages appear, see [SMP/E for z/OS Messages, Codes, and Diagnosis](#) for guidance information about their meaning and take the appropriate action.

If the DFHINST6 job finishes abnormally, examine the job log to determine the cause, correct the problem, and then repeat all the jobs, beginning with DFHINST1. This correction avoids SMP/E space problems, and consequent X37 abends, during reruns of these SMP/E jobs.

If the DFHINST6 job fails and you are using an existing global zone (that is, you specified the GZONE parameter of the DFHISTAR job with the disposition parameter OLD), perform the following steps:

1. REJECT the CICS Transaction Server base-level function SYSMOD.
2. Rerun the DFHINST1 job. When you rerun the installation jobs, some steps that were successfully completed in the previous run produce return codes with a value of 8.

Running the DFHIJVMJ job

The DFHIJVMJ job creates the customized sample JVM profiles that you require if you want to run a JVM program in CICS.

DFHIJVMJ reads the supplied JVM profiles in the partitioned data set SDFHENV. It replaces the symbol `&JAVA_HOME` in the files with the value you specify on the **JAVADIR** parameter in the DFHISTAR installation job. The extra `//` characters on each side of the symbol in the supplied files are removed during symbol substitution.

The customized JVM profiles are then written as z/OS UNIX files in the directories listed in [“Verifying Java components checklist”](#) on page 191.

Running the DFHIFTG1 job

The DFHIFTG1 job is an optional job that tags the text files in the CICS USSHOME directory with the correct coded character set.

The expected highest return code is 0.

Running the DFHCSV CJ job

The DFHCSV CJ job runs the DFHCSV CU utility program to dynamically update the MVS SVC number that is specified in the PARM statement to invoke the required module, and so removes the need to restart the z/OS LPAR in order to use a new or updated SVC. You can use the DFHCSV CU utility program only if you have the required authorization.

The DFHCSV CU utility updates or adds a single SVC entry in the SVC table with a pointer containing the entry point of the supplied module. For an existing SVC number, before the SVC is updated, all CICS regions using that SVC number must be shut down, otherwise results can be unpredictable.

Before the DFHCSV CU utility program is run, the target program must be loaded into the MVS link pack area (LPA) using the following MVS console command:

```
SETPROG LPA,ADD,MODNAME=module,DSNAME=data set
```

Where:

- *module* is the module to be invoked by the SVC and
- *data set* is the load library where the module resides.

Because this utility issues SVCs (supervisor calls) it must be invoked from an authorized library by an operator who has sufficient authority to run the program.

The **SVCnnn=module** parameter identifies the number of the SVC to be modified and the program to be invoked by the SVC. The following checks are made by the utility program before any update is attempted:

1. A PARM statement is specified on the EXEC JCL card.
2. The PARM statement contains an SVCnnn= keyword statement.
3. The number specified is greater than 199 and less than 256.
4. The SVC type for the specified SVC number is 3 or 4.
5. The module name is less than or equal to 8 characters in length.

If any of these checks fail, the utility ends with the return code set to 12. If you do not load the target program into the MVS LPA, the utility ends with the return code set to 28.

When an existing SVC is to be updated the utility asks the operator to confirm that the SVC update is to proceed. Any response other than Yes results in the utility ending with the return code set to 12.

Note: Updates that are made by the DFHCSVCU utility are temporary, and are only valid until the system is restarted or another instance of the utility program is executed. For the updates to be permanent, member IEASVCxx in SYS1.PARMLIB must be updated with the changes.

Checking the output from the installation jobs

When you have successfully run all of the installation jobs, CICS Transaction Server is loaded.

You now have CICS Transaction Server installed on your DASD. Back up the volume on which CICS Transaction Server resides. If any errors occur during customization later, you do not have to re-run the installation jobs.

Postinstallation activities

Copy the CICS Transaction Server procedures into a cataloged procedure library, load any CICS features that you have, and tailor the CICS Transaction Server.

For information about tailoring CICS Transaction Server, see [“Tailoring the CICS-supplied skeleton jobs”](#) on page 167.

Copying the CICS Transaction Server procedures into a procedure library

CICS Transaction Server supplies procedures that can be tailored or are copied directly to your procedure library when you run the installation jobs. All procedures are described in the table.

The procedures DFHAUPLE, DFHSMPE, DFHSTART, EYUCMASP, EYUCSYSP, and EYUWUIP are tailored to your CICS Transaction Server environment and stored in the CICSTS56.XDFHINST library when you run the DFHISTAR job. The other procedures are not modified by the DFHISTAR job and are copied into the CICSTS56.SDFHPROC library when you run the CICS Transaction Server installation jobs.

Copy all these procedures into a cataloged procedure library; for example, SYS1.PROCLIB. Before you copy the procedures, read the following instructions:

1. Your procedure library might already contain procedures, supplied with an earlier release of CICS, that have the same names as the new procedures but are, in fact, different. If so, you must find some way of selecting the right release. Here are some ways of using the new versions:
 - a. For the time being, rename either set of procedures and modify the appropriate jobs to use the new names.
 - b. Insert the new procedures into the job streams that use them and use the procedures as in-stream procedures. Place the inserted procedures between the JOB statement and the first EXEC statement. You must insert a // PEND statement after the inserted procedures. When the new release becomes the production system, you can copy the new procedures into your procedure library.

- c. Indicate the DDNAME of the cataloged procedure library that is to be used to convert the JCL for the job. For example, you could use the JES2 /*JOBPARM PROCLIB=xxxxxxx. For more information about specifying DDNAMEs in JCL, see the [z/OS MVS JCL Reference](#).
 - d. Specify, in the JCLLIB statement, the name of the procedure libraries that you want to search for the named procedure or procedures. For more information about JCLLIB, see the [z/OS MVS JCL Reference](#).
2. If service is applied to the CICS Transaction Server procedures, the versions in the libraries CICSTS56.CICS.SDFHINST and CICSTS56.CICS.SDFHPROC are updated by SMP/E. You must then copy the updated procedures into your procedure library.
 3. The default for the symbolic parameter GZONE in the procedures DFHSMPE and DFHAUPLE is taken from the value that you specified by the GZONE parameter of the DFHISTAR job.
 4. The default for the ZNAME symbolic parameter in the procedures DFHSMPE and DFHAUPLE is taken from the value that you specified by the TZONE parameter of the DFHISTAR job. For a description of how the ZNAME parameter is used, see the SMPCTL DD statement in [The CICS TS-supplied SMP/E procedure](#).
 5. Change the OUTC parameter as required.

When you have read these instructions, and acted on them as necessary, copy the procedures into a cataloged procedure library. The CICS Transaction Server-supplied procedures are listed in [Table 6](#) on [page 53](#).

CICS-supplied procedures

The CICS-supplied procedures and their descriptions are listed in an alphabetical table. Copy the procedures from the SDFHPROC library, unless the procedure specifies a different library.

<i>Table 6. CICS-supplied procedures</i>	
Procedure	Description
DFHASMVS	Assembles some CICS Transaction Server programs and user-written assembler language programs.
DFHAUPLE	Assembles and link-edits CICS Transaction Server control tables and makes the assembly and link-edit information available to SMP/E. DFHAUPLE is installed in SDFHINST.
DFHBMSU	Runs the BMS load module disassemble utility program, DFHBMSUP.
DFHEBTAL	Translates, assembles, and link-edits assembler application programs using EXEC DLI commands in a batch environment under Language Environment®.
DFHEGTAL	Translates, assembles, and link-edits AMODE(64) assembler application programs using the command-level interface.
DFHEITAL	Translates, assembles, and link-edits AMODE(24) and AMODE(31) assembler application programs using the command-level interface.
DFHEXTAL	Translates, assembles, and link-edits assembler application programs using the external CICS Transaction Server interface.
DFHLNKVS	Link-edits CICS Transaction Server programs and application programs.
DFHMAPS	Prepares physical and symbolic maps.
DFHMAPT	Prepares physical and symbolic maps for C++.
DFHSMPE	Runs SMP/E. DFHSMPE is installed in SDFHINST.
DFHSTART	Starts CICS. DFHSTART is installed in SDFHINST.
DFHYBTPL	Translates, compiles, and link-edits PL/I application programs using EXEC DLI commands in a batch environment under Language Environment.

Table 6. CICS-supplied procedures (continued)

Procedure	Description
DFHYBTVL	Translates, compiles, and link-edits COBOL application programs using EXEC DLI commands in a batch environment under Language Environment.
DFHYITDL	Translates, compiles, and link-edits C/370 application programs using the command-level interface under Language Environment.
DFHYITEL	Translates, compiles, and link-edits C++ application programs using the command-level interface under Language Environment.
DFHYITFL	Translates, compiles, and link-edits C application programs using the XPLINK compiler option under Language Environment.
DFHYITGL	Translates, compiles, and link-edits C++ application programs using the XPLINK compiler option under Language Environment.
DFHYITPL	Translates, compiles, and link-edits PL/I application programs using the command-level interface under Language Environment.
DFHYITVL	Translates, compiles, and link-edits COBOL application programs using the command-level interface under Language Environment.
DFHYXTDL	Translates, compiles, and link-edits C/370 application programs using the external CICS Transaction Server interface under Language Environment.
DFHYXTEL	Translates, compiles, and link-edits C++ application programs using the external CICS Transaction Server interface under Language Environment.
DFHYXTPL	Translates, compiles, and link-edits PL/I application programs using the external CICS Transaction Server interface under Language Environment.
DFHYXTVL	Translates, compiles, and link-edits COBOL application programs using the external CICS Transaction Server interface under Language Environment.
DFHZITCL	Translates, compiles, and link-edits COBOL application programs using the integrated CICS translator.
DFHZITDL	Translates, compiles, and link-edits C application programs using the integrated CICS translator.
DFHZITEL	Translates, compiles, and link-edits C++ application programs using the integrated CICS translator.
DFHZITFL	Translates, compiles, and link-edits C XPLINK application programs using the integrated CICS translator.
DFHZITGL	Translates, compiles, and link-edits C++ XPLINK application programs using the integrated CICS translator.
DFHZITPL	Translates, compiles, and link-edits PL/I application programs using the integrated CICS translator.
DFHZXTCL	Translates, compiles, and link-edits EXCI COBOL application programs using the integrated CICS translator.
DFHZXTDL	Translates, compiles, and link-edits EXCI C application programs using the integrated CICS translator.
DFHZXTEL	Translates, compiles, and link-edits EXCI C++ application programs using the integrated CICS translator.
DFHZXTPL	Translates, compiles, and link-edits EXCI PL/I application programs using the integrated CICS translator.

<i>Table 6. CICS-supplied procedures (continued)</i>	
Procedure	Description
EYUCMASP	Starts a CMAS.
EYUCSYSP	Starts a MAS.
EYUEITAL	Assembler sample procedure provided in the SEYUPROC library to create replacement modules for EYU9WRAM
EYUEITDL	C sample procedure provided in the SEYUPROC library to create replacement modules for EYU9WRAM
EYUEITPL	PL/I sample procedure provided in the SEYUPROC library to create replacement modules for EYU9WRAM
EYUEITVL	COBOL sample procedure provided in the SEYUPROC library to create replacement modules for EYU9WRAM
EYUJXBTP	JCL procedure used by the samples, EYUJXBT1, EYUJXBT2, EYUJXBT3, and EYUJXBT4, to invoke the EYU9XDBT program
EYUJXDDP	Contains procedural JCL for running the DLA.
EYUWUIP	Starts a WUI.
ICCFCC	CICS foundation classes.
ICCFCL	CICS foundation classes.
ICCFGL	CICS foundation classes.
ICCFCL	CICS foundation classes.

For further information about using the DFHSMPE and DFHSTART procedures, see [“The CICS TS-supplied SMP/E procedure” on page 158.](#)

Creating extra sets of CICS Transaction Server target libraries (optional)

You can use the CICS Transaction Server installation job, DFHISTAR, to generate an optional installation job, DFHINSTA, which you can use to create extra copies of the CICS Transaction Server target libraries and UNIX System Services directories.

About this task

What are the benefits of using multiple libraries

- Backing out PTFs and APARs. If you apply PTFs or APARs to CICS Transaction Server and if they fail a fix-test, you can back out the changes with minimum disruption.
- DASD failure. Multiple libraries protect you against failure of the DASD on which the CICS Transaction Server load libraries reside.

What are the factors you should consider when making the decision to use multiple libraries

- Your requirement for high availability. As already stated, the use of multiple libraries can protect you against CICS Transaction Server downtime caused by DASD failure or incorrect service, either from IBM-supplied PTFs or from your own modifications to your CICS Transaction Server region.
- The extra DASD required. Multiple libraries require more disk space.
- Other ways of providing high availability. For example, use a CICSplex, z/OS Communications Server persistent sessions, and MVS functions to provide restart of CICS Transaction Server regions.
- The added complexity of maintaining multiple sets of CICS Transaction Server libraries. Two or more sets of CICS Transaction Server target libraries, together with the SMP/E procedures that are

required to support them, increase the complexity of maintenance. Define procedures to ensure that upgrades to the CICS Transaction Server libraries are kept under control.

- Alternative solutions. If you have already established a proven process for fix verification and for testing applications developed for your production CICS Transaction Server region, you might decide that you do not require multiple CICS Transaction Server libraries.

You can use the DFHINSTA job, generated by the DFHISTAR job, to create extra sets of CICS Transaction Server target libraries fully under the control of SMP/E. Each time you run the DFHINSTA job, you can generate only one extra set of target libraries.

To create an extra sets of target libraries, complete the following procedure. You can repeat the steps to create more sets of target libraries.

Procedure

1. Edit the DFHISTAR job to specify values:

- The ADDTVOL, AINDEX, ASMPSCDS, AZONE, AZONECSI, AZONELOG, and USSDIRA parameters, for the new set of target libraries.
- The INDEX, TZONE, TZONECSI, and TZONELOG parameters, for the primary target libraries you want to copy from. The TZONE, TZONECSI, and TZONELOG parameters must specify the target zone that contains the CICS Transaction Server target libraries defined with the high-level qualifier provided by the INDEX parameter.
- The DZONE, DZONECSI, and DZONELOG parameters, for the distribution libraries to be associated with the new set of target libraries.

For further information about editing the DFHISTAR job and about the parameters of the DFHISTAR job, see [“Editing the DFHISTAR job” on page 22](#). Do not change the other parameters in the DFHISTAR job.

2. Submit the DFHISTAR job.

When you run the DFHISTAR job, it saves the generated version of the DFHINSTA job in the CICSTS56.XDFHINST library with the member name specified on the SELECT parameter in the DFHISTAR job. The LIB parameter of the DFHISTAR job specifies the data set name of the CICSTS56.XDFHINST library.

3. Consider running the DFHIHFSA job.

If you decide to create an additional SMP/E target zone using job DFHINSTA, run the DFHIHFSA job before running the DFHINSTA job. DFHIHFSA creates an additional SMP/E target zone for z/OS UNIX.

This job performs the following steps:

- Unmounts the z/OS UNIX file system at directory */pathprefix/usr/lpp/cicsts/ussdir* to allow the job to rerun and, if necessary, forces return code 0.
- Deletes the */ussdira* directory at */pathprefix/usr/lpp/cicsts*, where *ussdira* is the name of the directory specified on the *ussdira* parameter in the DFHISTAR job. This allows the job to rerun and, if necessary, forces return code 0.
- Deletes the file system specified in the HFSADSN parameter of the DFHISTAR job to allow the job to rerun and, if necessary, forces return code 0.
- Creates the file system specified in the HFSADSN parameter of the DFHISTAR job.
- Creates the */ussdira* directory at */pathprefix/usr/lpp/cicsts*, where *ussdira* is the name of the directory specified in the *ussdira* parameter in the DFHISTAR job.
- Mounts the file system at directory */pathprefix/usr/lpp/cicsts/ussdira*
- Creates the *dfhconfig* directory at */pathprefix/usr/lpp/cicsts/ussdira*.
- Creates empty file *featuretoggle.properties* in the *dfhconfig* directory.
- Changes the permission settings for the *ussdira* directory and the *ussdira/dfhconfig* directory to 775.

All steps of this job must end with return code zero for the job to be successful.

CICS requires the MOUNT issued by DFHIHFSA to access files stored in the z/OS UNIX, but the MOUNT command is lost when you re-IPL MVS. SDFHINST member DFHBXPA contains a MOUNT command for *@pathprefix@/uss_path_a@* where *uss_path_a* is the name of the directory specified in the *uss_path* parameter in the DFHISTAR job. Copy this command into a BPXPRMxx member of the SYS1.PARMLIB data set to ensure that the mount is restored when MVS is IPLed.

4. Submit DFHINSTA.

The DFHINSTA job, or a copy of it, copies the CICS Transaction Server target libraries specified by the INDEX parameter, and creates corresponding CICS Transaction Server SMP/E data sets for them. In particular, it allocates a new SMP/E CSI data set for the extra target zone.

So that you can run the DFHINSTA job more than once, step 1 deletes previous copies of the data sets to be created. Step 3 deletes the SMP/E CSI data set. Step 6 removes the ZONEINDEX entry for the extra target zone.

The first time the DFHINSTA job is run it will fail with return code 8, and step 6 will give the following messages:

```
GIM35701E ** ZINDEX SUBENTRY azone WAS NOT DELETED BECAUSE
              IT DOES NOT EXIST.
GIM25601I    THE SPECIFIED ENTRY WAS NOT UPDATED BECAUSE OF
              AN ERROR DURING UCLIN PROCESSING.
```

You can ignore these messages the first time the job is run.

Loading other optional features

The sample job, DFHINSTJ, is supplied for loading other optional features.

CICS Transaction Server contains no optional features at General Availability time, but you can use the sample job if any optional features are shipped at a later date. The DFHINSTJ job is generated in the CICSTS56.XDFHINST library when you run the DFHISTAR job.

Chapter 6. Activating CICS Transaction Server

Perform the following steps after you have loaded CICS Transaction Server to disk.

1. Apply service; see [“Applying service to CICS Transaction Server for z/OS”](#) on page 156.
2. Integrate CICS Transaction Server with MVS; see [Chapter 7, “Setting up the MVS environment for CICS,”](#) on page 61.
3. Create CICS Transaction Server system data sets; see [“Jobs for creating the CICS data sets”](#) on page 170.
4. (Optional) Install CICS IMS DBCTL support; see [“Defining DL/I support”](#) on page 192.
5. (Optional) Install MRO and ISC support, see [“Installing MRO support”](#) on page 182.
6. (Optional) Run the installation verification procedures (IVPs); see [Chapter 10, “Verifying the CICS installation,”](#) on page 197.

Information about putting CICS Transaction Server into operational status is included as part of the process of verifying the installation; see [Chapter 10, “Verifying the CICS installation,”](#) on page 197.

Chapter 7. Setting up the MVS environment for CICS

You must perform a number of tasks after you have loaded the elements to DASD and before you run CICS.

The information about ACF/Communications Server, MVS, RACF, and other products given is for guidance only. Always consult the current publications of these products for the latest information.

The term "RACF" is used throughout this information to mean the MVS Resource Access Control Facility (RACF) or any other external security manager that provides equivalent function.

Authorizing the CICS and CICSplex SM libraries

The SDFHAUTH, SEYUAUTH, SDFHLINK, SEYULINK libraries and the libraries for your product license modules (SDFHLIC, SDFHVUE for Value Unit Edition, and SDFHDEV for Developer Trial) must be APF-authorized.

The SDFHLPA and SEYULPA libraries do not need to be APF-authorized, because any module in the link pack area (pageable LPA, modified LPA, fixed LPA, or dynamic LPA) is treated by the system as though it came from an APF-authorized library. However, you must ensure that you properly protect any data set in the LPALST to avoid system security and integrity exposures, just as you would protect any APF-authorized library.

1. APF-authorize these libraries by adding them to the list of APF-authorized libraries in the appropriate PROGxx or IEAAPFxx member in SYS1.PARMLIB. These libraries must be APF-authorized to enable certain modules, such as DFHSIP, to run in supervisor state.
2. If your lists of APF-authorized libraries are specified in the dynamic format in a PROGxx member, refresh the APF list dynamically by using the SETPROG or SET PROG=xx command.
3. If your lists of APF-authorized libraries are specified in the static format in IEAAPFxx members, schedule an MVS IPL for the APF-authorization to take effect.
4. When you prepare your startup job stream, provide a STEPLIB DD statement for the SDFHAUTH and SEYUAUTH libraries. The SEYUAUTH library loads the message files, therefore it is required for both basic CICS functions and if you are using any CICSplex SM or CICS Explorer functions. When you define your STEPLIB DD statement, remember that all other libraries concatenated with the SDFHAUTH and SEYUAUTH libraries must also be APF-authorized. If any of the libraries in a STEPLIB concatenation are not authorized, MVS regards all of them as unauthorized.
5. The SDFHLOAD and SEYULOAD libraries contain only programs that run in problem state, and **must not** be authorized. You must include the SDFHLOAD library in the CICS DFHRPL library concatenation. An example of this library DD statement is in the sample job stream in [A sample CICS startup job](#). For offline utilities the SDFHLOAD and SEYULOAD libraries are included in the STEPLIB concatenation of the job but again they **must not** be authorized. For example, the CICS translator will abend with abend code U0101 if SDFHLOAD is authorized.

Although, in general, CICS runs in problem state, the CICS initialization program, DFHSIP, must run in supervisor state for part of its execution. The CMAS startup program, EYU9XECs, also requires APF authorization.

For a module to run in supervisor state, it must be link-edited as an authorized module into a partitioned data set, which must also be defined to the operating system as APF-authorized. For CICS-supplied modules, the link-editing has been done for you. The CICS-supplied DFHSIP module is link-edited with the authorized attribute, using SETCODE AC(1), and is installed in the SDFHAUTH library.

For information about maintaining lists of APF-authorized libraries, see the [z/OS MVS Initialization and Tuning Guide](#).

For information about authorizing access to CICS data sets, see [Authorizing access to CICS data sets](#).

Authorizing CICS regions to access MVS resources

Protect your data sets, authorize your user IDs and regions and activate your RACF resource classes when you set up your security requirements to run CICS.

Grant access to the PDS libraries

Protect your data sets that use RACF. See [“Protecting CICS load module data sets”](#) on page 62.

SNA ACB

Authorize each CICS region user ID to open the SNA ACB for the region applid.

CICS system transactions

Authorize each CICS region user ID to access the CICS category 1 system transactions.

SMSVSAM server

Authorize each CICS region to open the SMSVSAM control ACB if you plan to use CICS with VSAM record-level data sharing.

System logger log streams

Authorize each CICS region user ID to access the MVS system logger log streams that are used by CICS.

z/OS UNIX

Include an OMVS segment in the CICS region user profile, specifying the UID parameter, to ensure that your CICS regions have the required access to z/OS UNIX. Failure to do so results in CICS failing to start under that region user ID with RACF message ICH408I and CICS message DFHKE002. Other messages such as DFHKE0501 and DFHDM0105 might also occur if different CICS domains fail to initialize.

RACF resource classes

Activate the appropriate RACF resource classes to enable terminal users to access CICS resources and user-defined resources.

VSAM catalogs

Give the CICS region user ID read access to each VSAM catalog that contains files, where CICS has the file definitions installed, and where these files are opened during or after CICS startup. Include the VSAM catalog for the DFHCSD file for the CICS system definition data set (CSD).

Protecting CICS load module data sets

To prevent unauthorized or accidental modification of *hlq.SDFHAUTH*, you must protect it with RACF.

Without protection, the integrity and security of your MVS system are at risk. Additionally, if you require protection against the unauthorized use of DFHSIP, do not place this module in the LPA and do not include *hlq.SDFHAUTH* in the MVS LNKST unless DFHSIP is RACF-protected as a controlled program with a profile in the RACF PROGRAM resource class.

You must also RACF-protect the other libraries, including *hlq.SDFHLOAD*, that make up the STEPLIB and DFHRPL library concatenations.

For information about authorizing access to CICS data sets, see [Implementing RACF protection in a single CICS region](#).

Authorizing access to data set services modules

During initialization, CICS determines the availability of backup-while-open (BWO) support by linking to the callable services modules IGWAMCS2 and IGWABWO.

CICS also checks the DFSMSdss or DFDSS release level by linking to the module ADRRELVL. If access to this data set services module is controlled by means of RACF PROGRAM general resource profiles, security violation messages are issued against the CICS region user ID, unless the user ID is authorized to access ADR-prefixed module names.

You can avoid security violation messages against the CICS region user IDs, and still control access to data set services:

- If you have generic PROGRAM profiles protecting access to ADR modules, create specific PROGRAM profiles for the ADDRELVL module and ensure that your CICS region user IDs have READ access to these specific profiles.
- Instead of using PROGRAM profiles to protect access to data set services, use one of the following methods:
 - Define suitable profiles in the DASDVOL general resource class.
 - Define profiles in the FACILITY general resource class that are supported by DFSMS to control access to data set services.

For information about using DASDVOL and FACILITY class profiles to control the uses of data set services, see [z/OS DFSMSdss Storage Administration Guide](#).

Authorizing access to a CICS region z/OS Communications Server ACB

You can control which users, among those who are running non-APF-authorized programs, can open the SNA ACB associated with a CICS address space (CICS region).

Control ensures that only authorized CICS regions can present themselves as z/OS Communications Server applications that provide services with this applid, thus preventing unauthorized users from impersonating real CICS regions. The CICS region user ID requires the OPEN access, not the issuer of the z/OS Communications Server SET VTAM® OPEN command.

1. To enable CICS to start with external security, authorize the CICS region user ID to open the CICS region's z/OS Communications Server ACB with the applid specified on the APPLID system initialization parameter.
2. For each applid, create an z/OS Communications Server APPL profile, and give the CICS region user ID READ access. For example:

```
RDEFINE VTAMAPPL applid UACC(NONE) NOTIFY(userid)
PERMIT applid CLASS(VTAMAPPL) ID(cics_region_userid) ACCESS(READ)
```

The correct CICS applid to specify in the VTAMAPPL class is the specific applid, as specified in the CICS system initialization parameters. If you are using XRF (that is, if CICS is started with XRF=YES), you must define two VTAMAPPL profiles — one each for both the active and alternate CICS region's specific applid (the second operand on the CICS APPLID startup option).

3. Ensure that the VTAMAPPL class is active and RACLISTed for this protection to be in effect; for example:

```
SETOPTS CLASSACT(VTAMAPPL) RACLIST(VTAMAPPL)
```

4. If a CICS region will not use z/OS Communications Server, do not reauthorize the CICS region user ID for the CICS applid.
5. If you do not control the opening of a CICS region's z/OS Communications Server ACB, a new z/OS Communications Server application started with the same applid as that of a running CICS region has the following effect:
 - The running CICS region performs a FORCECLOSE of its SNA ACB and issues message DFHZC0101.
 - The running CICS region either ends or continues, depending on your use of the XXRSTAT exit. The default is to end. If the CICS region continues, it no longer uses z/OS Communications Server.
 - The new application opens the z/OS Communications Server ACB with the specified applid.
 - If the first running CICS region used z/OS Communications Server persistent sessions, the new application recovers any z/OS Communications Server sessions that persist from that CICS region.

For information about creating z/OS Communications Server APPL profiles for CICS region applids, see the [CICS system resource security](#). For information about the XXRSTAT exit, see the [Exit XXRSTAT](#).

Authorizing the region user ID to access category 1 transactions

To enable CICS to start using external security, you must first have authorized the CICS region user ID to access the category 1 system transactions. If the region user ID does not have this authority at CICS startup, CICS issues message DFHXS1113 for each unauthorized category 1 system transaction and ends with message DFHXS1404.

To give the region user ID the authority to access the category 1 system transactions, edit and submit the sample job stream in [Figure 3 on page 64](#) to run the CICS-supplied sample CLIST, DFH\$CAT1. This job uses the RACF commands in the CLIST to update the RACF database.

Only a user with the appropriate RACF authority can run the CLIST to update the RACF database.

```
//RACFMIG JOB 'accounting information',  
//          CLASS=A,USER=userid,PASSWORD=password  
//DEFINE EXEC PGM=IKJEFT01  
//SYSPRINT DD SYSOUT=A  
//SYSTSPRT DD SYSOUT=A  
//SYSUDUMP DD SYSOUT=A  
//SYSTSIN DD *  
EXEC 'CICSTS56.CICS.SDFHSAMP(DFH$CAT1)' LIST  
/*  
//
```

Figure 3. Batch job to run the sample CLIST, DFH\$CAT1

Authorizing access to an SMSVSAM server

If you plan to run CICS with VSAM record-level sharing (RLS), you must authorize each region that connects to an SMSVSAM server to have access to that server.

In a test environment you might want to use the default action and allow any CICS region using VSAM RLS to connect to an SMSVSAM server. If you want to protect this access, the RACF SUBSYSNM general resource class must be active and you must authorize each CICS region that connects to an SMSVSAM server to have access to that server. This means granting access to the appropriate profile in the RACF SUBSYSNM general resource class. You define profiles in the SUBSYSNM resource class to control access by subsystems like CICS that want to connect to SMSVSAM.

A SUBSYSNM profile name is the name by which a given subsystem, such as CICS, is known to VSAM. For CICS regions, you must use the CICS applid as the profile name in the SUBSYSNM general resource class.

When CICS attempts to register the control ACB during CICS initialization, SMSVSAM calls RACF to check that the CICS region user ID is authorized to a profile name in the SUBSYSNM class that matches the CICS applid. If the CICS region user ID does not have READ authority, the register fails.

For example, if the applid of a CICS AOR is *CICSDAA1*, and the CICS region user ID (shared by a number of AORs) is *CICSDA##*, define and authorize the profile:

```
RDEFINE SUBSYSNM CICSDAA1 UACC(NONE) NOTIFY(userid)  
PERMIT CICSDAA1 CLASS(SUBSYSNM) ID(CICSDA##) ACCESS(READ)
```

You can use wildcard characters on the applid to specify more than one CICS region, for example:

```
PERMIT CICSD%% CLASS(SUBSYSNM) ID(CICSDGRP) ACCESS(READ)
```

Authorizing access to MVS log streams

CICS does not control LOGSTRM security checks. To control them, the MVS security administrator activates the LOGSTRM and FACILITY general resource classes with the SETROPTS command.

Users of the IXCMIAPU administrative data utility and CICS regions both require appropriate authorizations to log streams and IXLSTR coupling facility structures.

Authorizing users of IXCMIAPU

You can create log structures and define log streams using the IXCMIAPU administrative data utility to update the LOGR data set. Your user ID must have the appropriate level of authorization.

Here are examples of levels of user ID authorization:

Coupling facility structures

To define and delete log structures using IXCMIAPU, you must have ALTER access to the LOGR resource profile named MVSADMIN.LOGR in the FACILITY general resource class. For example, use the following RACF command:

```
PERMIT MVSADMIN.LOGR CLASS(FACILITY) ACCESS(ALTER) ID(your_userid)
```

Coupling facility log streams

To define, delete, and update log streams, including log stream models, that are defined in coupling facility structures, you require these accesses:

- ALTER access to the appropriate log stream profile defined in the LOGSTRM general resource class
- UPDATE access to the coupling facility structure (IXLSTR) profile defined in the FACILITY general resource class; in this case, profile names are prefixed with IXLSTR.

For example, if the log stream and structure resource profiles are defined to RACF with the following commands:

```
RDEFINE LOGSTRM log_stream_profile UACC(NONE) [NOTIFY]  
RDEFINE FACILITY IXLSTR.structure_name_a UACC(NONE) [NOTIFY]
```

Use the following RACF commands to give your user ID the required authorizations to these two profiles:

```
PERMIT log_stream_profile CLASS(LOGSTRM) ACCESS(ALTER) ID(your_userid)  
PERMIT IXLSTR.structure_name_a CLASS(FACILITY) ACCESS(UPDATE) ID(your_userid)
```

Authorizations for CICS regions

If the LOGSTRM resource class is active, the level of authorization required depends on whether log streams are always explicitly defined to the MVS system logger.

Ensure that the CICS region user ID is authorized to write to, and create if necessary, the log streams and log stream data sets that are used for its system log and general logs. See [Defining the logger environment for CICS](#). You do this by granting the appropriate access authorization to log stream profiles in the RACF LOGSTRM general resource class:

- If CICS is expected to create log streams dynamically, CICS must have **ALTER** authority to the relevant log stream (LOGSTRM) profiles, and **UPDATE** authority to the relevant coupling facility structure (IXLSTR and IXGLOGR) profiles. Here is an example:

```
PERMIT region_userid.applid.* CLASS(LOGSTRM) ACCESS(ALTER)  
ID(region_userid)  
PERMIT IXLSTR.structurename CLASS(FACILITY) ACCESS(UPDATE)  
ID(region_userid)
```

```
PERMIT IXGLOGR.region_userid.* CLASS(DATASET) ACCESS(UPDATE)  
ID(region_userid)
```

- If all the log streams to which CICS writes are already defined to MVS, CICS requires only **UPDATE** authority to the log stream profiles:

```
PERMIT region_userid.applid* CLASS(LOGSTRM) ACCESS(UPDATE)  
ID(region_userid)
```

```
PERMIT IXGLOGR.region_userid.* CLASS(DATASET) ACCESS(UPDATE)  
ID(region_userid)
```

In the above examples, *region_userid.applid.** is the generic profile name of the log stream resource. These examples illustrate a resource name prefixed by the region user ID and applid. *region_userid* is the CICS region user ID under which CICS is running, either as a started task or batch job.

Permit READ access to those users who read the CICS log streams. You must permit UPDATE access to those users who update journals by granting the user the appropriate authority to the log stream, in the LOGSTRM resource class, and to the JOURNALNAME in the JCICSJCT class. You must also grant access to the data set profile protecting the data set that contains the log stream.

The user ID for the JCICSJCT is the logon user ID, not the CICS region user ID. The example shows how to define it:

```
PERMIT journal_name CLASS(JCICSJCT) ACCESS(UPDATE) ID(logon_userid)
```

You can define the generic profile in the following example to cover all the log streams referenced by the CICS region identified by its region user ID and applid:

```
RDEFINE LOGSTRM region_userid.** UACC(NONE)
```

If, however, you have multiple CICS systems sharing the same region user ID, but with differing security requirements, include the applid in the generic profile:

```
RDEFINE LOGSTRM region_userid.applid.* UACC(NONE)
```

The following example allows the CICS region user ID under which CICS is running to write journal and log records to log streams in the named coupling facility structure:

```
PERMIT IXLSTR.structurename CLASS(FACILITY) ACCESS(UPDATE)  
ID(region_userid)
```

The following examples give access to two categories of user:

```
PERMIT region_userid.applid.* CLASS(LOGSTRM) ACCESS(READ)  
ID(authorized_browsers)  
PERMIT region_userid.applid* CLASS(LOGSTRM) ACCESS(UPDATE)  
ID(archive_userid)
```

In these examples, *archive_userid* is the user ID under which an application program runs to purge old data from CICS logs when the data is no longer required and *authorized_browsers* refers to the user IDs of users allowed to read log streams but not to purge data.

If several CICS regions share the same CICS region user ID, you can make profiles more generic by specifying * for the *applid* qualifier.

The number of profiles you define depends on the naming conventions of the logs and to what extent you can use generic profiling.

Authorizing access to z/OS UNIX System Services

CICS requires access to z/OS UNIX System Services and to directories and files in the file system. CICS facilities that require this access include TCP/IP support, Java support, CICS Web support, and task-related user exits that are enabled with the OPENAPI option, including the task-related user exit of the CICS Db2 attachment facility.

Before you begin

You must set up z/OS UNIX System Services, as described in [z/OS UNIX System Services Planning](#).

About this task

To satisfy a CICS region request for a z/OS UNIX function for the first time, RACF does the following actions:

- Verifies that the user (the CICS region user ID) is defined as a z/OS UNIX user.

- Verifies that the user's current connect group is defined as a z/OS UNIX group.
- Initializes the control blocks required for subsequent security checks.

Follow the steps listed to ensure that each CICS region meets the security requirements:

1. Choose a RACF group that all your CICS regions can use to access z/OS UNIX and give a z/OS UNIX group identifier (GID) to this RACF group.
2. Give a z/OS UNIX user identifier (UID) to each CICS region user ID.
3. Make sure that each CICS region user ID connects to the RACF group that you chose.
4. Set up a home directory on z/OS UNIX for each of your CICS regions.

The UID and GID are numbers that can be in the range 0 to 16 777 216. 0 is a superuser ID. Give some thought to naming conventions and to any existing UIDs and GIDs in your z/OS UNIX system. For information on how to manage the UIDs and GIDs for your z/OS UNIX system, see [z/OS UNIX System Services Planning](#).

Note: If you are upgrading to z/OS 2.1.0 or above and have defined a BPX.DEFAULT.USER profile, please note that this is no longer supported. Instead you must define the BPX.UNIQUE.USER profile to enable the automatic assigning of UIDs and GIDs to users without an assigned OMVS segment. For more information, see [Using the BPX.UNIQUE.USER profile instead of BPX.DEFAULT.USER in z/OS Migration](#).

To assign a z/OS UNIX UID and GID for your CICS regions and set up a home directory:

Procedure

1. Choose a RACF group that can be used by all your CICS regions. For example, you might use a RACF group that is defined as the default group of your CICS region user IDs, or you could set up a RACF group to be used only for access to z/OS UNIX System Services.
When you are setting up facilities such as Java support or CICS Web support, you might want to use this RACF group for giving file access permissions on z/OS UNIX, in which case the RACF group's z/OS UNIX group identifier (GID) is associated with these directories and files. This association means that the owner of these directories and files, and anyone who is not the owner but has to carry out operations with these files, must have this group as their group or one of their supplementary groups. The [RACF profiles](#) explains how RACF groups work.
2. Choose a suitable z/OS UNIX group identifier (GID) for the RACF group, and assign the GID to the RACF group. To assign a GID, specify the GID value in the OMVS segment of the RACF group profile. For example, if the RACF group is CICSTSAB, and the GID you want to assign is 9, use the command:

```
ALTGROUP CICSTSAB OMVS(GID(9))
```

3. Choose a suitable z/OS UNIX user identifier (UID) for each CICS region. [Implementing security for z/OS UNIX files](#) explains how the region user ID under which CICS runs is specified when CICS is run as a started task, as a started job, or as a job.
 - a) Assign your chosen UIDs to each of your CICS region user IDs. To assign UIDs, specify the UID value in the OMVS segment of the RACF user profile for each CICS region user ID.
 - b) Also specify the name of a home directory for each CICS region using the HOME option. The directory name format is */u/CICS region user ID*.

CICS TS security tells you how to update a RACF user profile using the **ALTUSER** command.

For example, if the CICS region user ID is CICS HAA1, and the UID you want to assign is 2001, use the command:

```
ALTUSER CICS HAA1 OMVS(UID(2001) HOME('/u/cicshaa1'))
```

If you want to know about the other information that can be specified in an OMVS segment parameter in a user profile besides the UID and home directory, see [z/OS Security Server RACF Command Language Reference](#). You can assign the same UID to more than one CICS region user ID. If all your CICS regions must use the same z/OS UNIX files (for example, the files required for Java support),

you can give all the CICS regions the same UID, and then you can use that UID for giving file access permissions on z/OS UNIX. However, bear in mind these points:

- a. The sharing of UIDs allows each CICS region to access all of the z/OS UNIX resources that the other CICS regions with that shared UID can access, and this level of access might not be appropriate in your system.
 - b. The sharing of UIDs is not normally suitable in a z/OS UNIX system.
 - c. If you do choose to share UIDs, the z/OS UNIX System Services parameter **MAXPROCUSER** limits the maximum number of processes that a single user (that is, with the same UID) can have concurrently active.
4. Set up each of the z/OS UNIX directories that you have specified as a home directory for one of your CICS regions.

Further information on each of these steps is provided in [z/OS UNIX System Services Planning](#).

- a) If you are not using an automount facility, use the **mkdir** command to create the z/OS UNIX directories.

For example, issuing the UNIX command

```
mkdir /u/cicshaa1
```

Creates the z/OS UNIX directory `/u/cicshaa1`. If you are using the TSO command, enclose the directory name in single quotation marks.

- b) Whether or not you are using an automount facility, allocate a z/OS UNIX data set for each directory.

- c) If you are not using an automount facility, mount the data set that you have allocated.

The z/OS UNIX data set that you allocate for a CICS region's home directory has a finite size. If a particular CICS region is using the home directory extensively, you might have to increase the amount of space that the region has available.

5. Verify that the permission defined for the `/etc` directory is set to 755, so that CICS can access the files. The `/etc` directory provides a symbolic link to the `/SYSTEM/etc` directory. The `/SYSTEM/etc` directory is created with a permission of 700, so you need to check that the `/SYSTEM/etc` directory permission is set to 755.

- a) Check the permission set in the `/SYSTEM/etc` directory, from the Unix shell:

```
ls -ld /SYSTEM/etc
```

- b) If the permission is not `dwxr-xr-x`, issue the following Unix shell command to set permission to 755:

```
chmod 755 /SYSTEM/etc
```

If permission is not set to 755, you might receive a RACF error indicating that you have insufficient authority for CICS to access the files.

6. Make sure that each of your CICS region user IDs connects to the RACF group to which you assigned a z/OS UNIX group identifier (GID). If your CICS region user IDs connect to more than one RACF group, a RACF list of groups must be active in your system.

What to do next

Your CICS regions now have access to z/OS UNIX System Services. When you set up facilities such as Java support or CICS Web support, use the UIDs or GIDs to give the CICS regions permission to access directories and files on z/OS UNIX.

To check the UID and GID details for a user, use the **id** command in the UNIX environment. For example, issuing the **id** command for our example CICS region user ID CICS_HAA1 gives the following result:

```
uid=2001(CICS_HAA1) gid=9(CICSTSAB)
```

For more general information about RACF facilities for controlling access to z/OS UNIX System Services, see [z/OS Security Server RACF Security Administrator's Guide](#).

Activating RACF resource classes

Before you can use RACF for CICS resources and for user-defined resources, you must activate the associated RACF resource classes by using the RACF SETROPTS command.

To run the CICS-supplied IVPs with external security, you must activate the resource classes for CICS resources.

To use your own user-defined resources with external security in your CICS environment:

- Define resource classes for your resources.
- Activate the resource classes.
- Optionally apply the RACLIST option to the resource classes to be used for QUERY SECURITY commands, to build in storage profiles for those resource classes.

For information about RACF resource classes, see the [RACF classes for CICS resources](#).

Defining the default CICS user ID to RACF

If you intend to use RACF to authorize terminal users to access CICS resources, define a default CICS user ID to RACF and specify it on the CICS system initialization parameter, DFLTUSER.

This default user ID assigns the security attributes to be used for all CICS terminal users who do not sign on with the CESN transaction or a user-written equivalent.

During startup, CICS tries to sign on the default user ID. If it is not signed on, perhaps because it is not defined, CICS issues message DFHSN0701 and stops CICS initialization. After the valid default CICS user ID is signed on, its security attributes are used for all CICS terminal users who do not sign on with the CESN transaction. If the default user ID is defined to RACF with a CICS segment, the operator attributes in that segment are also used for users who do not sign on.

For information about defining the user ID to RACF, see the [CICS installation requirements for RACF](#).

Installing CICS-required modules in the MVS linklist

There are two categories of modules that are loaded from the MVS linklist. These are CICS- and CICSplex SM-supplied modules, and modules of other MVS products, for example, DFSMS.

Look through the subtopics in this section and select the ones that are relevant to your installation. Work through these topics to create your customized MVS linklist.

CICS- and CICSplex SM-supplied modules required in the MVS linklist

CICS TS supplies the modules listed in the SDFHLINK and SEYULINK libraries when you install CICS TS.

The following CICS modules are in SDFHLINK:

AMDUSREF

Alias of DFHTG730.

AXMSC

AXM server connection routines for CICS data-sharing servers.

AXMSI

AXM subsystem initialization routine for CICS data-sharing servers.

DFHDTVC

Connection validation subroutine for shared data tables.

DFHDTVC

Shared data tables SVC services.

DFHGTCNV

Subroutine used by LOGR subsystem interface.

DFHHCADD

CICS Health Checker CSZADDCHECK exit routine.

DFHHCCHK

CICS Health Checker.

DFHHZMGT

CICS Health Checker messages.

DFHLGCNV

Exit routine for LOGR subsystem interface.

DFHMRMS

General MVS RESMGR exit stub.

DFHNCIF

Named counter server interface.

DFHNCOPT

Named counter server options.

DFHPD730

Dump formatting routine for use with IPCS.

DFHRPDUF

System dump formatting routine for ONC RPC.

DFHRPTRI

Trace interpretation routine for ONC RPC.

DFHRXSVC

RRS domain authorized services.

DFHSNNFY

RACF CICS segment changes notification routine.

DFHSNPTO

CICS RACF dynamic parse TIMEOUT keyword print routine.

DFHSNVCL

CICS RACF dynamic parse OPCLASS validation routine.

DFHSNVID

CICS RACF dynamic parse OPIDENT validation routine.

DFHSNVPR

CICS RACF dynamic parse OPPTY validation routine.

DFHSNVTO

CICS RACF dynamic parse TIMEOUT validation routine.

DFHSSIN

CICS subsystem that initializes the console message handling facilities.

DFHSSMGT

CICS subsystem message table that contains the text of messages for the subsystem interface modules.

DFHTG730

Link module for the CICS GTF trace printing load module DFHTRGTF.

DFHTR730

Link module for the CICS GTF trace printing load module DFHTR730.

DFHTT730

Link module used for trace interpretation. You do not have to include DFHTT730 in the MVS linklist. If it is not in the linklist, an APF authorized library in the STEPLIB concatenation of the batch job is required for every job that uses the module.

DFHXCSVC

External CICS interface (EXCI) SVC services routine.

The following CICSplex SM modules are in SEYULINK:

EYU9D560

Dump formatting routine for use with IPCS.

EYU9X560

In each MVS image that contains a CMAS. EYU9X560 is the initialization module for the ESSS.

EYU9A560

In each MVS image that contains a CMAS where you want to run the CICSplex SM API. EYU9A560 is the CICSplex SM API subtask module.

EYU9T560

CICSplex SM to Tivoli® NetView Communications module.

These modules are supplied in an APF-authorized library in the MVS linklist because of the following reasons:

1. Non-CICS regions such as batch jobs or a CICS data sharing server can require them.
2. They must be consistent across several CICS regions.
3. Both CICS and non-CICS regions can require them.
4. The Security Administrator who runs the ADDUSER or ALTUSER commands under TSO requires the RACF dynamic parse routines. For information about the RACF interface routines, see the [An overview of the CICSplex SM ESM interface](#).

Ensure the modules supplied in SDFHLINK and SEYULINK are available from an APF-authorized library in the MVS linklist:

- Add these modules, as required, to an existing APF-authorized library defined in the MVS linklist
or
- Define SDFHLINK and SEYULINK as APF-authorized libraries and include them in the MVS linklist

Compatibility with earlier CICS releases

Unless otherwise stated, the CICS Transaction Server for z/OS, Version 5 Release 6 levels of the modules in SDFHLINK are compatible with earlier releases of CICS. The CICSplex SM modules in SEYULINK are not compatible with earlier releases. CICSplex SM modules in SEYULINK are release specific. If you plan to run multiple releases of CICSplex SM on the same MVS image, you must have the equivalent modules specific to the releases you are running.

DFHPD730, DFHTG730, DFHTR730, and DFHTT730 are release-dependent. If you run more than one release of CICS, ensure that the correct versions are available.

CICS shared data tables modules for the MVS linklist

CICS supplies the following modules, for shared data tables, in *hlq*.SDFHLINK.

If you intend using shared data tables, ensure that these modules are available in the MVS linklist or the MVS link pack area:

- DFHDSVC and DFHDTVC, because all regions using shared data tables must use the same level of SVC code.
- DFHMVRMS, the RESMGR exit stub, because JOBLIB and STEPLIB data sets are unavailable at end-of-memory.

CICSplex SM API linklist modules

In each MVS image that contains a CMAS in which you plan to run the CICSplex SM API, ensure that these required modules are in the correct location.

These modules are supplied in the SEYUAUTH library.

EYU9AB00

In either the MVS linklist or the STEPLIB concatenation of the application that calls the API.
EYU9AB00 is the API batch interface module.

EYU9XESV

In an authorized library in either the MVS linklist or the CMAS STEPLIB concatenation. EYU9XESV is the API security exit module.

In addition, any application that calls the API must be link-edited with one of the following stub routine modules, regardless of what programming language is used:

EYU9ABSI

For batch, TSO, or NetView programs. EYU9ABSI is supplied in the SEYUAUTH library.

EYU9AMSI

For application programs running in CICS. EYU9AMSI is supplied in the SEYULOAD library.

Installing the REXX function package

The REXX runtime interface to the API consists of a function package and host command environment.

The interface consists of a single load module that contains two entry points:

EYU9AR00

The function package

EYU9AR01

The host command

EYU9AR00 is supplied in the SEYUAUTH library with an alias of IRXFLOC.

For a REXX program to access the function package, the module EYU9AR00, with its alternate entry point EYU9AR01, and its alias IRXFLOC, must be in an authorized library in one of these places:

- The MVS linklist
- The STEPLIB concatenation of the application that calls the API

For a REXX program to access the function package from NetView, the EYU9AR00 module must also be aliased to DSIRXLFP and placed in an authorized library in either the MVS linklist or the STEPLIB concatenation for the NetView system.

Users of the CICSplex SM runtime interface are subject to the normal CICSplex SM API security checks. See the information in the [CICSplex SM security](#).

The following members contain SMP/E user modification control statements that you can use to move the necessary API load modules to the SEYULINK library. These members are supplied in CICSTS56 SEYUSAMP.

Member	Load module
EYU\$UM11	EYU9AR00
EYU\$UM12	EYU9AB00
EYU\$UM13	EYU9XESV

If you use the IRXFLOC or DSIRXLFP aliases to provide access to the REXX function package, you must place them ahead of any other IRXFLOC or DSIRXLFP modules in the STEPLIB or MVS linklist concatenation.

If you do not want to use the aliases for the REXX function package, you must modify your REXX parameter modules IRXPARMs, IRXTSPRM, and IRXISPRM. If you do this, do the following actions:

- Add the function package supplied by CICSplex SM as a System function package, rather than a Local or User function package.
- Add a new host command entry. Here are some examples:
 - An 8-byte Command Environment name of 'CPSM '
 - An 8-byte Command Routine name of 'EYU9AR01'
 - A 16-byte Command Token of blanks

To complete installing the REXX function package:

- Increase the number of entries in the appropriate function package table.
- Add an entry to that table for EYU9AR00.

For more information about REXX function packages and host commands, see [Developing REXX applications](#).

Modules of other MVS products in the MVS linklist

CICS loads some DFSMS modules from the MVS linklist. This requirement either depends on the function you are using, such as backup-while-open (BWO) support, or on the release of DFSMS.

The following modules are loaded:

IGWABWO

CICS loads this module, supplied in the MVS callable services library SYS1.CSSLIB, from the MVS linklist if you are using BWO for files accessed in non-RLS mode. In addition to IGWABWO in the linklist, IGWAMCS2 must be installed in the LPA. CICS tests for the presence of this module in the LPA to determine that BWO support is present in the MVS image before attempting to load IGWABWO.

For files that are accessed in RLS mode, CICS does not require IGWABWO or IGWAMCS2.

IGWARLS

CICS loads this module, supplied in the MVS callable services library SYS1.CSSLIB, from the MVS linklist. CICS issues the following message if it cannot load IGWARLS:

```
DFHFC0116  APPLID THE LOAD OF CALLABLE
           SERVICE IGWARLS HAS FAILED WITH RETURN CODE
           X'EEEE' .
```

CICS initialization fails if CICS cannot load this callable services module.

Defining CICS as an MVS subsystem

You define CICS as an MVS subsystem before you can use any of facilities listed.

- The console message-handling facility
- Multiregion operation (MRO)
- CICS shared data tables
- External CICS interface (EXCI).

For information about the console message-handling facility, see [Administering CICS operations](#).

For information about MRO, see [Multiregion operation](#).

The definition of CICS as an MVS subsystem involves three members of the SYS1.PARMLIB partitioned data set: IEASYSxx, IEFSSNaa, and DFHSSIyy. You require only member DFHSSIyy if you want the console message-handling facility.

In an IEASYSxx member of the SYS1.PARMLIB library used for MVS initialization, include the parameter SSN=aa, where aa refers to the SYS1.PARMLIB member IEFSSNaa that contains the definitions for all subsystems required for this IPL of MVS, including the definition of CICS as an MVS subsystem.

aa,xx,yy represent suffixes that are used to distinguish different versions of members of the SYS1.PARMLIB library.

To start CICS with the START command:

- Give the MVS started task procedure a name different from the subsystem name in IEFSSNaa (default 'CICS'),
or
- Issue the start command with the parameter SUB=JES2 or SUB=JES3 as appropriate.

For more information about the subsystem interface, see [z/OS MVS Using the Subsystem Interface](#).

The following topics cover:

- [“Noting IEASYSxx values for CICSplex SM” on page 74](#)
- [“Specifying each CMAS correctly in IEASYSxx” on page 75](#)
- [“Coding the IEFSSNaa MVS subsystem initialization member” on page 76](#)
- [“The SYS1.PARMLIB\(BPXPRMxx\) parameters” on page 77](#)
- [“The console message-handling facility” on page 79](#)
- [“EXCI pipe allocation” on page 83](#)

Coexistence with automation programs and other CICS releases

If you are using automated-operation programs or multiple CICS releases, then you must plan for them to coexist with the CICS message-handling facilities.

If your automation system must see the console messages before they are reformatted by CICS, place its subsystem definition in IEFSSNXX before the definition for CICS. But, if your automation system must see the reformatted messages, its definition must come after that of CICS. Consult the documentation of your automation package to determine which applies to you.

If you have defined the message-handling facility has been defined to MVS (by the CICS entry in the IEFSSNaa member of the SYS1.PARMLIB library), CICS regions running earlier releases of CICS in the same MVS image have the full benefit of the message handling that has been defined if either of the following cases is true:

- An automated-operation program, such as NetView, is active in the MVS image.
- A CICS region that supports message handling (see [“The console message-handling facility” on page 79](#) for a list) is running in the same MVS image.

A consequence of the standard format console messages is that they no longer include date, time, and informational messages or information. If you use this kind of information as a token, you must make a change to the code so that it looks for a different token.

Noting IEASYSxx values for CICSplex SM

Some of the MVS initialization values located in an IEASYSxx member of the SYS1.PARMLIB library are referenced during installation of CICSplex SM address spaces.

Access the IEASYSxx member of the SYS1.PARMLIB library that is used to initialize your MVS system and make note of the values assigned to the following parameters:

APF=

Completes the name of the parmlib member (IEAAPFxx) that contains authorized library names.

CMD=

Completes the name of the parmlib member (COMMNDxx) that contains commands to be issued internally during main scheduler initialization.

LNK=

Completes the name of one or more parmlib members (LNKLSTxx) that contain names of data sets that are to be concatenated to SYS1.LINKLIB.

LNKAUTH=

Specifies that all data sets in the LNKLST concatenation will be treated as APF-authorized or that only those that are named in the APF table will be treated as APF-authorized.

LPA=

Completes the name of one or more parmlib members (LPALSTxx) that are concatenated to SYS1.LPALIB for the purpose of building the pageable LPA (PLPA and extended PLPA).

MAXUSER=

Specifies a value that the system uses, along with the RSVSTRT and RSVNONR parameter values, to limit the number of jobs and started tasks that the system can run concurrently during a given IPL.

NSYSLX=

Specifies the number of linkage indexes (LXs), in addition to those in the system function table, that will be reserved for system linkage indexes (LXs).

PROG=

Completes the name of the parmlib member (PROGxx) that contains authorized library names when a dynamic APF list is being used.

RSVNONR=

Specifies the number of address space vector table (ASVT) entries that will be reserved for replacing those entries marked nonreusable for the duration of an IPL.

RSVSTRT=

Specifies the number of ASVT entries that will be reserved for address spaces created in response to a START command.

SYSNAME=

Specifies the name of the system being initialized.

For more information about these parameters, see [z/OS MVS Initialization and Tuning Reference](#).

Specifying each CMAS correctly in IEASYSxx

In every z/OS image that contains a CMAS, verify that the IEASYSxx member of the SYS1.PARMLIB library that you use for z/OS initialization includes the **MAXCAD** and **NSYSLX** parameters.

MAXCAD=nnn

Set the value to include the number of SCOPE=COMMON MVS™ data spaces that are required for each LPAR. This value will impact the number of concurrent CMASes that can run on each LPAR.

Each CMAS requires a number of SCOPE=ALL data spaces. This number depends on the **CACHEDSNUM** value for the CMAS. This setting controls the number of data spaces allocated for each component, except the data cache manager, which has only one data space. Therefore, the number of SCOPE=ALL data spaces required for a CMAS can be calculated as follows:

$$\text{number of SCOPE=ALL data spaces} = (9 * \text{CACHEDSNUM}) + 1$$

The default value of the **CACHEDSNUM** parameter is 2, and normally you do not change it unless advised to do so by IBM Support. Therefore, each CMAS typically requires 19 SCOPE=ALL data spaces.

Include any SCOPE=COMMON data spaces that might be in use by other products in the **MAXCAD** value.

The lower the **MAXCAD** value, the more CMASs can be run at the same time on the LPAR.

At CMAS startup, if MVS cannot allocate the required number of SCOPE=ALL data spaces, CICSplex SM produces message EYUXC0011E with a return code of '0000000C' and the CMAS shuts down. To reduce the chance of this happening, either lower the **MAXCAD** value if possible, or plan to reduce the number of CMASes on the LPAR.

Calculate the maximum number of CMASes that you can have on the LPAR at this CICS release as follows. Be aware that other releases of CICS TS might have different requirements:

1. Subtract the maximum number of entries in the PASN-AL (510) from the current **MAXCAD** value. This provides the number of PASN-AL entries that remain available for use by the Environment Services System Services (ESSS) address space to access the required data spaces. For example:
 - If **MAXCAD** is set to the maximum possible value of 250, and the PASN-AL limit is 510, the calculation is: 510 (PASN-AL limit) - 250 (**MAXCAD** value) = 260 available entries left on the PASN-AL.
2. Divide the remaining PASN-AL entries by the number of SCOPE=ALL data spaces that each CMAS requires. This is the number of CMASes per release that can be started at the same time on the LPAR. For example, taking the number of PASN-AL entries from the previous step:
 - Each CMAS typically uses 19 SCOPE=ALL data spaces, so the calculation is: 260/19 = 13.68. The maximum number of CMASs on the LPAR is 13.

To establish the count of SCOPE=ALL data spaces that are allocated to the ESSS, issue this command from z/OS:

```
/D A,EYUX540
```

The last 4 bytes of the DSPNAME contain the SYSID of each CMAS.

NSYSLX=nnn

Set or increase the value to include the minimum number of linkage indexes (LXs) that are required by CICSplex SM. One LX is required for the Environment Services System Services (ESSS), so the minimum number of LXs required for use by CICSplex SM is one.

For more information about this parameter, see [z/OS MVS Initialization and Tuning Reference](#).

Coding the IEFSSNaa MVS subsystem initialization member

To define CICS as an MVS subsystem, code an entry in the IEFSSNaa member in the SYS1.PARMLIB library.

If you want to use the console message handling facility or to change the number of pipes that can be allocated in an EXCI address space, code the entry by using one of the following methods:

```
CICS,DFHSSIN,DFHSSIyy
```

or

```
SUBSYS SUBNAME(CICS)  
INITRTN(DFHSSIN)  
INITPARM(DFHSSIyy)
```

This entry is used for every CICS region that runs under MVS that you have IPLed with this version of the IEFSSN member. You do not have to specify both DFHSSIN and DFHSSIyy, however apart from the suffix yy, you must code the entry for each parameter using the exact format given in the example. The terms have the following meanings:

CICS

The name of the CICS subsystem.

DFHSSIN

The name of the CICS subsystem routine that initializes the console message-handling facilities and the number of pipes that can be allocated in an EXCI address space. If you omit this name, CICS is defined as an MVS subsystem, but none of the console message-handling facilities are enabled, and the default number of pipes that can be allocated in an EXCI address space is used. That default is 100.

DFHSSIyy

The name of a SYS1.PARMLIB member in which you have defined initialization parameters for message formatting and EXCI pipe allocation for the CICS subsystem. If you specify DFHSSIN

but omit DFHSSIyy, the DFHSSIN routine tries to use the parameters that are defined in member DFHSSI00.

If the DFHSSI00 member does not exist, the routine uses the default values:

- For message formatting the default values are defined in the DFHSSIN member. They are described in [“Default message-formatting initialization parameters”](#) on page 82.
- For EXCI pipe allocation, the fixed value is 100.

The IEFSSNaa member in the SYS1.PARMLIB library also contains the definitions for all the other subsystems required for this IPL of MVS; for example, JES2, IRLM and Db2.

The SYS1.PARMLIB(BPXPRMxx) parameters

If you use certain CICS functions and run many CICS systems in an LPAR, the z/OS UNIX System Services parameters, specified in [BPXPRMxx](#) members of SYS1.PARMLIB, might not be sufficient.

You must review the parameters in BPXPRMxx members based on use of following functions:

- Programs that run on open TCBs and use Unix System Services APIs, and programs that use other APIs that use Unix System Services, for example, IP CICS SOCKETS
- SSL TCBs specified by the **MAXSSLTCBS** system initialization parameters
- Java programs running in JVM servers
- Node.js applications

If you use Unix System Services, you must increase the **MAXPROCUSER** and **MAXPROCSYS** values. See [“Sizing MAXPROCUSER”](#) on page 79 and [“Sizing MAXPROCSYS”](#) on page 77 for guidance.

If CICS is configured to use SSL, you might have to increase the **MAXTHREADS** and **MAXTHREADTASKS** values.

If CICS is configured to use JVM servers, you might have to increase the **MAXTHREADS** and **MAXTHREADTASKS** values.

If you are running Node.js applications, each enabled NODEJSAPP resource requires a process and you should ensure that **MAXPROCUSER** and **MAXPROCSYS** are sized accordingly.

If your system uses two or more of these facilities, corresponding further increases in the values for these parameters is appropriate.

Sizing MAXPROCSYS

The z/OS **MAXPROCSYS** parameter specifies the maximum number of processes that can be active at the same time in the LPAR. **MAXPROCSYS** allows you to manage system resources by limiting the number of processes that the system is to support. If you set the **MAXPROCSYS** value too low, regions might abend because CICS cannot create a process when attempting to attach a TCB or when using a USS service. The value specified limits the total number of processes to be used across all address spaces on an LPAR, which includes other subsystems, utilities that use shell scripts, and so on.

CICS uses at least three processes for each CICS region. If you have many CICS regions in the LPAR, you must set your system limit to handle these processes. The following table explains which TCBs become processes, depending on the CICS system:

Table 7. TCBs that become processes		
TCB	Is the TCB always a process?	Description
Jobstep	Yes	The jobstep TCB is always created.
SO	Yes	The SO TCB is always created.

Table 7. TCBs that become processes (continued)

TCB	Is the TCB always a process?	Description
QR	Yes	The QR TCB becomes a process when an open TCB is attached (that is, L8, L9, X8, or X9).
SL	No	The SL TCB is created unless the system initialization parameter, TCPIP=NO , is specified.
SP	No	The SP TCB becomes a process, and the associated S8 TCBs are created and become threads only when SSL is used.
JVMSERVER (TP and T8)	Yes	Each enabled JVM server requires a single process regardless of how many threads it supports.
NODEJSAPP threads (one thread per NODEJSAPP)	Yes	Each NODEJSAPP runs as a process.
L8, L9, X8, and X9 open TCBs	No	If applications running on open TCBs use USS APIs, then a process is associated with the open TCB. The process is terminated when the open TCB is freed from the CICS task and returned to the dispatcher pool. An exception to this is when applications use the IP CICS Sockets interface, which uses USS APIs. With APAR PH29073 applied, CICS will not terminate the process associated with an L8 or L9 TCB when the TCB is returned to the dispatcher pool, if the IP CICS Sockets TRUE has been enabled in a CICS region.

To summarize:

- A minimum of three processes is required for each CICS region but there might be a need for more, dependent on the type of workload the CICS region is processing. You must calculate for each CICS region on the LPAR. The maximum number of processes required for a region depends on which additional TCBs you are running and what open TCBs are used for. If USS services are not used either directly by applications or through use of services listed in [Table 7 on page 77](#), then the minimum of three processes per region will suffice. For example, that applies to a region using L8 TCBs only for the CICS-Db2 attach because processing Db2 work for a CICS application does not use USS services.
- A CICS region using IP CICS Sockets uses USS processes, so you must take into account the peak sockets workload. For example, a region with MAXOPENTCBs=100 and using IP CICS Sockets with OTE=YES set in the EZACICFG configuration file, may require 103 processes. This can be calculated by counting the three processes for the Jobstep, SO and QR TCBs, plus 100 for the L8 TCBs that could be used by the IP CICS Sockets feature. How many of the 100 possible L8 TCBs are used by the IP CICS Sockets is governed by the **TCBLIM** parameter set in the EZACICFG configuration file, so that 103 figure should be adjusted accordingly. Note that the IP CICS Sockets interface does not use L9 TCBs nor X8 or X9 TCBs. Note also that when the IP CICS Sockets feature is configured with OTE=NO, the NTASKS option in the configuration file determines the number of non-CICS TCBs used. Each of these non-CICS TCBs nevertheless will consume a USS process in the same way as an L8 TCB does and needs to be included in the calculation.
- In some cases, a CICS region that is not configured to use IP CICS Sockets will still be consuming USS processes because of other uses of USS.

For example, if a CICS region is configured to accept SSL requests using a TCPIP SERVICE and also contains a JVM SERVER, such a region would require three processes for the Jobstep, SO and QR TCBs, plus one for the JVM server's TP TCB, and one for the SP TCB. The JVM server's T8 TCBs and the S8 TCBs used for SSL are not counted because these TCBs are associated with threads not processes.

An example of further use of USS processes would be use of WEBSERVICES as CICS pipeline processing involves reading configuration information from zFS files. In this case, the process is associated with an L8 TCB, so the peak number of L8 TCBs used for WEBSERVICES needs to be added to the process count. In these types of regions that do not use IP CICS Sockets, when the CICS task ends, the USS process is removed from the L8 TCB and is terminated. The act of terminating the process involves some asynchronous processing, which means that a so called "zombie process" will continue to be counted towards **MAXPROCSYS** momentarily until it is terminated. For heavy workloads, allow extra capacity in the number of processes to allow for these temporary zombie processes.

- The use of USS by other products on the LPAR (for example, IBM MQ channel initiator tasks) also needs to be factored in to the overall total for the LPAR.

You can issue the following command to give you a list of the processes that are running on your system. *addressspaceid* is the address space that you want to query.

```
D OMVS, A=addressspaceid
```

You can issue the following command to display the current and high water mark usage for **MAXPROCSYS**. If you run this command at system startup, and again when your system is stable running a workload, you can determine the most appropriate number for **MAXPROCSYS** and ensure the system is not close to the limit.

```
D OMVS,LIMITS
```

Sizing MAXPROCUSER

The z/OS **MAXPROCUSER** parameter specifies the maximum number of processes that a single user (that is, with the same UID) can have concurrently active. To use USS, a z/OS user ID should be configured to have a z/OS UNIX UID, and therefore **MAXPROCUSER** specifies the maximum number of processes that a single user ID can have. For use of USS services within a CICS region, the UID used is always the one associated with the CICS region user ID, not the ID of the signed-on end user. For each region, calculate the maximum number of processes that can be used. Set **MAXPROCUSER** equal to or higher than the sum of the maximum number of processes for all regions running with the same region user ID. If you have multiple region user IDs, set **MAXPROCUSER** to the highest required across all region user IDs.

Related information

For more information about changing BPXPRMxx parameters, monitoring system limits, and calculating values for system resources, see [z/OS UNIX System Services Planning](#).

The console message-handling facility

The console message-handling facility is an optional feature of the CICS subsystem that can affect the appearance of CICS messages displayed on an MVS console. It is effective when you specify **FORMATMSG=YES** as a parameter in the message-formatting initialization member for the CICS subsystem.

The subsystem reformatting is enabled when at least one of the following applications is executing in the MVS image where the subsystem is defined:

- Any version of CICS Transaction Server
- A message automation subsystem, such as NetView, which enables the MVS subsystem console message broadcasting service

The console message-handling facility affects the messages that are displayed on MVS system consoles in the following ways:

- The subsystem tries to ensure that all console messages issued by all CICS regions have a standard format. The format is:

```
+DFHnnnn  APPLID  MESSAGE-TEXT
```


In this message:

+DFHnnnn

- Begins in column 1
- The plus sign (+) that precedes **DFHnnnn** is added by MVS to indicate that a problem-state program issued the message. It is not present when CICS issues the message while it is in supervisor state.

APPLID

- Begins in column 13
- The applid inserted into the message is the specific application identifier. This identifier is specified in the **APPLID** system initialization parameter.

MESSAGE-TEXT

- Begins in column 22.
 - If the original CICS message is long (greater than 113 characters), the message text begins on the next line.
- The subsystem adds route codes specified in the **ROUTECD** subsystem initialization parameter, so the messages might be sent to more destinations.
 - The subsystem reformats messages for all CICS releases.
 - The subsystem does not reformat messages that are issued by a CICS region that has not yet determined its applid. These messages include those that are issued while processing the system initialization table and its overrides.
 - The subsystem routine that reformats the messages does not receive control until after the message has been recorded in the CICS job log. Therefore, the reformatting is not usually apparent in the job log.
 - Messages issued by the message domain already contain the applid. The subsystem does not insert the applid into such messages, but it might insert blank characters to cause alignment into standard locations.
 - Multiple line messages might include a first control line with the applid inserted as well as being included in the MESSAGE-TEXT at its normal position.
 - If the original CICS message is long, the addition of the applid might cause the message to exceed the maximum length for an MVS console message. In this case, the original message is suppressed and does not appear on the console, and the reformatted message is issued using the MVS multiple-line console message service to split the message over several lines. Both the original message and perhaps several instances of the reformatted multiple-line message appear in the job log, but only one copy of the reformatted message is displayed on the console.
 - For some messages in which the applid normally follows a time and date stamp, the insertion of the applid in the standard position results in the applid being duplicated in the message. For these messages, the subsystem eliminates the time and date stamp, because these are available from other sources, and only one occurrence of the applid is shown.

Specifying the DFHSSIyy message-formatting initialization member

You can specify message-formatting initialization parameters for the CICS subsystem in a member DFHSSIyy of the SYS1.PARMLIB library, where yy is the suffix that identifies the SYS1.PARMLIB member used to define the CICS subsystem.

These parameters are FORMATMSG, HIDEPASSWORD, and ROUTECODES. Code the parameters in columns 1 through 71 of the DFHSSIyy member, like this:

```
FORMATMSG=YES,HIDEPASSWORD=YES,ROUTECD=(1,2)
```

or with additional route codes:

```
FORMATMSG=YES  
HIDEPASSWORD=YES
```



```
ROUTECODES=(1,2,
3,4,
5,6)
```

FORMATMSG={YES|NO}

Specifies whether the CICS applid is to be inserted into all DFH console messages that do not use the CICS message domain.

YES

Insert CICS applid into messages.

NO

Do not insert CICS applid into messages.

HIDEPASSWORD={YES|NO}

Specifies whether to mask the password or password phrase from MODIFY commands used to enter the CICS sign-on transactions at an MVS console.

YES

Mask the password or password phrase.

NO

Do not mask the password or password phrase.

ROUTECODES=(n1[,n2])

n1, n2... are numbers representing generic routecodes that are added to *all* DFH console messages issued by CICS. The routecodes 1-12 have special meanings:

1

Main console action

2

Main console information

3

Tape pool

4

Direct access pool

5

Tape library

6

Disk library

7

Unit record pool

8

Teleprocessing control

9

System security

10

System error/maintenance

11

Programmer information

12

Emulators

The status of other routecodes are listed:

13-20

Available for customer use

29-40

Reserved

41-128

Available to authorized programs only

For more information about these routing codes, see [z/OS MVS Initialization and Tuning Reference](#).

Default message-formatting initialization parameters

You can define message-formatting initialization parameters for the CICS subsystem in a member DFHSSIyy of the SYS1.PARMLIB library.

To use parameters defined in a DFHSSIyy member other than the DFHSSI00 member, you must specify DFHSSIyy on the IEFSSNaa member in the SYS1.PARMLIB library used to define CICS as an MVS subsystem. If you do not specify DFHSSIyy, the DFHSSIN routine tries to use the parameters that are defined in the DFHSSI00 member. If the DFHSSI00 member does not exist, it uses the default parameters that are defined in the DFHSSIN routine.

If you specify DFHSSIyy but it does not exist, the DFHSSIN routine uses the default message-formatting initialization parameters that are defined in the DFHSSIN routine.

The default message-formatting initialization parameters defined in the DFHSSIN routine are as follows:

```
FORMATMSG=YES,HIDEPASSWORD=YES  
(generic routecodes are not added to messages)
```

The default facilities:

- Insert the CICS applid into the CICS console message between the message identifier and the message text. The applid is inserted into only those console messages (starting with DFH) that do not use the CICS message domain. The CICS message domain inserts the CICS applid into all messages that it handles.

If the original message is a long one, insertion of the CICS applid might cause the message to exceed the maximum length for an MVS console message. In this case, the original message is suppressed and does not appear on the console, and the reformatted message is issued using the MVS multiple-line console message service to split the message text over several lines. Both the original message and perhaps several instances of the reformatted multiple-line message appear in the job log, but only one copy of the reformatted message is displayed on the console.

- Examine each MODIFY command to see if it resembles a MODIFY CICS,CESN ... command. If the MODIFY command contains an old or new password (PS=xxxx,NEWPS=xxxx), the default facilities obliterate the password with asterisks. If the MODIFY command does not contain a password, the password you enter at the MVS console is masked.
- If your primary subsystem is JES3, the old and new passwords still appear in the JES3 hardcopy log. JES3 records the MODIFY command before the CICS message formatting subsystem can obliterate the password. (This processing does not happen when the primary subsystem is JES2.) The passwords are suppressed from the console for both JES2 and JES3.

If you do not specify DFHSSIN in the IEFSSNaa entry that defines CICS, the message handling facilities are not enabled. Also, if you run CICS as a started task, you cannot use the name “CICS” for the procedure name.

Activating message formatting

The next MVS subsystem to invoke the subsystem console message broadcasting service of MVS console support activates message-handling. Before activating message-handling, you must define CICS as an MVS subsystem with support for console message-handling, and also specify the message-formatting parameters in the DFHSSIyy member of the SYS1.PARMLIB library.

Message-handling is activated when you start a supported CICS region or if an automated-operation program, such as NetView, is active in the MVS image. See [“The console message-handling facility” on page 79](#) for more information on console message-handling.

A newly started CICS region determines its own applid during initialization. Until the applid is known, message-formatting cannot operate. Therefore, messages issued very early in CICS initialization are not formatted.

Modules required to use the console message-handling facilities

To use the console message-handling facilities that are provided by the MVS subsystem functions of CICS, the CICS modules, DFHSEN, DFHSSGC, DFHSSMGT, and DFHSSWT must be available at MVS IPL time.

DFHSEN

The module that cleans up CICS resources at end-of-memory and at end-of-task.

DFHSSGC

The subsystem generic connect module that connects an active CICS region to the CICS subsystem.

DFHSSIN

The CICS subsystem initialization module.

DFHSSMGT

The subsystem message table that contains the text of messages for the subsystem interface modules.

DFHSSWT

The subsystem interface write-to-operator (WTO) router that determines whether to route WTO calls to the appropriate CICS-dependent modules.

These modules must reside in the LPA or in an APF-authorized library in the MVS linklist:

- The modules DFHSSIN and DFHSSMGT, installed in the *hlq*.SDFHLINK library, must reside in an APF-authorized library in the MVS linklist.
- The DFHSEN module, installed in the *hlq*.SDFHLPA library, must reside in the LPA.
- The modules DFHSSGC and DFHSSWT, installed in the *hlq*.SDFHLPA library, must reside either in the LPA or in an APF-authorized library in the MVS linklist.

The LINDEX parameter in the DFHISTAR installation job defines *hlq*.

The current versions of these modules are compatible with earlier CICS releases that support console message handling.

For information about adding modules that are installed in the *hlq*.SDFHLINK library to the MVS linklist, see [“Installing CICS-required modules in the MVS linklist” on page 69](#).

For information about adding modules installed in the *hlq*.SDFHLPA library to the LPA, see [“Installing CICS modules in the MVS link pack area” on page 98](#).

EXCI pipe allocation

The external CICS interface is an application programming interface that enables a non-CICS program (a client program) running in MVS to call a program (a server program) running in a CICS region and to pass and receive data by means of a communications area.

The CICS application is invoked as though it is linked to by another CICS application program.

This programming interface allows a user to allocate and open sessions or pipes which operate in "half-duplex", flip-flop" mode, to a CICS region and to pass distributed program link requests over them. The multiregion operation facility of CICS interregion communication supports these requests, and each pipe maps onto one MRO session, in which the client program represents the sending process and the CICS server region represents the receiving process. A default limit of 100 pipes for each EXCI address space applies.

Specifying the EXCI pipe allocation limit

Specify the EXCI pipe allocation limit for the CICS subsystem in a member DFHSSIyy of the SYS1.PARMLIB library, where yy is the suffix that identifies the SYS1.PARMLIB member used to define the CICS subsystem. The parameter is LOGONLIM.

Code the parameter in columns 1 through 71 of the DFHSSIyy member, like this:

```
LOGONLIM=200
```

LOGONLIM=nn

The minimum and maximum values that can be specified for nn are 100 and 250.

If you omit the parameter or the value specified lies outside the allowed range, CICS assumes a limit of 100.

The EXCI pipe allocation limit

CICS publishes the limit if it is determined during subsystem initialization, by creating a system-level name token pair formatted like this:

```
Name: input, fixed length 16 byte type
  Bytes 0-7 : The character string 'DFHIRP '
  Bytes 8-15: The character string 'LOGONLIM'
Token: output, fixed length 16 byte type
  Bytes 0-3 : The logon limit, held as fullword binary file
  Bytes 4-15: Reserved, set to nulls
```

You can use the callable service, IEANTRT, to retrieve the token. Invoke IEANTRT with level IEANT_SYSTEM_LEVEL (EQU 4). The return code is interpreted in the following way:

0

The name and token pair exists and the token has been retrieved. The logon limit can be extracted from the token.

4

The name and token pair does not exist. The logon limit is assumed to be 100.

Any other value indicates that the callable service has detected an error.

Default EXCI pipe allocation limit initialization parameter

Define the EXCI pipe allocation limit parameter for the CICS subsystem in a member DFHSSIyy of the SYS1.PARMLIB library.

To use parameters defined in any DFHSSIyy member except the DFHSSI00 member, specify DFHSSIyy in the IEFSSNaa member in the SYS1.PARMLIB library used to define CICS as an MVS subsystem. For more information on IEFSSNaa, see [“Coding the IEFSSNaa MVS subsystem initialization member” on page 76](#).

- If you do not specify DFHSSIyy, the DFHSSIN routine tries to use the parameters that are defined in the DFHSSI00 member.
- If the DFHSSI00 member does not exist, the DFHSSIN routine uses the default parameters that are defined in the DFHSSIN routine.
- If you specify DFHSSIyy but it does not exist, the DFHSSIN routine uses the default parameters that are defined in the DFHSSIN routine.

The default EXCI pipe allocation initialization parameter defined in the DFHSSIN routine is LOGONLIM=100.

Installing the CICS SVCs

Install the current level of the CICS Type 3 SVC, DFHCSVC, and the high performance option (HPO) SVC before you attempt to start a region.

If IBM changes the Type 3 SVC, for example at a new release or because of a service update, you must reinstall the current level of the CICS Type 3 SVC into the link pack area (LPA) and perform an IPL with the CLPA option.

To install the CICS Type 3 SVC, define the CICS SVCs to z/OS, install the DFHCSVC module into the LPA, and specify the DFHCSVC number on the **CICSSVC** system initialization parameter.

Alternatively, if you have the required authorization, you can use the DFHCSVCU utility program to install the CICS Type 3 SVC dynamically, without the need to perform an IPL of the z/OS system. For further information about running programs that require APF authorization, see [Managing system security - APF-authorized library list in z/OS MVS Initialization and Tuning Reference](#). For further information about the DFHCSVCU utility program, see [DFHCSVCU](#).

Defining the CICS SVCs to your z/OS system

1. Define both the CICS Type 3 SVC and the HPO SVC to your z/OS system by specifying SVC Parm statements. You define the CICS SVCs in an IEASVCxx member of the SYS1.PARMLIB library, using SVC Parm statements. See [z/OS MVS Initialization and Tuning Guide](#) and [z/OS MVS Initialization and Tuning Reference](#) for a description of the SVC Parm statements.
2. If you are using the default SVC numbers, the following CICS entries are shown:

```
SVC Parm 216,REPLACE,TYPE(3),EPNAME(DFHCSVC)
SVC Parm 215,REPLACE,TYPE(6),EPNAME(DFHHPSVC) [Only required for HPO]
```

For the current SVC modules, you specify the EPNAME parameters as in the sample CICS entries.

3. If you have a version of the DFHHPSVC module from an earlier release of CICS already link-edited into your z/OS nucleus, you do not have to replace it with the latest version. Versions of the DFHHPSVC module from earlier releases of CICS are compatible with the current release. The version of the DFHHPSVC module from the current release of CICS is compatible with earlier releases of CICS. The CSECT name (EPNAME) of the version of the DFHHPSVC module from earlier releases is IGC215 or IGCnnn, if SRBSVC=nnn was used as a CICS system generation parameter in the earlier release.
4. If you are not using the default SVC numbers, change the values 215 and 216 to the SVC numbers you have chosen.
5. Select the required IEASVCyy member by coding the SVC parameter (SVC=yy) in a SYS1.PARMLIB member (IEASYSxx), which you use to perform an IPL of your z/OS system. When you code new SVC numbers, they do not come into effect until you next perform an IPL of your z/OS system.

Installing the DFHCSVC module into the LPA

Use the following guidelines when you install the DFHCSVC module into the LPA:

- Do not change DFHCSVC attributes.
- Do not relink-edit the DFHCSVC module to install it into the LPA. The term *install* means move or copy a module into the LPA by using SMP/E, or a copying method that relocks the copied modules when the target data set has a smaller block size than the data set you are copying from.
- The DFHCSVC module, as supplied, has the attributes AMODE(31) and RMODE(ANY); do not change these attributes.

For further information about installing the DFHCSVC module in the LPA, see [“Installing CICS modules in the MVS link pack area” on page 98](#).

Specifying the DFHCSVC number on the CICSSVC system initialization parameter

The current version of the CICS SVC module is compatible with all earlier releases of CICS, which enables you to run your earlier CICS regions with current regions in the same MVS image.

CICS contains a test to verify that it is using the correct level of the CICS DFHCSVC module. If CICS calls an SVC module using the SVC number specified on the **CICSSVC** system initialization parameter and that module is not at the current level, CICS issues message DFHKE0104. As a result of this message, CICS either abends with a system dump or prompts the operator to enter an alternative SVC number, depending on the option specified on the **PARMERR** system initialization parameter.

Configuring related security

You must use RACF dataset security to ensure that only CICS regions and authorized users who need access to programs in the authorized libraries SDFHAUTH and the SEYUAUTH. For more information on RACF dataset security, see [Protecting data sets in z/OS Security Server RACF Security Administrator's Guide](#).

If only a subset of the authorized users (above) need access to DFHCSVC program, use [program security](#) to further restrict this access to a smaller RACF group.

Using more than one version of the CICS Type 3 SVC

You might have to use more than one version of the CICS Type 3 SVC; for example, to test service applied to the DFHCSVC module while using the current version in a production system.

You can run several CICS regions, at different release levels, in the same MVS image, with each region using its own version of the CICS SVC. However, if some of those regions use MRO, all regions that use MRO must use the latest CICS Type 3 SVC (DFHCSVC module) and the latest DFHIRP module. For information about using the latest SVC with earlier releases of CICS, see [“MRO between different CICS releases with a changed SVC number”](#) on page 87.

To use more than one version of the CICS SVC, either use the DFHCSVCU utility program, or rename the new SVC module in the LPA, and then respecify the SVC in the SVC Parm statements.

You must have authorization to use the DFHCSVCU utility program. For more information about this program, see [“Installing the CICS SVCs”](#) on page 84.

To rename the new CICS SVC module, use the renaming facility of ISPF or IEBCOPY, or the TSO command RENAME, renaming the module to a unique name of your choice. Use SMP/E to rename the CICS SVC module in the SDFHLPA library. Use the SMP/E RENAME command to inform SMP/E of the change to the name of the CICS SVC module. Therefore, if you later use SMP/E to apply service to that module, the service is applied to the renamed module in the LPA, and *not* the DFHCSVC module.

For example, you might want to use an SVC number 255 for a test CICS region, and the default CICS SVC number 216 for your production system:

1. Create and apply an SMP/E USERMOD to rename the new CICS SVC module:

```
++USERMOD (umod1).  
++VER(C150) FMID(HCI7300).  
++RENAME (DFHCSVC) TONAME(newname).
```

2. Specify the number 255 for the new CICS SVC version by adding an appropriate statement to the list of SVC Parm statements. That list then reads:

```
SVC Parm 216,REPLACE,TYPE(3),EPNAME(DFHCSVC)  
SVC Parm 215,REPLACE,TYPE(6),EPNAME(DFHHP SVC) [Only required for HP0]  
SVC Parm 255,REPLACE,TYPE(3),EPNAME(newname) [New CICS SVC version]
```

The **EPNAME** parameter for the new CICS SVC specifies the module name, not the CSECT name, for the new CICS SVC module.

All the SVC Parm statements apply to the same IEASVCxx member of the SYS1.PARMLIB library.

3. Perform another IPL of MVS to enable all the SVC versions specified in the SVC Parm statements. After you perform another IPL of MVS, you can use both versions of the CICS SVC, provided that both regions do not use MRO concurrently. If both systems use MRO, only the new, latest version of the SVC and the latest DFHIRP module are used by both regions.
4. In your production system, specify the number of the current CICS SVC in the **CICSSVC** system initialization parameter. Similarly, in the test system, specify the number of the new CICS SVC version.

Installing newest release CICS Type 3 SVC and DFHIRP modules supplied through maintenance

Ahead of the general availability (GA) of a new CICS TS release, IBM may ship maintenance that contains the newest release type 3 SVC module and the IRP module required for MRO communications, on currently supported CICS TS releases, to facilitate the evaluation or quicker adoption of a new CICS release. You can opt in and schedule an upgrade of these LPA modules to fit in with your scheduled z/OS IPLs ahead of the CICS GA.

About this task

The supplied new release modules are named as DFHNCSVC and DFHNIRP in the SDFHLPA library. Unless you opt in as instructed below, application of this maintenance has no effect. To utilize the new release modules, you must rename them for CICS use.

Use SMP/E to rename the CICS SVC module and IRP module. Use the SMP/E RENAME command to inform SMP/E of the change to the name of the CICS SVC module. Therefore, if you later use SMP/E to apply service to that module, the service is applied to the renamed module in the LPA.

The procedure below requires you to create two SMP/E USERMODs. For more information about how to code a USERMOD, see [SMP/E for z/OS Reference](#).

Procedure

1. Create and apply an SMP/E USERMOD to rename the existing CICS SVC and IRP modules to a name of your choice:

```
++USERMOD (umod1).  
++VER(C150) FMID(HCI7300).  
++RENAME (DFHCSVC) TONAME(old_csvc_name).  
++RENAME (DFHIRP) TONAME(old_irp_name).
```

2. Create and apply an SMP/E USERMOD to rename the new release CICS SVC and IRP modules:

```
++USERMOD (umod2).  
++VER(C150) FMID(HCI7300) PRE(umod1).  
++RENAME (DFHNCSVC) TONAME(DFHCSVC).  
++RENAME (DFHNIRP) TONAME(DFHIRP).
```

What to do next

When the renaming is complete, perform an IPL of the z/OS system to activate use of the new release modules. These modules are downward compatible with all existing CICS TS releases.

When the new release is made generally available and installed, you can add the SDFHLPA library for the new release to the LPA concatenation as normal but an immediate IPL is not required. The type3 SVC and IRP modules applied by this maintenance procedure are the same as the new release modules, so are sufficient until the next scheduled IPL, at which time you can revert to using the new release SDFHLPA library. The other CICS modules that are contained in SDFHLPA are not release specific and are upward and downward compatible, so earlier release modules operate correctly with the new GA release.

MRO between different CICS releases with a changed SVC number

If a CICS TS 5.6 region, and other CICS regions from earlier releases, in the same MVS image use MRO, all the regions must use the CICS TS 5.6 SVC module.

If, when you install the CICS TS 5.6 SVC in the LPA, you give the SVC a number different from the number defined to the earlier CICS regions, you must respecify the SVC number. On each CICS region from an earlier release that will use the CICS TS 5.6 SVC, specify the new SVC number on the CICSSVC system initialization parameter.

Selecting the high-performance option

The high-performance option (HPO) is for users whose top priority is to optimize terminal response times and maximize transaction throughput. HPO improves performance by reducing the transaction path length; that is, the number of instructions required to service each request.

Use of HPO potentially allows CICS application programs to bypass all MVS integrity controls. If you decide to use HPO, ensure that the application programs used on your CICS system meet your own installation's integrity requirements. The code to support the SNA authorized path feature of HPO, which is the improved path through SNA, is in CICS.

Defining DFHHP SVC to MVS

Define the DFHHP SVC module to MVS as a Type 6 SVC; the default HPO SVC number defined in the DFHSIT module is 215.

To change the default Type 6 SVC number:

1. Define the new number to MVS. See [“Installing the CICS SVCs”](#) on page 84.
2. Define the new number to CICS by using the SRBSVC system initialization parameter.

If you are not using HPO, do not load the DFHHP SVC module into the MVS nucleus. You choose to use HPO explicitly by coding HPO=YES in the system initialization table.

Loading module DFHHP SVC

Before you can use HPO, ensure that the HPO SVC module is included in the MVS nucleus.

Use one of the following methods:

- Copy the DFHHP SVC module into SYS1.NUCLEUS, renaming it to IGC215 or the appropriate name if you are not using the default, and specify it on an INCLUDE statement in the NUCSTxx member of the SYS1.PARMLIB library. You must also specify the name of the NUCSTxx member on the NUCST statement of the LOADxx member of the SYS1.PARMLIB library. The NUCSTxx method provides you with greater flexibility in customizing the MVS nucleus than the NMLDEF method described in the second method.

For further information about coding a NUCSTxx member and about a comparison with the NMLDEF macro, see the [z/OS MVS Initialization and Tuning Guide](#).

- Copy the DFHHP SVC module into SYS1.NUCLEUS and specify it in a nucleus module list (NML) for CICS, created using the NMLDEF macro shown in the sample job. This NML selects the CICS members in SYS1.NUCLEUS that are to be loaded into the MVS nucleus, and eliminates the requirement for the MVS nucleus to be re-link-edited for the DFHHP SVC module or any other module in the MVS nucleus.
- For information about coding an NMLDEF macro, see [z/OS MVS Programming: Authorized Assembler Services Reference \(Volume 3\)](#). The sample job stream loads the CICS Type 6 SVC into the MVS nucleus:

```
//LOADSVC   JOB 'accounting info',MSGCLASS=A,CLASS=A
//NMLDEF    EXEC ASMHCL
//C.SYSIN   DD *
IEANCnnn NMLDEF NUCL=DFHHP SVC
//L.SYSLMOD DD DSN=SYS1.NUCLEUS,UNIT=3380,DISP=OLD
//L.SYSIN   DD *
           NAME IEANCnnn
/*
//
```

nnn is the number of the CICS NML, in the range 001 through 256. Choose the value of *nnn* to be unique in your MVS nucleus.

Removing existing DFHHP SVC modules from the MVS nucleus

If you have a version of the DFHHP SVC module from an earlier release of CICS already installed in your MVS nucleus, you do not have to replace it with the latest version. Versions of the DFHHP SVC module from earlier releases of CICS are compatible with the current release. The version of the DFHHP SVC module from the current release of CICS is compatible with earlier releases of CICS.

However, you can remove a link-edited version of the DFHHP SVC module from the MVS nucleus by running a link-edit job to replace the existing version of the nucleus with one that does not contain the module to be removed, in one of the following ways:

- If the existing nucleus-resident DFHHP SVC module is known to SMP/E, use the SMP/E UCLIN statement to remove the module entry.
- You must link-edit the nucleus module, IEANUC0x, with the scatter (SCTR) attribute. If you do not do this, MVS enters a non-restartable wait state at system initialization.
-

Defining EYU9XENF as a TSO/E authorized command

Define the EYU9XENF command during set up to streamline problem diagnosis later, if required.

About this task

Defining EYU9XENF as a TSO/E authorized command will allow you to use the Environment Services System Services (ESSS) information display utility, should you be requested to do so by customer support when diagnosing problems with MVS systems.

For more detailed information about the ESSS information display utility, see [Using the ESSS Information Display Utility \(EYU9XENF\)](#).

Procedure

1. Add EYU9XENF to the 'AUTHCMD NAMES' section of the IKJTSOxx SYS1.PARMLIB member used by the MVS image.
2. Issue from TSO a 'PARMLIB UPDATE(xx)' command to implement the changes made to the IKJTSOxx member.
3. Ensure that CPSM560 .SEYUAUTH is in the TSO users STEPLIB concatenation.

Results

EYU9XENF is now an authorized TSO/E command, and will be available for use should it be necessary in later problem diagnosis.

Defining CICS regions as applications to SNA

To use SNA LUs with CICS, ensure that your CICS regions are defined to SNA before you attempt to run them.

To define your CICS regions to SNA as SNA application programs:

1. Define SNA application program minor nodes for your CICS regions, by specifying APPL definition statements in a member of the SYS1.VTAMLST library or your own *user.VTAMLST* library.
2. Issue a VARY ACT command to activate the APPL definitions and enable the CICS regions to connect to SNA.
3. Ensure that you have properly defined your SNA LUs for connection to CICS. You must do so if you intend using CICS autoinstall. For those LUs for which you want to use autoinstall, code LOGON mode table entries that match the model TYPETERM and TERMINAL definitions that CICS uses. You can either code your own autoinstall models, or use the CICS-supplied model definitions that are generated for you when you initialize the CICS system definition data set.

Define SNA application program minor nodes for your CICS regions, by specifying APPL definition statements in a member of the SYS1.VTAMLST library or your own user.VTAMLST library. user.vtamlst will ONLY work if user.vtamlst is a concatenated data set in the //VTAMLST DD in the VTAM PROC, usually found in SYS1.PROCLIB or SYS1.IBM.PROCLIB.

For further information about defining SNA resources, see [z/OS Communications Server: SNA Network Implementation Guide](#) and [z/OS Communications Server: SNA Resource Definition Reference](#).

The following topics provide more detailed information about defining your CICS regions to SNA.

Defining specific APPL definitions and APPL parameters to SNA

To define a CICS region to SNA, specify the minor node name to be used for the CICS region on the SNA APPL definition statement. Include specific parameters on the SNA APPL statement.

Defining specific CICS APPL statements to SNA

For example, you might use the following definition for the CICS region to be identified as CICSHTH1:

```
*****
* Specific APPL definition for CICS region CICSHTH1
*****
CICSHTH1 APPL AUTH=(ACQ,VPACE,PASS),VPACING=0,EAS=5000,PARSESS=YES X
          SONSCIP=YES,LUAPFX=XX
*****
```

- Code CICSHTH1 on the CICS system initialization parameter **APPLID** to define the SNA application identifier to CICS.
-
- See [“Data set naming conventions” on page 170](#) for information about the naming convention that is used for the CICSHTH1 applid.

SNA APPL parameters for CICS regions

ACBNAME=acbname

Specifies the minor node name (*acbname*) that is assigned to this application. This name must be unique in the domain. If you do not specify this parameter, the name of the SNA APPL statement is taken.

AUTH=(ACQ,VPACE[,PASS])

Allows CICS to acquire LUTYPE 6 sessions. VPACE allows pacing of the intersystem flows. Specify PASS if you intend to use the **EXEC CICS ISSUE PASS** command to pass existing terminal sessions to other SNA applications.

EAS=number

Specifies the number of network-addressable units. The number must include the total number of parallel sessions for this CICS system.

HAVAIL=YES

Indicates that the application supports XRF sessions and can initiate XRF sessions.

LOGMODE=name

(For CICS-to-CICS APPC systems.) Defines the name of the MODE table that contains the LU6.2 MODEENT for the secondary sessions.

LUAPFX=string

Specifies the prefix characters of the LU alias to be assigned when a dynamically generated cross-network CDRSC with NQNMODE=NQNAME is created for a session with CICS. SNA concatenates the characters specified with the next sequential number available to form an SNA-generated LUALIAS name for the cross-network dynamic CDRSC.

string

Indicates the two characters to be used as the prefix for all dynamically generated LUALIAS names for dynamic cross-network CDRSCs in session with the CICS region defined by the APPL statement. Take into account the SNA naming conventions when choosing this prefix. For CICS information about specifying the LU alias string, see [“Choosing an LUAPFX value” on page 94](#).

SNA deletes a dynamically generated LU alias after a terminal session is closed, or the last session of an APPC parallel sessions connection is closed, and the CDRSCTI-specified timeout interval has expired. The permitted range of timeout values is 1 second to 7 days, but generally the default of 8 minutes is acceptable in most situations. The CDRSCTI timer does not start until no more sessions that involve the resource represented by a CDRSC remain.

For more information about CICS support for the SNA dynamic LU alias facility, see [“SNA LU alias facility” on page 92](#).

PARSESS=YES

Specifies LUTYPE 6 parallel session support.

PERSIST=MULTI

Indicates that the application supports Multi Node Persistent Sessions. For further information, see [z/OS Communications Server: SNA Network Implementation Guide](#).

SONSCIP=YES

Specifies session outage notification (SON) support. SON enables CICS, in certain cases, to recover a session after session failure without operator intervention.

VPACING=number

Specifies the maximum number of normal-flow requests that another logical unit can send on an intersystem session before waiting to receive a pacing response. Start with a value of 5.

SNA version and release level indicator

The terminal control modules in CICS are assembled against z/OS Communication Server.

You can use any release of z/OS Communication Server. For details of the minimum level of products that you can use with the current release, see [Detailed system requirements](#).

CICS can communicate with different levels of z/OS Communication Server. It can find out which level you are using and the level of function that is available. So you can upgrade CICS and z/OS Communication Server at different times. CICS finds out whether extra function is available when a new version of z/OS Communication Server is installed, and produces a message if the function is not being exploited fully.

Defining cross-domain services when using SNA

If you want to use SNA services to access a CICS region on another MVS image, you must ensure that the required cross-domain services are defined to the SNAs involved.

For example, to be able to use an SNA APPC connection between a CICS region (applid CICSHTH1) on MVS image MVSH and a CICS region (applid CICSJA1) on MVS image MVSJ:

1. Define the cross-domain services (CDRSC) for accessing CICSJA1 in a member of the SYS1.VTAMLST library, or your own user.VTAMLST library, for MVSH.
2. Issue a VARY ACT command on MVSH to activate the CDRSC definition for accessing CICSJA1.
3. Define the cross-domain services (CDRSC) for accessing CICSHTH1 in a member of the SYS1.VTAMLST library, or your own user.VTAMLST library, for MVSJ.
4. Issue a VARY ACT command on MVSJ to activate the CDRSC definition for accessing CICSHTH1.

Here is an example:

1. Create the following CDRSC definition in a member of the VTAMLST library on MVSH:

```
CDIDHAJ1 VBUILD TYPE=CDRSC
*****
* CDRSC for access to applid CICSJA1 on MVSJ
```

```
*****
CICSHAJ1 CDRSC CDRM=IYAMCDRM  MVSJ
```

2. Issue the following command on MVSH to activate the cross-domain services to CICSHAJ1 on MVSJ:

```
/V NET,ACT,ID=CDIDHAJ1
```

3. Create the following CDRSC definition in a member of the VTAMLST library on MVSJ:

```
CDIDH1 VBUILD TYPE=CDRSC
*****
* CDRSC for access to applid CICSH1 on MVSH
*****
CICSH1 CDRSC CDRM=IYALCDRM  MVSH
```

4. Issue the following command on MVSJ to activate the cross-domain services to CICSH1 on MVSH:

```
/V NET,ACT,ID=CDIDH1
```

SNA LU alias facility

Specifying a prefix string on the LUAPFX parameter of the CICS APPL statement indicates that SNA is to generate LUALIAS names for dynamic cross-network CDRSCs in session with the CICS region defined by the APPL statement.

In this way, CICS can use an LU alias for autoinstalled terminals and workstations and ensure unique names in a CICSplex comprising terminal-owning and application-owning regions. SNA generates the LUALIAS names dynamically.

CICS supports both the predefined and dynamic forms of the SNA alias function only where shown in the following table:

	CICS-to-CICS APPC connections (APPL definitions)		APPC devices (LU definitions)				Terminals	
	Synclevel 1	Synclevel 2	Synclevel 1		Synclevel 2			
	Predefined alias only		Predefined alias	Dynamic alias	Predefined alias	Dynamic alias	Predefined alias	Dynamic alias
SNA	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
CICS	Yes	No	Yes	Yes	No	No	Yes	Yes

- The LU alias is used as the NETNAME for terminals and workstations that log on to a CICS region.
- CICS does not support LU alias for synclevel 2 connections (LUTYPE 6.1 and 6.2), ignores any LU alias for these LU types, and continues to use the network name defined in the SNA APPL statement.

Dynamic LU alias support

CICS supports the use of a dynamic LU alias for CICS terminals and workstations that are autoinstalled only.

You enable dynamic LU alias support by specifying LUAPFX on the SNA APPL definition for any CICS terminal-owning region that can receive duplicate netnames. Also, when starting SNA, specify the following options on the SNA START command:

- NQNM=NAME.
- CDRSCTI=*n* to specify the length of time that the session name lasts after the last session has logged off.
 1. Make the time specified on CDRSCTI long enough to cover any time interval specified on CICS START commands that are issued against a terminal resource that uses a dynamic LU alias. This requirement applies to STARTS with a delay that run on both a TOR or AOR. If the CDRSCTI time is

not long enough, a resource can log off and then log back on again with a different network name and thus a different TERMID.

2. Also, the CDRSCTI time interval must be greater than that specified on the CICS [AILDELAY](#) system initialization parameter. However, if your applications have no dependency on the network name or TERMID, you can disregard CDRSCTI or set it to 1.

SNA generates a dynamic LU alias only if LUAPFX is specified on the CICS APPL statement and the resource comes from another network. That is, it has a different network name from the network to which the CICS region belongs.

When to use dynamic LU alias

Use dynamic LU alias when you are using autoinstalled cross-network terminals or duplicate network names. For example, in the following circumstances:

- Your cross-network terminals and workstations that log on to CICS are mainly autoinstalled.

The CICS region receives logons from terminals and synclevel 1 connections (both parallel and single sessions) and those logons (or binds) are from cross-network resources that might have duplicate network names.

However, be aware that synclevel 1 connections can become synclevel 2 in the future. For example, if you have a connection between a TXSeries® and CICS TS, the connection is synclevel 1; but, if you change to using TXSeries with a PPC gateway, synclevel 2 is used. CICS does not support dynamic LU aliases for synclevel 2 APPC connections.

- An AOR receives shipped terminals or connections with duplicate network names from different TORs.

Predefined LU alias support

CICS supports the use of a predefined LU alias for CICS terminals and workstations that are explicitly defined and those that are autoinstalled.

You can also use a predefined LU alias for CICS regions that communicate using CICS intersystem communication. You enable predefined alias support by specifying `LUALIAS=alias` on any cross-domain resource (CDRSC) that requires a specific alias. A terminal or APPC synclevel 1 workstation that is defined to CICS on an explicit resource definition (that is, it is not autoinstalled) and is in a different network, requires a CDRSC definition with a specific alias on the LUALIAS parameter. This alias overrides the dynamic generation of an alias where LUAPFX is specified on the CICS region's APPL statement. To ensure that CICS can match the SNA LU alias with the installed terminal definition, the LUALIAS value must match the NETNAME specified on the CICS TERMINAL resource definition.

An LUALIAS option in the CDRSC is effective if the resource comes from another SNA domain. That is, it is not used if the resource comes from the same MVS image, but is used if the resource comes from another MVS image regardless of whether it is from the same sysplex, another sysplex in the same network, or from a different sysplex. If an LU alias is predefined, a dynamic LU alias is not generated.

When to use predefined LU alias

Use predefined LU alias where you do not have dynamic LU alias enabled or where you want to override dynamic LU aliases.

- Dynamic LU alias is in operation in a CICS region and your terminals or workstations are explicitly defined on CICS terminal resource definitions with explicit terminal identifiers. In this case, you use predefined LU aliases to override the generation of dynamic LU aliases, which CICS fails to match with any installed resource definition.
- Dynamic LU alias is not in operation in a CICS region, to avoid any conflict with duplicate network names.

Cross-network resources that require predefined LU alias

If the following SNA cross-network resources are to be connected to a CICS region that is defined to SNA, with LUAPFX specified on its APPL statement, they must each have a CDRSC LUALIAS=netname entry.

- CICS RDO-defined terminals connected from another network, including SNA LUs that cannot be autoinstalled:
 - Pipeline terminals
 - Automatic teller machines (3614 and 3624)
 - Devices for which CICS does not receive logons, such as printers.
- LUTYPE 6.2 synclevel 1 connections that might be bound using limited resources.

Like other LUTYPE 6.2 connections, limited resource connections release their dynamic LU alias when CDRSCTI ends after the last session is unbound. However, these sessions are unbound whenever they are not in use, and, if they rebind after the dynamic LU alias is released, CICS installs another connection, potentially with a different LU alias.

- CICS RDO-defined work stations (LUTYPE 6.2 synclevel 1 connections) connected from another network.
- Resources that require an LU name in a RACF profile definition or resources for which prior knowledge of the LU name is required.

Choosing an LUAPFX value

When you choose an LUAPFX value, consider the scope of this parameter in the CICSplex, and also consider its scope in the sysplex in which your CICS regions operate.

A predefined LUALIAS name is supplied to CICS for cross-domain and cross-network resources. All the CICS regions in an MVS image share the same SNA and are in the same domain. A CICS region in a different MVS image uses a different SNA and is thus in a different domain. Resources coming from one SNA to another, but which share the name NETID, are cross-domain resources

A dynamic LUALIAS name is supplied to CICS only for cross-network resources. A resource is a cross-network resource if it has a different network id. SNA ensures that all the dynamic LUALIAS names assigned in one MVS image are unique. However, CICS requires network names to be unique across MVS images so that there are no network name clashes in AORs.

It is important that all CICS regions across all connected networks use unique APPLIDs. This requirement is true whether or not dynamic LUALIASs are used; it is more important with dynamic LUALIASs.

To ensure that all SNA resources in a CICSplex have unique network names, use the LUAPFX prefix:

- Specify LUAPFX on terminal-owning regions (TORs) only.
- Use the same LUAPFX value for all the CICS TORs in the same MVS image (that is, for all the TORs that are connected to the same SNA), but ensure the LUAPFX is different in each MVS image in the sysplex..

If the LUAPFX values are not the same throughout an MVS image, you risk one resource having two different network names in the CICS regions in that image.

If the LUAPFX values are not unique to each MVS image in the sysplex, you risk two resources attempting to install in a TOR with the same dynamic LUALIAS, or having two resources with the same network name in an AOR.

To ensure the uniqueness of the LU prefix in each MVS, use model APPL definitions, and in these use an MVS system symbol (&SYSCONE) as suggested in the [z/OS Communications Server: SNA Resource Definition Reference](#).

If you use SNA generic resources and your CICS TORs are spread across different MVS images, be aware that if a resource with a dynamically allocated LU alias logs off and then logs on again, and SNA switches the resource to an SNA in another MVS image, a different LUALIAS is assigned because of the different LUAPFX value.

- Avoid using an LUAPFX value that corresponds to the first two characters of CICS RDO-defined terminal names or connection names installed in the CICSplex.

Using LU aliases

Factors to consider when you are planning to use SNA LU aliases with CDRSC resources.

Predictable TERMIDs

If your autoinstalled terminal resources must have a predictable and reproducible TERMID for such things as temporary storage queue names and START requests, you can modify your autoinstall user-replaceable module (URM) to select a reproducible TERMID from the network qualified name, NQNAME, supplied in the CINIT or the BIND.

The sample autoinstall URM contains an example of such code (commented-out), which extracts the network qualified name from the CINIT and BIND. The example illustrates how to create a TERMID from the last nonblank character of the NETID and the last 3 nonblank characters of the *real* network name (NETNAME).

MVS workload management

If your MVS workload policies specify LU name classifications, remove the LU name for any cross-network resources that are autoinstalled by CICS.

Recovery and persistent sessions support

Resources for which CICS uses any SNA LU alias, predefined or dynamic, and which come from a different network are not cataloged by a CICS region that is not using a persistent session. Therefore, the terminal sessions for the resources cannot be recovered during an emergency restart.

Resources for which CICS uses any SNA LU alias, predefined or dynamic, and which come from a different network are catalogued if CICS is using persistent sessions, so CICS can restore resource terminal session information from the CICS catalog pending recovery of the session from SNA. However, if the resource does not persist, the resource is deleted during an emergency restart.

This action is necessary because SNA might have been restarted, causing dynamic LU aliases to be reissued to different sessions. CICS cannot recognize if SNA has been restarted and CICS cannot differentiate between a predefined and a dynamic LU alias.

CLSDST PASS

If you ISSUE PASS (CLSDST PASS) for a terminal that uses a dynamic LU alias to pass control to another CICS region in another MVS image, the resource is known by a different network name in the receiving CICS. This condition is true if the APPL statement of only one or both the CICS regions specify LUAPFX to activate dynamic LU alias.

Generic resources

If a number of generic resource TORs are in two different MVS images, a terminal or work station that logs on to one image is assigned a different network name if it logs off and logs on to a TOR in another image.

FEPI

FEPI front end systems are not supported by SNA LU alias.

Defining SNA requirements for CICSplex SM

You require ACF/SNA definitions to identify each CMAS used by CICSplex SM. You must create z/OS Communications Server application definitions and, optionally, cross-domain resource management definitions. There are a number of steps to perform to create z/OS Communications Server application definitions and cross-domain resource management definitions for a CMAS.

1. Before you perform these steps, specify the sizes of the SNA buffers.

- For the SNA-to-NCP connection, specify:
 - MAXDATA ≥ 4096
- For the NCP-to-SNA connection, specify:
 - MAXBFRU * IOBUF ≥ 4096

- MAXBFRU * UNITSZ ≥ 4096
- For the NCP-to-NCP connection, specify
 - TRANSFR * BFRS = RUSIZE ≥ 4096

Specify the size as 36 bytes smaller than the smallest MAXDATA value in any NCP through which the link might pass. The 36 bytes allow for the headers that are required for SNA. For more information about the requirements for the SNA-to-NCP connection, refer to [z/OS Communications Server: SNA Resource Definition Reference](#) for your level of SNA. For more information about the requirements for the NCP-to-SNA and the NCP-to-NCP connections, refer to [z/OS Communications Server: SNA Resource Definition Reference](#) for your level of NCP.

If you require help determining or modifying your z/OS Communications Server buffer specifications, discuss with the z/OS Communications Server system programmer at your enterprise.

2. Depending on your SNA conventions, you might have to modify the procedures that are described in this section:
 - Change references to the SYS1.VTAMLST library if you do not keep your definitions in the default z/OS Communications Server list.
 - Modify the APPL and CDRSC statements if you want to add these statements to existing members, rather than create new ones.
3. Use the sample SNA APPL statements in the SEYUDEF library members, EYUDVTIA and EYUDVTIB. You can modify these samples to fit your requirements.

Step 1: Creating a z/OS Communications Server application definition for a CMAS

To establish a z/OS Communications Server for SNA application definition for a CMAS, either create a new member (*major node*) or access an existing member in the SYS1.VTAMLST library.

Add the following APPL statement to the member:

	VBUILD TYPE=APPL	
name	APPL ACBNAME=acbname, AUTH=(VPACE, ACQ, SPO, PASS),	x
	EAS=10, PARSESS=YES, SONSCIP=YES, APPC=NO,	x
	VPACING=number	

where:

name

A 1-character to 8-character unique name.

acbname

The node name of this CMAS. This name must be unique in the domain. If you omit this parameter, the name of the SNA APPL statement is used.

vpacing

The maximum number of normal-flow requests that another logical unit can send on an intersystem session before waiting to receive a pacing response. Start with a value of five.

For example, to create a z/OS Communications Server application definition for the CMAS on SYSA, you might create a member APCMAS1 in the SYS1.VTAMLST library that contains the APPL statement:

	VBUILD TYPE=APPL	
CMS1	APPL ACBNAME=CMS1, AUTH=(VPACE, ACQ, SPO, PASS),	x
	EAS=10, PARSESS=YES, SONSCIP=YES, APPC=NO,	x
	VPACING=5	

You require the same type of definition for each CMAS you use.

Step 2: Defining cross-domain resources for a CMAS

Define cross-domain resources (CDRSCs) when a CMAS that is to communicate with another CMAS cannot take advantage of adjacent CDRSCs or you want to minimize the effort involved in using adjacent CDRSCs.

To establish a CDRSC definition, you must either create a new member or access an existing member in the SYS1.VTAMLST library. In the new or existing member, specify the following CDRSC statement for each CMAS that you want to communicate with:

```
name          VBUILD TYPE=CDRSC
              CDRSC CDRM=cdrm
```

where:

name

The name you assigned to a CMAS in Step 1.

cdrm

The name of the MVS image previously identified as the cross-domain resource manager (CDRM).

For example, to allow the CMAS on SYSA to communicate with the CMASs on SYSB and SYSC, you might create the member CDRCMS1 in the SYS1.VTAMLST library, that contains the CDRSC statements:

```
CMS2          VBUILD TYPE=CDRSC
              CDRSC CDRM=VTAMB
CMS3          CDRSC CDRM=VTAMC
```

where VTAMB and VTAMC are the cross-domain resource manager names that are assigned to SYSB and SYSC respectively.

You also require the same types of definitions for the CMASs on SYSB and SYSC. That is, for the CMAS on SYSB, you might create a member CDRCMS2, that contains the CDRSC statements:

```
CMS1          VBUILD TYPE=CDRSC
              CDRSC CDRM=VTAMA
CMS3          CDRSC CDRM=VTAMC
```

Note: VTAM is now z/OS Communications Server.

Step 3: Updating the configuration list for a CMAS

If, in Step 1 or 2, you created new members in the SYS1.VTAMLST library, you must update the SNA configuration list for each MVS image. When SNA starts, it automatically activates the new members.

To make these updates, add the new member names to the end of the configuration list in the appropriate ATCCONxx member of the SYS1.VTAMLST library. To find the suffix of the ATCCONxx member:

- Get the suffix of the COMMNDxx member from the CMD= parameter in the IEASYSxx member in SYS1.PARMLIB.
- Get the suffix of the ATCSTRxx member from the LIST= parameter on the command that is used to start SNA in the COMMNDxx member in SYS1.PARMLIB. If you do not start SNA from the COMMNDxx member, get the suffix from the LIST= parameter of the command that you use to start SNA.
- Get the suffix of the ATCCONxx member from the CONFIG= parameter in the ATCSTRxx member in SYS1.VTAMLST.

As an illustration, the examples shown in Steps 1 and 2 assume that the members APCMAS1 and CDRCMS1 exist. To add these members to the end of the configuration list in ATCCONxx, specify:

```
APCMAS1,
CDRCMS1                                     x
```

If you added the CMAS and cross-domain definitions to existing members, ATCCONxx already contains these member names.

Step 4: Activating the major nodes for a CMAS

You can activate the definitions that are created in Steps 1 and 2 either by restarting the z/OS Communications Server for each system or by manually activating the definitions.

To manually activate a major node, you can issue the following commands, where name identifies a major mode that was created or modified in Steps 1 and 2:

- Deactivate the major node if it is currently active by issuing this command:

```
VARY NET,INACT,ID=name
```

- Activate or reactivate the major node by issuing this command:

```
VARY NET,ACT,ID=name
```

To ensure that the major node has been activated, issue this command:

```
D NET,ID=name
```

For example, to activate the member APCMAS1 and then ensure that it has been activated, issue these commands:

```
VARY NET,INACT,ID=APCMAS1  
VARY NET,ACT,ID=APCMAS1  
D NET,ID=APCMAS1
```

Perform the preceding steps for each CMAS you are using.

Installing CICS modules in the MVS link pack area

How to define the CICS LPA library to your MVS, and how to install and use the CICS modules.

Preparing to install CICS modules in the MVS link pack area

Before you install modules in the MVS link pack area, you need to understand the benefits, the scope, service level implications, and the modules required in the MVS link pack area.

Benefits of using the MVS link pack area

There are several benefits of placing code in the MVS link pack area.

- You protect the code from possible corruption by user applications. Because the MVS link pack area is in protected storage, it is virtually impossible to modify the contents of these programs.
- You can improve the performance, and reduce the demand for real storage, if you use the MVS link pack area for program modules. If more than one copy of the same release of CICS is running in multiple address spaces of the same processor, each address space requires access to the CICS nucleus modules. These modules can either be loaded into each of the address spaces or shared in the MVS link pack area. If they are shared in the MVS link pack area, the working set and, therefore, the demand for real storage (paging) can be reduced.
- You can decrease the storage requirement in the private area by careful allocation of the unused storage in the MVS link pack area created by rounding to the next segment.

If you know the amount of space that you require in the LPA, and from that the total size of the MVS common area above the CICS private storage, you can determine the 1 MB segment on which the boundary between the two areas lies. This calculation might indicate some space in the MVS common area is left unused, which you can use for CICS LPA-eligible modules. By moving more modules from CICS private storage to the LPA, you decrease the space that is required for modules in CICS private storage.

What is meant by the MVS link pack area?

The MVS link pack area comprises several areas, both above and below 16 MB. In these topics, the term MVS link pack area refers to the pageable link pack areas above and below 16 MB where modules that are used from the MVS link pack area are normally installed.

The MVS link pack area has both pageable and fixed parts. Although you can install CICS modules into the fixed parts, use the pageable areas for performance reasons.

The term *LPA* specifically refers to the MVS link pack area below 16 MB, and the term *ELPA* specifically refers to the area above 16 MB. A module that is link-edited with the RMODE(ANY) attribute is loaded into the ELPA.

If you install a module into the LPA or ELPA, that module cannot be retrieved from the MVS link pack area until you re-IPL your MVS with CLPA specified. To use the new module and avoid an IPL, you can use the MVS dynamic LPA in one of the following ways:

- With the SETPROG LPA command
- With an LPA statement in a PROGxx member of PARMLIB, which specifies the SET PROG=xx command

Mismatch of service levels

If you use modules with mismatching service levels, you can cause unpredictable results. To be safe, do not use the LPA version of a module if it differs from the version in the CICS libraries that you are using.

Except for *DFH99SVC* and except for when you use the *DFHCSVCU* utility program to dynamically update the CICS SVC, do not use dynamic LPA with the 8 CICS modules that must be installed in the MVS link pack area. For details about the *DFHCSVCU* utility program, see, [“Running the DFHCSVCJ job” on page 51](#).

Load modules used from the LPA might be at a lower service level than the rest of your CICS region in any of the following circumstances:

- You are running CICS from libraries that belong to a target zone currently at a higher service level than the LPA zone.
- You have applied service to the LPA zone since the last IPL of MVS.
- You are not using the MLPA to replace service-updated load modules, but have applied service to the LPA zone since the last IPL of MVS for which CLPA (create link pack area) was specified.

Thus, if you have applied service to a load module in your CICS libraries, you must also apply the service to the LPA version of the module, if one exists, so that the MVS link pack area always contains tested load modules.

Use the SMP/E RESTORE function to remove the USERMOD before the LPA zone is updated or copied. Then apply the USERMOD again.

If you have used a copy of the CICS-supplied USERMODs to install modules into the MVS link pack area, and the original USERMOD is serviced, you can reflect the changes in your version of the USERMOD.

Although it is possible for CICS PTFs to make changes to the CICS modules which must reside in the MVS link pack area, in reality it is a fairly rare event. The modules have to be downwardly compatible with lower releases, and applies to any code changes made to them by PTFs, as well as to the code at base level.

Which modules must be installed in the MVS link pack area?

The CICS modules that must be in the MVS link pack area are listed.

Table 8. CICS modules required in the LPA			
Module	Description	When required in LPA	See notes after this table
DFHCSVC	CICS Type 3 SVC	Always	<u>1</u> , <u>2</u> , <u>4</u> , and <u>6</u>

Table 8. CICS modules required in the LPA (continued)

Module	Description	When required in LPA	See notes after this table
DFHDSPEX	CICS postexit stub	Always	1 , 3 , and 5
DFHDUMPX	SDUMPX IEASDUMP QUERY exit	Always	1 and 3
DFHIRP	Interregion communication program	To use MRO, EXCI, or the console message-handling facility	1 , 2 , and 6
DFHSSSEN	Subsystem interface end-of-memory and end-of-task clean up routine	To use the console message-handling facility	1 , 3 , and 6
DFHSSGC	Subsystem generic connect module	To use the console message-handling facility	3 and 7
DFHSSWT	Subsystem interface WTO router	To use the console message-handling facility	3 and 7
DFH99SVC	Dynamic allocation - SVC services	Always	1 and 3

1. The module can be used only from the MVS link pack area and you must install it there before CICS can be started.
2. You must always install the latest service level of the modules DFHCSVC and DFHIRP (if required). The version of the modules supplied with the current release is downward-compatible with earlier releases of CICS. It works correctly with CICS regions running earlier releases. Therefore, if you are running different releases of CICS on the same MVS image, use the latest version of the modules.
3. There is no requirement to use the latest version of this module. The version supplied with the current release is the same as in previous releases.
4. You must define the DFHCSVC module to MVS as a Type 3 SVC. The default SVC number is 216. If you use a non-default SVC number, you must define it to CICS on the CICSSVC system initialization parameter.

Moving DFHCSVC into the MVS link pack area:

Do not use the link editor to install the CICS SVC module into a library in the MVS link pack area. To copy or move the module from the *hlq.SDFHAUTH* library to the nominated library in the MVS link pack area, use either a suitable copy utility program, such as IEBCOPY, or an SMP/E USERMOD with ++MOVE statements.

5. The DFHDSPEX module must be in the MVS link pack area for integrity reasons, but the postexit routine, DFHDSAUT, can be either in the MVS link pack area or in the CICS address space. This choice enables you to use different levels of the DFHDSAUT module in different CICS regions running in the same MVS image, because the DFHDSAUT module might not be compatible between CICS versions.
6. To communicate by MRO, all CICS regions in the same MVS image must use the latest level of the modules DFHCSVC and DFHIRP in the MVS link pack area.

If a region detects that DFHIRP is at a lower level when it attempts to open interregion communication, it issues message DFHIR3799 and interregion communication fails to open.

7. To use console message-handling, these modules must be either in the MVS link pack area or in an APF-authorized library in the MVS linklist.

If an LPA module is only ever referred to by name, then refreshing it in the LPA is valid. That is, if each time it is used, a new reference to it is obtained by using an MVS LOAD macro. However, most references to LPA modules do not work this way. The address of an LPA module could be saved into a control block after the initial LOAD of the module. The implication is that this address remains unchanged without a CLPA IPL. If dynamic LPA is used to pick up a change to an LPA module, then the address saved in the control block is

no longer valid. Of the 8 CICS modules which must be installed in the LPA, only DFH99SVC is eligible to be refreshed in this way.

Which modules can be in the MVS link pack area?

A number of CICS modules and user application program modules are available from the MVS link pack area.

CICS modules

The following CICS modules can be installed in the LPA:

- Modules that can reside above 16 MB (for example, the CICS message table, DFHMGT). You can also install these modules in the extended link pack area (ELPA).
- Modules that are optionally installed in the MVS link pack area; that is, a module that is not required in the MVS link pack area can be used only by the release of CICS to which it relates.
- Modules that are eligible to be used from the MVS link pack area are listed in the CICS-supplied USERMODs. The USERMOD member for base CICS modules is DFH\$UMOD, which is in the *hlq.SDFHSAMP* library. Refer to DFH\$UMOD for a list of these modules, and to the details in [“LPA-required and LPA-eligible modules” on page 101](#), to help you select the CICS modules that you want to install in the MVS link pack area.

User application programs

You can use user application programs from the MVS link pack area if they are read-only and meet one of the following criteria:

- They are written in COBOL, do not overwrite WORKING STORAGE, and are compiled using VS COBOL II, or a later version. The CICS translator generates a CBL statement with the required compiler options.
- They are written in PL/I (so they do not overwrite STATIC storage) and are compiled using one of the versions of PL/I for z/OS listed in [Changes to CICS support for application programming languages](#). The CICS translator inserts the required REENTRANT option into the PROCEDURE statement.
- They are written in C/370, compiled with the RENT option, and link-edited with the RENT option.
- They are written in assembler language, assembled with the RENT option, and link-edited with the RENT and REFR options.

Command-level user application programs compiled using a Language Environment-conforming compiler, or written in assembler language or C/370, can be loaded above 16 MB. For information about installing application programs, see [Installing application programs](#).

A read-only module that can be above 16 MB is also eligible for the ELPA.

LPA-required and LPA-eligible modules

The LPA-required modules, that is, modules that must be installed in the MVS link pack area, as supplied in *hlq.SDFHLPA*, are listed. The LPA-eligible modules that require associated system initialization parameters, or that are installed in the LPA below 16 MB, are also listed.

For a complete list of LPA-eligible modules, see the CICS-supplied sample DFH\$UMOD. For further information about this sample, see [“Installing CICS modules in the LPA” on page 112](#).

To find the size of each module, check the load library directory information.

LPA-required modules

All LPA-required modules are loaded into the part of the MVS link pack area that is above 16 MB (ELPA).

Table 9. LPA-required modules, supplied in hlq.SDFHLPA		
Name	Description	Notes
DFHCSVC	CICS SVC startup	CICSSVC “1” on page 102 “2” on page 102
DFHDSPEX	DS domain - MVS POST exit stub	“2” on page 102
DFHDUMPX	SDUMPX IEASDUMP QUERY exit	“2” on page 102
DFHIRP	Interregion communication program	“1” on page 102 “2” on page 102 “3” on page 102
DFHSSSEN	Subsystem interface end-of-memory / end-of-task clean up routine	“1” on page 102 “2” on page 102 “3” on page 102
DFHSSGC	Subsystem interface generic connect	“2” on page 102 “4” on page 103
DFHSSWT	Subsystem interface WTO router	“2” on page 102 “4” on page 103
DFH99SVC	Dyn alloc - SVC services	“1” on page 102 “2” on page 102

Notes:

1. You must always install the latest service level of the CICS SVC module, DFHCSVC. Install the DFHCSVC module into the MVS link pack area before you run the CICS installation verification procedures.

You must define the DFHCSVC module in an IEASVCxx member of the SYS1.PARMLIB library with SVC Parm statements. You select the required IEASVCxx member by coding the SVC parameter (SVC=xx) in a SYS1.PARMLIB member (IEASYSy), which you use to IPL your MVS.

You can run several CICS regions, at different release levels, in the same MVS image. Each region uses its own version of the DFHCSVC module. If some regions use MRO, all regions that use MRO must use the latest DFHCSVC module and the latest DFHIRP module.

If some regions use the DFHCSVC module, and you allocate to the SVC a number different from the SVC number that is used by the regions, you must generate a new version of the DFHCRC program on the regions.

For more information about the DFHCSVC module, see [“Program Directories” on page 2](#).

2. All LPA-required modules are compatible with earlier releases of CICS. If you are running earlier releases of CICS, you must ensure that the correct version is installed in the LPA. The module must be in the LPA for integrity reasons, but the post exit routine itself can stay in the LPA or in the CICS address space. You can then use different versions of the DFHDSAUT module in different CICS regions that run in the same MVS image, because the DFHDSAUT module might not be compatible with all releases.
3. The DFHIRP module must be in the MVS link pack area only if you are using MRO, CICS shared database, or the console message-handling facility. If you install the DFHIRP module in the MVS link pack area, you must also install DFHSSSEN if you are using the console message-handling facility.

You must always install the latest service level of the DFHIRP (if required) and DFHSSSEN.

If you are running CICS with MRO at different release levels, all regions in the same MVS image must use the latest DFHIRP module.

4. To use the console message formatting facility of the MVS subsystem interface, you must install the modules DFHSSGC and DFHSSWT either in the MVS link pack area or in an APF-authorized library in the MVS link list. These modules are used by the subsystem interface and not directly by CICS. Therefore, the use of these modules from the MVS link pack area is not controlled by CICS parameters or options.

For information about enabling the console message-formatting facility, and about the other modules it requires, see [“Modules required to use the console message-handling facilities”](#) on page 83.

LPA-eligible modules

Table 10 on page 103 and Table 11 on page 111 shows the LPA-eligible modules that require associated system initialization parameters, or that are affected by an option in the resource definition of the program.

- If an LPA-eligible module requires an associated system initialization parameter, you must specify this parameter to use the function that is associated with that module. For more information about the system initialization parameters, see [System initialization parameter descriptions and summary](#).
- Some LPA-eligible modules are affected by the USELPACOPY option of the resource definition of the program. If USELPACOPY=NO, you do not need to include that module in the MVS link pack area. For more information about the USELPACOPY option, see [PROGRAM attributes](#).

Table 10 on page 103 shows the LPA-eligible modules that require system initialization parameters or are affected by USELPACOPY and that are loaded into the part of the MVS link pack area above the 16 MB line (ELPA).

Table 11 on page 111 shows the LPA-eligible modules that require system initialization parameters or are affected by USELPACOPY and that are loaded into the link pack area below the 16MB line (LPA).

Table 10. LPA-eligible modules that are loaded into the ELPA that require system initialization parameters or are affected by USELPACOPY		
Name	Description	System initialization parameter or note
DFHAIIN	Terminal Autoinstall - initialization	AIEXIT
DFHAIIQ	Terminal Autoinstall - locate/unlock/inquire/browse	AIEXIT
DFHAIRP	Terminal Autoinstall - recovery	AIEXIT
DFHAITM	Terminal Autoinstall - add replace/delete	AIEXIT
DFHALP	Terminal allocation	AIEXIT
DFHAPHN	WLM Health Notify Gate	WLMHEALTH=(xx,yy)
DFHAPTБ	AP table read handling	
DFHBRAI	3270 Bridge Autoinstall	AIBRIDGE=YES
DFHCCNV	Codepage conversion	“1” on page 110
DFHCEGN	Goodnight transaction stub	“1” on page 110
DFHCESD	Shutdown transaction	“1” on page 110
DFHCHS	CICS OS/2 mirror program	“1” on page 110
DFHCIS4	ESI over IPIC Message processing	“1” on page 110
DFHCMAC	ME domain - CICS messages and codes transaction (CMAC)	“1” on page 110
DFHCNV	Codepage conversion table	“1” on page 110
DFHCRNP	Interregion connection manager	“1” on page 110

Table 10. LPA-eligible modules that are loaded into the ELPA that require system initialization parameters or are affected by USELPACOPY (continued)

Name	Description	System initialization parameter or note
DFHCRQ	ATI purge program	“1” on page 110
DFHCRR	Interregion session recovery program	“1” on page 110
DFHCRS	Remote scheduler program	“1” on page 110
DFHCRSP	CICS IRC startup module	“1” on page 110
DFHCRT	Transaction routing relay program for APPC devices	“1” on page 110
DFHDBAT	CICS DBCTL adapter/transformer	“1” on page 110
DFHDBCT	CICS DBCTL control program	“1” on page 110
DFHDBTI	EXEC DLI LD table	“1” on page 110
DFHDIP	Data interchange program	DIP=YES
DFHDIPDY	Data interchange program (dummy)	DIP=NO
DFHDPWM0	Application Debugging profile - English Messages	“1” on page 110
DFHDPWM1	Application Debugging profile - German Messages	“1” on page 110
DFHDPWM2	Application Debugging profile - Spanish Messages	“1” on page 110
DFHDPWM3	Application Debugging profile - French Messages	“1” on page 110
DFHDPWM4	Application Debugging profile - Japanese Messages	“1” on page 110
DFHDPWM5	Application Debugging profile - Korean Messages	“1” on page 110
DFHDPWM6	Application Debugging profile - Chinese Messages	“1” on page 110
DFHDPWT0	Application Debugging profile - English Template	“1” on page 110
DFHDPWT1	Application Debugging profile - German Template	“1” on page 110
DFHDPWT2	Application Debugging profile - Spanish Template	“1” on page 110
DFHDPWT3	Application Debugging profile - French Template	“1” on page 110
DFHDPWT4	Application Debugging profile - Japanese Template	“1” on page 110
DFHDPWT5	Application Debugging profile - Korean Template	“1” on page 110
DFHDPWT6	Application Debugging profile - Chinese Template	“1” on page 110
DFHDSAUT	DS domain - authorized services	“2” on page 110
DFHDSBA\$	BMS data stream build (standard)	BMS=STANDARD
DFHDSB1\$	BMS data stream build (full)	BMS=FULL
DFHDSRP	Sample Distributed Dynamic Routing Program	“1” on page 110
DFHDYP	Dynamic routing program	DTRPGM=DFHDYP “1” on page 110
DFHD2CM0	CICS Db2 PLTPI Program	“1” on page 110
DFHD2CM1	CICS Db2 Command processor	“1” on page 110
DFHD2CM2	CICS Db2 Shutdown Quiesce Program	“1” on page 110

Table 10. LPA-eligible modules that are loaded into the ELPA that require system initialization parameters or are affected by USELPACOPY (continued)

Name	Description	System initialization parameter or note
DFHD2CM3	CICS Db2 PLTPI Program	“1” on page 110
DFHD2EDF	CICS Db2 EDF Processor	“1” on page 110
DFHD2EX1	CICS Db2 Task Related User Exit Program	“1” on page 110
DFHD2EX2	CICS Db2 Service Task Program	“1” on page 110
DFHD2INI	CICS Db2 Initparm Processor	“1” on page 110
DFHEBU	EXEC FMH construction	ISC=YES/xx
DFHECBAM	Business Transaction Services Program	“1” on page 110
DFHECID	CECI service program	“1” on page 110
DFHECIP	Command interpreter (CECI) program	“1” on page 110
DFHECSP	Command syntax check (CECS) program	“1” on page 110
DFHEDAD	RDO (CEDA) service program	“1” on page 110
DFHEDAP	RDO (CEDA) program	“1” on page 110
DFHEDFBR	Temporary-storage browse transaction, CEBR	“1” on page 110
DFHEDFD	EDF display program	“1” on page 110
DFHEDFE	EDF attach error handler	“1” on page 110
DFHEDFP	EDF control program	“1” on page 110
DFHEDFR	EDF response table	“1” on page 110
DFHEDFX	EDF task switch program	“1” on page 110
DFHEDI	EXEC interface for data interchange	DIP=YES
DFHEDP	EXEC DLI command stub	“1” on page 110
DFHEGL	EXEC interface for unmapped LU6.2 commands	VTAM=YES
DFHEIGDS	Translator table (GDS commands)	“1” on page 110
DFHEIPSE	EXEC interface for perform security	SEC=YES
DFHEITMT	Command language table for CEMT	“1” on page 110
DFHEITOT	Command language table for CEOT	“1” on page 110
DFHEITST	CEST language definition table	“1” on page 110
DFHEITSZ	EXEC CICS language definition table	“1” on page 110
DFHEMS	EXEC interface for BMS	BMS=STANDARD or FULL
DFHEMTA	Programmable interface to Main terminal program	“1” on page 110
DFHEMTD	Main terminal (CEMT) service program	“1” on page 110
DFHEMTP	Main terminal (CEMT) program	“1” on page 110
DFHEOTP	CEOT service program	“1” on page 110

Table 10. LPA-eligible modules that are loaded into the ELPA that require system initialization parameters or are affected by USELPACOPY (continued)

Name	Description	System initialization parameter or note
DFHEPS	System spooling interface stub	SPOOL=YES
DFHESTP	CEST service program	“1” on page 110
DFHETR	EXEC interface for trace control	“3” on page 110
DFHETRX	EXEC interface for enter tracenum, monitor	USERTR
DFHGMM	z/OS Communications Server LU startup message	“1” on page 110
DFHIEP	IP ECI Listener Program	“1” on page 110
DFHINDAP	Indoubt tool	“1” on page 110
DFHINDT	Indoubt tool	“1” on page 110
DFHINTRU	Indoubt tool task-related user exit	“1” on page 110
DFHISAIP	IPCONN Autoinstall Program	“1” on page 110
DFHISCOP	IPCONN - acquire and release program	“1” on page 110
DFHISEMP	ISC over TCPIP error handler	“1” on page 110
DFHISLQP	ISC over TCPIP - local queue processing	“1” on page 110
DFHISREU	IS resync UOWs	“1” on page 110
DFHISREX	IS resync XIDs	“1” on page 110
DFHISRRP	ISC over TCPIP - receiver	“1” on page 110
DFHISRSP	ISC over TCPIP - remote scheduler	“1” on page 110
DFHISP	Intersystem communication program	ISC=YES
DFHLDMAP	Loader map program	“1” on page 110
DFHLDMHF	Loader map HFS output program	“1” on page 110
DFHLDMHS	Loader map spool output program	“1” on page 110
DFHMCPA\$	BMS mapping control program (standard)	BMS=STANDARD
DFHMCPE\$	BMS mapping control program (minimum)	BMS=MINIMUM
DFHMCPI\$	BMS mapping control program (full)	BMS=FULL
DFHMCX	BMS fast path module	BMS
DFHMCY	Process MAPPINGEV Requests	BMS
DFHMIRS	Mirror program	ISC=YES “1” on page 110
DFHMLBST	Schema to Language structure bootstrap program	ISC=YES “1” on page 110
DFHML1	BMS LU1 printer mapping program	BMS
DFHMQBAS	CICS MQ interface - base panel program	“1” on page 110
DFHMQBP0	CICS MQ bridge - request handler	“1” on page 110
DFHMQBP1	CICS MQ bridge - DPL abend handler	“1” on page 110

Table 10. LPA-eligible modules that are loaded into the ELPA that require system initialization parameters or are affected by USELPACOPY (continued)

Name	Description	System initialization parameter or note
DFHMQBRO	CICS MQ bridge - monitor main program	“1” on page 110
DFHMQCOD	CICS MQ PLT program to start connection by using INITPARM	“1” on page 110
DFHMQCON	CICS MQ Back-end module that issues a connection request to the IBM MQ subsystem	“1” on page 110
DFHMQCTL	CICS MQ First level transaction for operations. This program handles text mode operation only	“1” on page 110
DFHMQDCI	CICS MQ Data conversion exit for inbound data to the bridge	“1” on page 110
DFHMQDCO	CICS MQ Data conversion exit for outbound data from the bridge	“1” on page 110
DFHMQDIS	CICS MQ Back-end module that services display requests	“1” on page 110
DFHMQDSC	CICS MQ Back-end module that issues a disconnect request to the IBM MQ subsystem	“1” on page 110
DFHMQDSL	CICS MQ Second-level transaction for DISPLAY	“1” on page 110
DFHMQIG	CICS MQ inquire group	“1” on page 110
DFHMQMCM	CICS MQ async consume driver	“1” on page 110
DFHMQMON	CICS MQ Monitor program for pending events	“1” on page 110
DFHMQPLT	CICS MQ Sample CONNECT program in phase 2 PLT execution	“1” on page 110
DFHMQPOP	CICS MQ BMS program that handles all pull-down maps	“1” on page 110
DFHMQPRM	CICS MQ program - Retrieves all default adapter parameters	“1” on page 110
DFHMQPUL	CICS MQ BMS program that handles all pull-down maps	“1” on page 110
DFHMQQCN	CICS MQ Second-level transaction for CONNECT	“1” on page 110
DFHMQRET	CICS MQ BMS program that handles the final screen, including scrolling	“1” on page 110
DFHMQRS	CICS MQ Back-end module to service reset (change) requests	“1” on page 110
DFHMQSSQ	CICS MQ Back-end module to service start and stop CKTI requests	“1” on page 110
DFHMQTRU	CICS MQ task-related user exit program for IBM MQ	“1” on page 110
DFHMQTSK	CICS MQ Trigger monitor (CKTI transaction)	“1” on page 110
DFHMSP	Message switching program	“1” on page 110
DFHMXP	Local queuing shipper	“1” on page 110
DFHM32A\$	BMS 3270 mapping (standard)	BMS=STANDARD
DFHM321\$	BMS 3270 mapping (full)	BMS=FULL
DFHPBPA\$	BMS page and text build (standard)	BMS=STANDARD
DFHPBP1\$	BMS page and text build (full)	BMS=FULL
DFHPGADX	Program autoinstall exit - Assembler	“1” on page 110
DFHPHP	Partition handling program	BMS

Table 10. LPA-eligible modules that are loaded into the ELPA that require system initialization parameters or are affected by USELPACOPY (continued)

Name	Description	System initialization parameter or note
DFHPIAP	Pipeline - stub program	“1” on page 110
DFHPIDSH	Pipeline - HTTP inbound router	“1” on page 110
DFHPIDSQ	Pipeline - mq dispatcher	“1” on page 110
DFHPIITL	Pipeline - resolution program	“1” on page 110
DFHPILSQ	Pipeline - mq listener	“1” on page 110
DFHPIPA	Soap envelope SAX parsing	“1” on page 110
DFHPIRI	Pipeline - exec layer remote stub	“1” on page 110
DFHPIRS	Pipeline - registration services	“1” on page 110
DFHPIRT	Pipeline - outbound router	“1” on page 110
DFHPISN1	Soap 1.1 node program	“1” on page 110
DFHPISN2	Soap 1.2 node program	“1” on page 110
DFHPITP	Pipeline - exec layer program	“1” on page 110
DFHPITQ1	Pipeline - mq interface	“1” on page 110
DFHPIXC	Soap envelope sax parsing	“1” on page 110
DFHPSJP	Spool JCL parser	SPOOL=YES “4” on page 111
DFHPSP	System spooling interface program	SPOOL=YES
DFHPSSVC	System spooling interface, retrieve a data set name	SPOOL=YES
DFHQRY	Query transaction	“1” on page 110
DFHRLMF	RL Manifest XSDBIND file	“1” on page 110
DFHRLRA\$	BMS route list resolution (standard)	BMS=STANDARD
DFHRLRP	RL resolution program	“1” on page 110
DFHRLSC	RL SCDL schema bootstrap	“1” on page 110
DFHRLVC	RL vendor register/deregister	“1” on page 110
DFHRLR1\$	BMS route list resolution (full)	BMS=FULL
DFHRMSY	Resource manager resync program	“1” on page 110
DFHRPAL	ONC RPC feature alias list	“1” on page 110
DFHRTC	CRTE cancel command processor	“1” on page 110
DFHRTE	Transaction routing program	“1” on page 110
DFHRZJN	Remote join program	“1” on page 110
DFHSFP	Signoff program	“1” on page 110
DFHSHRRP	Scheduler services - request receiving program	“1” on page 110
DFHSHRSP	Scheduler services - request sending program	“1” on page 110

Table 10. LPA-eligible modules that are loaded into the ELPA that require system initialization parameters or are affected by USELPACOPY (continued)

Name	Description	System initialization parameter or note
DFHSJITL	SJ - Installation assist program	“1” on page 110
DFHSMTAB	CPSM LD table	“1” on page 110
DFHSNP	Signon program	“1” on page 110
DFHTPPA\$	BMS terminal page processor (standard)	BMS=STANDARD
DFHTPP1\$	BMS terminal page processor (full)	BMS=FULL
DFHTPQ	BMS terminal page clean up program	BMS “1” on page 110
DFHTPR	BMS terminal page retrieval program	BMS “1” on page 110
DFHTPS	BMS terminal page scheduling program	BMS “1” on page 110
DFHTRAO	TR domain - auxiliary trace output	AUXTR=ON
DFHTSP	Temporary-storage control program	TST=YES/xx
DFHWBAAX	WB - default analyzer program	“1” on page 110
DFHWBADX	WB - analyzer URM	“1” on page 110
DFHWBENV	WB - environment variables program	“1” on page 110
DFHWBERX	WB - error transaction program	“1” on page 110
DFHWBIMG	WB - binary image support	“1” on page 110
DFHWBPA	WB - parsing program	“1” on page 110
DFHWBUN	WB - unescaping program	“1” on page 110
DFHWSMS	DFHWSMS	XRF=YES/xx
DFHWSSON	CAVM state management signon request handler	XRF=YES/xx
DFHWTI	XRF takeover initiation program	XRF=YES/xx
DFHXFP	Online data transformation program	ISC=YES
DFHXFX	Optimized data transformation program	ISC=YES
DFHXRP	XRF request program	XRF=YES/xx
DFHXRSP	XRF surveillance program	XRF=YES/xx
DFHXSS	XS domain - supervisor request services	SEC=NO
DFHXSWM	XRF message manager for security manager	XRF=YES/xx
DFHXTP	Terminal sharing transformation program	ISC=YES
DFHZATA	Autoinstall program	“1” on page 110
DFHZATD	Autoinstall delete program	“1” on page 110
DFHZATDX	User-replaceable autoinstall exit	AIEXIT “1” on page 110
DFHZATDY	User-replaceable autoinstall exit with APPC	AIEXIT “1” on page 110
DFHZCA	z/OS Communications Server working set module	VTAM=YES
DFHZCB	z/OS Communications Server working set module	VTAM=YES

Table 10. LPA-eligible modules that are loaded into the ELPA that require system initialization parameters or are affected by USELPACOPY (continued)

Name	Description	System initialization parameter or note
DFHZCC	z/OS Communications Server working set module	VTAM=YES
DFHZCN1	CICS Client CCIN transaction	“1” on page 110
DFHZCP	Terminal management program	VTAM=YES
DFHZCT1	CICS Client CTIN transaction	“1” on page 110
DFHZCUT	Persistent verification signed-on-from list management program	VTAM=YES
DFHZCW	z/OS Communications Server nonworking set module	VTAM=YES
DFHZCX	LOCATE, ISC/IRC request	ISC=YES
DFHZCXR	Transaction routing module address list	ISC=YES
DFHZCY	z/OS Communications Server nonworking set module	VTAM=YES
DFHZCZ	z/OS Communications Server nonworking set module	VTAM=YES
DFHZGAI	APPC autoinstall - create APPC clones	AIEXIT
DFHZGBM	APPC manipulate bitmap	VTAM=YES
DFHZGCA	LU6.2 CNOS actioning	VTAM=YES
DFHZGCC	Catalog CNOS services	VTAM=YES
DFHZGCN	LU6.2 CNOS negotiation	VTAM=YES
DFHZGPR	z/OS Communications Server persistent sessions resource handler	VTAM=YES
DFHZHPRX	Authorized path SRB mode z/OS Communications Server EXECRPL	HPO=YES
DFHZLS1	LU6.2 CNOS request transaction program	VTAM=YES “1” on page 110
DFHZRSP	Resync send program	TCP=YES/xx “1” on page 110

Notes:

1. The program is used from the MVS link pack area only if you set the USELPACOPY option of its program resource definition to YES.
2. All LPA-required modules are compatible with earlier releases of CICS. If you are running earlier releases of CICS, you must ensure that the correct version is installed in the LPA. The module must be in the LPA for integrity reasons, but the post exit routine itself can stay in the LPA or in the CICS address space. You can then use different versions of the DFHDSAUT module in different CICS regions that run in the same MVS image, because the DFHDSAUT module might not be compatible with all releases.
3. You can set the system tracing status by coding appropriate system initialization parameters, and you can also set it dynamically by using the CETR transaction.

You can use the following system initialization parameters:

AUXTR

Activate auxiliary trace.

AUXTRSW

Define the auxiliary switch status.

GTFTR

Enable CICS to use MVS GTF tracing.

INTTR

Activate CICS internal tracing.

TRTABSZ

Specify the size of the internal trace table.

USERTR

Set the main user trace flag on or off.

For information about using CICS trace and controlling the tracing status with the CETR transaction, see [Setting trace destinations and tracing status](#).

4. In addition, the feature toggle for surrogate user checking for spool commands is enabled:

```
com.ibm.cics.spool.surrogate.check=true
```

For more information, see [Security for submitting a JCL job to the internal reader](#).

Table 11. LPA-eligible modules that are loaded into the LPA that require system initialization parameters or are affected by USELPACOPY

Name	Description	System initialization parameter or note
DFHFCBD	File control BDAM request processor	FCT=YES/xx
DFHFCU	File open utility program	“1” on page 111
DFHTRAOX	TR domain - auxiliary trace output	AUXTR=ON

Note:

1. The program is used from the MVS link pack area only if you set the USELPACOPY option of its program resource definition to YES.

Space requirements for CICS modules in the MVS link pack area

Allow enough space in the MVS link pack area for you to install those CICS modules that you intend using from there.

To find out how much space you require:

- Review the sizes of the modules that you want to install in the MVS link pack area, as given in the load library directory information.
- Review the module index of a system dump for the CICS region started with the system initialization parameter LPA=NO.
- Calculate the module sizes that are given for each module in the listing of modules that is provided by the IEHLIST utility program.

Remember also to allow space for any of your user application programs that you intend using from the MVS link pack area. The total space required depends on how the operating system packages the modules into the MVS link pack area.

When you have determined the space you require in the MVS link pack area, create a library with enough space and define it to your MVS. See [“Defining the CICS LPA library to your MVS” on page 112](#) for more information.

Defining the CICS LPA library to your MVS

CICS supplies the library *hlq.SDFHLPA*. This library contains the modules that must be in the LPA. You can also use this library to install other CICS modules or application programs that you want to use from the LPA.

You can give the *hlq.SDFHLPA* library your own index, but, if you do, you must specify the new index on the `LINDEX` parameter of the `DFHISTAR` job.

Add the full name of the *hlq.SDFHLPA* library to an `LPALSTxx` member of `SYS1.PARMLIB`. You must then IPL the system with the `CLPA` option to ensure that the library contents are loaded into the PLPA. Set the `CLPA` option by specifying it as an `IEASYSxx` member of `SYS1.PARMLIB` or in response to message `IEA101A`. Include `IEASYSxx` as the `SYSPARM` value in the `LOADxx` member.

RACF-protect the *hlq.SDFHLPA* library, to prevent unauthorized or accidental modification of this library. For information about protecting the CICS libraries, see the [CICS installation requirements for RACF](#).

You can install into the *hlq.SDFHLPA* library the CICS modules to be used from the MVS link pack area. See [“Installing CICS modules in the LPA” on page 112](#).

Installing CICS modules in the LPA

You can install modules into the MVS link pack area by using `SMP/E`.

In the context of the LPA, *install* means to move or copy a module into a suitable LPA library by using `SMP/E`, or by using a copying method that re-blocks the copied modules when the target data set has a smaller block size than the data set you copy from. For example, you can use the `COPYMOD` function of the `IEBCOPY` program.

Do not relink-edit the modules to put them into the LPA library. CICS modules are supplied with the necessary attributes such that MVS loads them automatically above the 16 MB line into the ELPA.

The following steps summarize the procedure to install modules in the CICS LPA library, and to ensure that `SMP/E` can continue to service them. Further information explains these steps in more detail.

1. Select those modules that you want to use from the MVS link pack area and specify them in the `SMP/E USERMOD` to be used to install the modules in the MVS link pack area.

For a list of CICS-supplied modules eligible for the MVS link pack area, see the CICS-supplied sample `DFH$UMOD`.

To install modules in the MVS link pack area, use one of the following methods:

- Use an `SMP/E USERMOD` that contains `++MOVE` statements for only the modules to be installed in the MVS link pack area.

CICS supplies an `SMP/E USERMOD` called `DFHUMOD` in member `DFH$UMOD` in the *hlq.SDFHSAMP* library. This `USERMOD` contains `++MOVE` statements for all CICS modules, in the *hlq.SDFHAUTH* and *hlq.SDFHLOAD* libraries, that are eligible for the MVS link pack area. The `USERMOD` also indicates whether each module is LPA- or ELPA-eligible. You can choose which of the modules to install in the LPA by creating your own version of the `USERMOD`. Include modules in the working set of the installation.

- Use your own version of a `USERMOD`. Your version can include `++MOVE` statements from both CICS-supplied `USERMODs`.

If you intend changing a CICS-supplied `USERMOD`, to choose modules to install in the MVS link pack area, take a copy of the `USERMOD` and update the copy only. If you have copied the *hlq.SDFHSAMP* library, for instance, when changing user-replaceable programs, you already have copies of the CICS-supplied `USERMODs`. If the original *hlq.SDFHSAMP* library is serviced, and the `USERMOD` is modified, you can reflect the changes in your version.

2. Choose which read-only modules to install in the MVS link pack area and edit your copy of the `SMP/E USERMOD`.
 - a. Comment out the `++MOVE` statements for the modules that you do not want to install in the LPA.

- b. Move the remaining ++MOVE statements for the modules that you do want to install in the LPA one column to the left, so that the ++MOVE statements start in column one of the USERMOD module.
 - c. Add ++MOVE statements for your user application program modules that you want to install in the LPA, with the ++MOVE statements starting in column one of the DFH\$UMOD module.
3. Receive the USERMOD into the CICS global zone. SMP/E moves those load modules you have specified from the named CICS target library (*hlq.SDFHLOAD* or *hlq.SDFHAUTH*) into the CICS LPA library.

To receive and apply the CICS-supplied sample USERMODs in DFH\$UMOD, you can use the associated job DFHLPUMD, which is tailored to your CICS environment and stored in the *hlq.XDFHINST* library when you run the DFHISTAR job.
4. Apply the USERMOD to the LPA zone. When you apply the USERMOD, you also update the corresponding LMOD entries in the target zone SMPCSI.
5. To enable CICS to use the modules that you have installed in the MVS link pack area, re-IPL your MVS with CLPA specified.

Also, specify that CICS uses modules from the LPA>. You can also control which modules are used from the MVS link pack area in several ways. See [“Controlling the use of modules from the MVS link pack area” on page 113](#) for more information.

Controlling the use of modules from the MVS link pack area

Using customizable options, CICS allows you to control which of the eligible modules are executed from the MVS link pack area.

The methods for controlling the use of modules from the MVS link pack area do not apply to the modules DFHCSVC, DFHDSPEX, and DFHIRP. These modules are used only from the MVS link pack area.

Modules in the MVS link pack area from *hlq.SDFHAUTH*

CICS uses standard MVS load facilities for modules installed in the MVS link pack area from the CICS APF-authorized library, *hlq.SDFHAUTH*.

That is, such a module is used from the first of the following locations where it is found:

1. STEPLIB concatenation
2. MVS link pack area
3. MVS LNKLST

To use any of the CICS modules installed in the MVS link pack area from the *hlq.SDFHAUTH* library, you must remove any version of the module from the *hlq.SDFHAUTH* library or any other library in the STEPLIB concatenation.

You can prevent CICS using modules installed in the MVS link pack area from the *hlq.SDFHAUTH* library by installing versions of those modules in a library in the STEPLIB concatenation.

CICS then uses the versions of the modules from the STEPLIB concatenation into the CICS address space, rather than any versions that might be in the MVS link pack area.

Modules in the MVS link pack area from *hlq.SDFHLOAD*

CICS system initialization parameters and resource definitions control the use of CICS modules installed in the MVS link pack area from the *hlq.SDFHLOAD* library.

The *hlq.SDFHLOAD* library is used for non-nucleus CICS modules, and some CICS nucleus modules. You can also use the library for your own user application programs.

Using modules from the MVS link pack area

You must follow these steps to use the CICS modules installed in the MVS link pack area from the *hlq.SDFHLOAD* library.

- Copy the modules into a CICS LPA library. That is, you do not have to remove them from the *hlq.SDFHLOAD* library.
- Specify the system initialization parameter LPA=YES. CICS then uses the following search order:
 1. MVS link pack area
 2. DFHRPL DD concatenation
- For a non-nucleus CICS module or user application program, specify USELPACOPY(YES) on the associated PROGRAM resource definition. These modules are identified in the CICS-supplied USERMODs by this statement:

```
/* Not loaded from LPA unless USELPACOPY is set to Y in the CSD */
```

For each CICS-supplied LPA-eligible module that requires USELPACOPY(YES) specified in its associated PROGRAM resource definition, you must create your own resource definition with USELPACOPY(YES) specified, and use it instead of the CICS-supplied resource definition, because you cannot modify the CICS-supplied resource definitions. For example, you can use the DFHCSDUP utility program for these purposes:

1. Copy the CICS-supplied resource groups that contain the module definitions to new resource groups.
2. For each module that requires USELPACOPY(YES), change the PROGRAM resource definition in the new resource groups to specify USELPACOPY(YES).
3. Add your new resource groups to a new group list; that is, at the start of the list.
4. Append the CICS-supplied group list DFHLIST, or your own equivalent of that group list, to your group list. Alternatively, include DFHLIST on the GRPLIST system initialization parameter as well as your group list.
5. Remove the CICS-supplied groups that you have copied.

When the program definitions have been changed on the CSD, perform the following steps:

- Reinitialize the CICS catalogs if you have been using modules not in the MVS link pack area, and now want to use those modules from the MVS link pack area
- Specify your new group list (and DFHLIST if your group list does not include the list of resource groups provided in DFHLIST) on the GRPLIST system initialization parameter.

A sample DFHCSDUP job for all CICS LPA-eligible jobs is in [“Sample DFHCSDUP job to specify USELPACOPY\(YES\)”](#) on page 117.

In the above example, instead of steps [“3” on page 114](#) and [“4” on page 114](#), you can use the CEDA transaction for these tasks:

- Copy your group list to create a new group list.
- Add the new (USELPACOPY(YES)) groups to the new group list *in the same place* as the original, CICS-supplied, groups.
- CICS uses eligible modules installed in the MVS link pack area, if the following criteria are met:
 - You have *not* specified the name of the module on the CICS system initialization parameter PRVMOD.
 - The module has not already been loaded from the DFHRPL concatenation.
- If CICS cannot find an eligible module in the MVS link pack area, it loads the private (non-shared) version into the CICS address space from the DFHRPL concatenation, after issuing the message DFHLD0109I to warn you that the module is not in the MVS link pack area. See [“Handling the module-not-found warning message, DFHLD0109I”](#) on page 116 for more information about this message.
- CICS assumes that the PL/I modules, IBMBPSLA and IBMBPSMA, are installed in the MVS link pack area and issues message DFHLD0109I if it fails to find them there. If you want your PL/I application programs to run with the PL/I shared library, ensure that the modules IBMBPSLA and IBMBPSMA are

installed in the MVS link pack area, in the *hlq.SDFHLOAD* library or in another library in the CICS DFHRPL library concatenation.

- Program list tables (PLTs) identify programs that run during CICS initialization and termination. There are two phases of PLT execution. You must place first-phase PLT programs in the DFHRPL concatenation, but can place second-phase PLT programs in a dynamic LIBRARY concatenation. However, before installing PROGRAM resource definitions for first-phase PLTPI programs and PLTSD programs (for example, early in CICS initialization), CICS scans the MVS link pack area for those programs and issues message DFHLD0109I if it cannot find such a program there.

Similarly, before installing PROGRAM resource definitions for global and task-related user exit programs (for example, early in CICS initialization), CICS scans the MVS link pack area for those programs, and issues message DFHLD0109I if it cannot find such a program there.

DFHLD0109I is issued only to inform that the program will be loaded from the DFHRPL concatenation, and does not indicate any error.

Specifying USELPACOPY(YES)

For every non-nucleus CICS module or user application program that you have moved to the MVS link pack area, that is, have removed from the DFHRPL concatenation, ensure that you have specified USELPACOPY(YES) on the associated PROGRAM resource definition. Otherwise, CICS cannot find the module and might fail to start successfully. See [“Sample DFHCSDUP job to specify USELPACOPY\(YES\)” on page 117](#) for a sample job to specify USELPACOPY(YES).

Using modules from DFHRPL

You can prevent CICS from using modules installed in the MVS link pack area from the *hlq.SDFHLOAD* library by specifying the NO option on the **LPA** system initialization parameter or by specifying the name of the module on the **PRVMOD** system initialization parameter.

- Specify NO on the **LPA** system initialization parameter.

The NO option prevents CICS from using any modules installed into the MVS link pack area from the *hlq.SDFHLOAD* library. CICS tries to load the modules from libraries in the DFHRPL concatenation.

You might use this option when you want to run CICS to test a lot of LPA-eligible modules before installing them in the MVS link pack area. For example, you can add the *hlq.SDFHLPA* library to the DFHRPL concatenation while testing CICS modules for the MVS link pack area. When you have verified the use of those modules from the MVS link pack area, specify the LPA=YES system initialization parameter and remove the *hlq.SDFHLPA* library from the DFHRPL concatenation.

- Specify the name of the module on the **PRVMOD** system initialization parameter.

```
PRVMOD={name | (name1, name2, ... ) }
```

Specifying the module name prevents CICS from using the specified modules from the MVS link pack area for only the run of CICS on which the **PRVMOD** parameter is specified. You might use the **PRVMOD** parameter when you want to run CICS to test a new version of an LPA-eligible module before replacing the version already in the MVS link pack area.

Specify the full module name on the **PRVMOD** parameter, including any suffix; for example, DFHMCP1\$. If only one module is named, the parentheses are optional. The **PRVMOD** parameter can span input lines. However, do not split module names across lines, because CICS system initialization adds a comma at the end of every input line that does not already end with a comma. The only validity check performed on a module name is to ensure that it does not exceed eight characters.

You cannot code the **PRVMOD** parameter in the DFHSIT module; you specify it in the **PARM** parameter, in the SYSIN data set, or through the system console.

- For a non-nucleus CICS module or user application program, specify USELPACOPY(NO), the default, on the associated PROGRAM resource definition. These modules are identified in the CICS-supplied USERMODs by the statement:

```
/* Not loaded from LPA unless USELPACOPY is set to Y in the CSD */
```

You might use the USELPACOPY(NO) option of the PROGRAM resource definition for a more permanent exclusion of an LPA-resident module than for the single run of CICS control provided by the **PRVMOD** system initialization parameter.

Verifying modules for the MVS link pack area

When verifying new versions of modules to be installed into the MVS link pack area, you can use any of the following options to instruct a CICS region to use the new versions from the DFHRPL concatenation.

- The LPA=NO system initialization parameter
- The PRVMOD system initialization parameter
- The USELPACOPY(NO) option of the associated PROGRAM resource definition (where applicable)

For further information about these options, see [“Using modules from DFHRPL” on page 115](#)

In all cases, you must install the new versions of the modules into the *hlq.SDFHLOAD* library or another library in the DFHRPL concatenation.

If you are verifying many CICS LPA-eligible modules, you might choose to add the *hlq.SDFHLPA* library to the DFHRPL concatenation, allowing you to check that the modules you have installed in the MVS link pack area are being loaded from there.

The CICS-supplied USERMODs use SMP/E to move CICS LPA-eligible modules into the *hlq.SDFHLPA* library. Similarly, if you use SMP/E to apply service to any of those modules, the versions in the *hlq.SDFHLPA* library are updated. The updated versions of the modules are used from the MVS link pack area after you next re-IPL your MVS with CLPA specified. Until then, if you add the *hlq.SDFHLPA* library to the DFHRPL concatenation of your CICS region, and specify that CICS is not to use the version of the modules in the MVS link pack area, the updated versions of the modules are used from the DFHRPL concatenation.

After you have installed and verified the use of modules from the MVS link pack area, remove the versions of the modules from the DFHRPL concatenation of your CICS startup job.

You can find out whether CICS is loading modules from the MVS link pack area or the DFHRPL concatenation by reviewing the index of a system dump for the CICS region started with the system initialization parameter LPA=YES. Modules loaded from the MVS link pack area have the dump option LD=3.

Handling the module-not-found warning message, DFHLD0109I

CICS issues message DFHLD0109I if it searches the MVS link pack area for a module installed there from *hlq.SDFHLOAD* and fails to find it.

If you see the module-not-found warning message, check that you have specified USELPACOPY(YES) on the associated PROGRAM resource definition, if applicable. For further information about using modules loaded in the MVS link pack area from the *hlq.SDFHLOAD* library, see [“Modules in the MVS link pack area from hlq.SDFHLOAD” on page 113](#).

CICS uses console routing code 11 for this particular message, which allows you to control the output of this message. For example, you can perform these actions:

1. Exclude, as required, routing code 11 from specific MVS console definitions in the CONSOLxx member of SYS1.PARMLIB.
2. Use the MVS VARY command to prevent this message from appearing on specified consoles by omitting route code 11 from a VARY command that defines which routing codes go to specified devices:

```
VARY devnum,CONSOLE,ROUT=(rtcode,rtcode,...)
```

Alternatively, you can remove route code 11 from those already defined by using the following VARY command:

```
VARY devnum,CONSOLE,DROUT=(11)
```

3. Use the MVS message processing facility (MPF) to inhibit the message. To use MPF, code an entry specifying the CICS message number in the MPFLSTxx member of SYS1.PARMLIB.

CICS assumes that the following PL/I modules are LPA eligible and issues message DFHLD0109I if it fails to find them there:

- IBMBPSLA
- IBMBPSMA

Sample DFHCSDUP job to specify USELPACOPY(YES)

The standard IBM-supplied program definitions in the CSD all specify USELPACOPY(NO). If you copy or move to the LPA the IBM programs defined by definitions in the CICS system definition data set (CSD), the next step is to modify the USELPACOPY attribute to ensure that CICS uses the LPA copy.

To simplify this task:

- IBM supplies, in the DFH\$ULPA member of the SDFHSAMP library, an alternate set of DEFINE statements for all the IBM-supplied programs. All the programs defined in DFH\$ULPA specify USELPACOPY(YES).
- If you do not want all the programs to be defined for LPA use, edit the member to remove the programs that are to remain as USELPACOPY(NO).
- The USELPACOPY(YES) versions are all defined in one new group called DFH\$ULPA. Change this group name if you want to use your own name.
- Run the sample DFHCSDUP job shown in [Figure 4 on page 117](#) to add the DFH\$ULPA versions of the definitions to your CSD.
- You do not have to remove the standard definitions from DFHLIST. If you specify your group list after DFHLIST on the GRPLIST system initialization parameter, you ensure that the modified definitions override the standard definitions.

```
//LPAMODS      JOB   (account_details),MSGCLASS=A,MSGLEVEL=(1,1),
//              CLASS=A,NOTIFY=userid
//DEFULPA      EXEC  PGM=DFHCSDUP
//STEPLIB      DD   DSN=CICSTS56.CICS.SDFHLOAD,DISP=SHR
//SYSPRINT     DD   SYSOUT=*
//DFHCSD       DD   DSN=user.CICSTS56.CICS.DFHCSD,DISP=OLD
//SYSIN        DD   DSN=CICSTS56.CICS.SDFHSAMP(DFH$ULPA),DISP=SHR
//*
//*
```

Figure 4. Sample DFHCSDUP job for all CICS LPA-eligible modules

Installing CICSplex SM modules in the MVS link pack area

There are benefits and points to consider if you use the MVS link pack area (LPA).

Benefits of using the LPA include:

- **Sharing.** Two or more CICS regions in the same MVS image can share modules in the LPA, giving an overall reduction in the total working set.
- **Integrity.** The LPA is page-protected, even against key 0 programs, so all modules placed there are automatically protected against overwriting by other programs such as CICS applications. This integrity feature applies equally to a single CICS system in the processor.

Every CICSplex SM module installed in the LPA can be used only by the release of CICSplex SM to which it relates.

CICSplex SM supplies prebuilt SMP/E USERMODs as members in the CICSTS56.CPSM.SEYUSAMP library. The USERMOD is:

EYU\$UM01 - Local MAS modules

These USERMODs contain ++MOVE statements for each module that is eligible for the extended link pack area (ELPA). A read-only module that can reside above 16 MB is eligible for the ELPA.

CICSplex SM allocates an empty library for your use, called SYS1.CICSTS56.CPSM.SEYULPA. You can use SYS1.CICSTS56.CPSM.SEYULPA as the LPA library or you can add the modules to another LPA library.

If you are going to use SYS1.CICSTS56.CPSM.SEYULPA, verify that you have already authorized this library, described in [“Authorizing the CICS and CICSplex SM libraries”](#) on page 61, and that you have applied appropriate security. You can give the SYS1.CICSTS56.CPSM.SEYULPA library your own high-level index. If you do, you must specify the new index on the LINDEX parameter of the DFHISTAR job.

Space requirements for CICSplex SM modules

Allow enough space in the link pack area for the installation of the selected CICSplex SM modules.

The total space depends on how the modules are packaged into the link pack area by the operating system.

Installing CICSplex SM modules into the LPA

In this context, the term *install* means to move or copy a module to the SYS1.CICSTS56.CPSM.SEYULPA library by using SMP/E, or by using a copying method that reblocks the copied modules when the target data set has a smaller block size than the data set you are copying from; for example, by using the COPYMOD function of the IEBCOPY program.

CICSplex SM modules that are eligible for inclusion in the link pack area are listed in [“CICSplex SM modules eligible for the MVS link pack area”](#) on page 119.

CICSplex SM creates default definitions for the LPA-eligible modules specifying USELPACOPY(YES). So you do not have to modify these definitions to be able to search the LPA. You can, however, specify that the LPA is searched by using the CICS system initialization parameters LPA and PRVMOD. Specify LPA=YES for CICS to search the LPA for these programs. If you specify LPA=NO, which is the default, the LPA is not searched for these programs.

If you specify LPA=YES but have not moved the modules to the LPA, message DFHLD0109I is issued for each module. If you specify LPA=YES and have moved these modules to the LPA, you can use the PRVMOD system initialization parameter to control which modules are to be used from the LPA. See [“Using modules from DFHRPL”](#) on page 115 for more information.

Do not relink-edit the modules to put them into the SYS1.CICSTS56.CPSM.SEYULPA library. CICSplex SM modules, as supplied, have the necessary attributes that cause MVS to load them automatically above 16 MB (into the ELPA).

The MVS link pack area has both pageable and fixed areas. Although you can install CICSplex SM modules into the fixed areas, for performance reasons, use the pageable areas.

Modules to be loaded into the MVS pageable link pack area (PLPA) must have been link-edited with the RENT attribute. The library that contains these modules must be named in an LPALSTxx member of the SYS1.PARMLIB library.

To install modules in the CICSplex SM LPA library, and to ensure that SMP/E can continue to service them, complete the following steps for one or both of the CICSplex SM-supplied USERMODs:

1. Receive the USERMOD into the CICSplex SM global zone and apply it to the CICSplex SM target zone.
2. Define the SYS1.CICSTS56.CPSM.SEYULPA library to your MVS.

Receiving and applying the USERMOD

To receive and apply the CICSplex SM-supplied USERMOD, in EYU\$UM01, you can use the sample job EYULPMOD, which is tailored by DFHISTAR and stored in the CICSTS56.XDFHINST library when you run the DFHISTAR job.

Receive the USERMOD into the CICSplex SM global zone and apply it to the CICSplex SM target zone. This causes SMP/E to move those load modules you have specified from the named CICSplex SM target library (either CICSTS56.CPSM.SEYUAUTH or CICSTS56.CPSM.SEYULOAD) into the SYS1.CICSTS56.CPSM.SEYULPA library.

When the USERMOD is applied, the corresponding LMOD entries in the target zone SMP CSI are updated. Either or both USERMODs can be applied depending on your enterprise's requirements.

Do not accept the USERMOD into the distribution zone, and, for the time being, do not apply it to any other target zone.

Defining the SYS1.CICSTS56.CPSM.SEYULPA library to your MVS

Add the full name of the SYS1.CICSTS56.CPSM.SEYULPA library to an LPALEX member of SYS1.PARMLIB to ensure that the library contents are loaded into the PLPA at the next IPL of your system when CLPA is specified.

When you have defined the SYS1.CICSTS56.CPSM.SEYULPA library to MVS, re-IPL your MVS with CLPA specified so that the modules in the SYS1.CICSTS56.CPSM.SEYULPA library can be used from the LPA.

Use the USERMOD EYU\$UM01, supplied in the SEYUSAMP library, to move the LPA-eligible modules into the SEYULPA library. These modules are listed in [“CICSplex SM modules eligible for the MVS link pack area”](#) on page 119.

Applying maintenance to LPA modules

Use the SMP/E RESTORE function to back off the USERMOD before modules in the LPA are updated or copied. Afterwards, you can reapply the USERMOD.

CICSplex SM modules eligible for the MVS link pack area

CICSplex SM modules that are eligible for installation in the MVS link pack area (LPA) are specified in the CICSplex SM supplied USERMOD, EYU\$UM01.

The following table lists the CICSplex SM modules that are eligible for inclusion in the LPA. The versions of these modules with the prefixes CJA, CJB, CJC, CJD, CJE, and CJF in place of the prefix EYU, are also eligible for inclusion in the LPA. For example, modules CJA9BA01, CJB9BA01, CJC9BA01, CJD9BA01, CJE9BA01, and CJF9BA01 as well as EYU9BA01, are eligible. The prefix relates to the CICS release specific agent code of the underlying module.

You must allow enough space in the link pack area for the installation of the selected CICSplex SM modules. The total space you require depends on how the modules are packaged into the link pack area by the operating system and a local MAS requires approximately 2034KB.

All CICSplex SM modules are installed in the LOAD library and loaded in the ELPA.

Table 12. CICSplex SM modules eligible for the LPA	
Module	Description
EYU9BA01	BAS MAS object
EYU9CM01	MAS communications
EYU9MN01	MAS monitor
EYU9NA01	MAS agents
EYU9PS01	MAS real-time analysis

Table 12. CICSplex SM modules eligible for the LPA (continued)	
Module	Description
EYU9TS01	MAS topology
EYU9WM01	MAS workload management
EYU9XC01	MAS cache
EYU9XD01	MAS data repository
EYU9XL01	MAS kernel linkage
EYU9XM01	MAS message format
EYU9XQ01	MAS queue manager
EYU9XS01	MAS common services
EYU9XZ01	MAS trace

Defining CICS IPCS exit control data to MVS

If you use the MVS interactive problem control system (IPCS) to format and analyze CICS system dumps, ensure that the release-specific CICS formatting routines are defined and available to MVS.

The formatting routine for use under IPCS has the release identifier as part of its name; that is, DFHPD730. You define this formatting routine to IPCS when formatting system dumps. The CICS formatting routine is release-specific, so, if you run more than one release of CICS, ensure that you use the correct version for the system dump you are formatting.

The DFHIPCSP CICS exit control data

IPCS provides an exit control table with IMBED statements to enable other products to supply exit control information.

The IPCS default table, BLSCECT, normally in the SYS1.PARMLIB library, has the following entry for CICS:

```
IMBED MEMBER(DFHIPCSP) ENVIRONMENT(ALL) /* CICS */
```

Ensure that your IPCS job can find the CICS-supplied DFHIPCSP member. The DFHIPCSP member is in the *hlq.SDFHPARM* library. You can either copy the DFHIPCSP member into SYS1.PARMLIB, so that it is in the same default library as BLSCECT, or provide an IPCSPARM DD statement to specify the library containing the IPCS control tables:

```
//IPCSARM DD DSN=SYS1.PARMLIB,DISP=SHR For BLSCECT
// DD DSN=CICSTS56.CICS.SDFHPARM,DISP=SHR For DFHIPCSP
```

For information about using IPCS to format CICS system dumps, see [Displaying and formatting dumps with IPCS](#).

Preparing to use the CICSplex SM IPCS tools

The interactive problem control system (IPCS) provides MVS users with interactive diagnostics of software failures. Before using IPCS, you must update BLSCECT, update library allocations and set SDUMP options.

You can use IPCS to format and analyze SDUMPs produced by CICSplex SM or stand-alone dumps obtained while CICSplex SM was active in the system being dumped. You can either view the dumps at your terminal or print them.

CICSplex SM provides a dump formatting routine that you can use with the VERBEXIT subcommand to format CMAS, MAS, WUI server, or SMSS dumps.

- For more information about IPCS, see the [z/OS MVS IPCS User's Guide](#).

Updating BLSCECT

IPCS provides an exit control table, BLSCECT, which is normally in SYS1.PARMLIB. This table contains imbed statements that you can update to enable other products to supply exit control information.

Perform the following steps to update BLSCECT:

1. Update the BLSCECT table with the following IMBED statement:

```
IMBED MEMBER(EYUIPCSP) ENVIRONMENT(ALL)
```

EYUIPCSP identifies the CICSplex SM formatting routine as EYU9D560 with a VERB name of CPSM560.

2. Make sure EYUIPCSP can be found by your IPCS job by doing one of the following steps:

- Copy EYUIPCSP from the CICSTS56.CPSM.SEYUPARM library into the same library as BLSCECT, usually SYS1.PARMLIB.
- Provide an IPCSPARM DD statement to specify the library that contains the IPCS control tables. For example, the DD statement for a batch TSO session might look like this:

```
//IPCSPARM DD DSN=SYS1.PARMLIB,DISP=SHR      for BLSCECT
//          DD DSN=CICSTS56.CPSM.SEYUPARM,DISP=SHR for EYUIPCSP
```

For more information about SYS1.PARMLIB library members related to IPCS, see [z/OS MVS IPCS Customization](#).

Updating library allocations

If CICSTS56.CPSM.SEYULINK is not in the linklist, include the EYU9D560 IPCS user exit routine.

Make sure that the EYU9D560 IPCS user exit routine is in a library in the linklist or a library that is accessed by the JOBLIB, STEPLIB, or TASKLIB option of the IPCS command, during the IPCS session. To accomplish this, perform one of these actions:

- Allocate CICSTS56.CPSM.SEYULINK to the intended DD statement.
- Copy CICSTS56.CPSM.SEYULINK (EYU9D560) to an appropriate library.
- Invoke IPCS, using the TASKLIB keyword to allocate CICSTS56.CPSM.SEYULINK .

For example, issue the TSO COMMAND:

```
IPCS NOPARM TASKLIB('CICSTS56.CPSM.SEYULINK ')
```

MVS Program properties table entries

You can optionally define some CICS properties to MVS in the MVS program properties table (PPT).

[Figure 5 on page 122](#) is an example of a CICS PPT entry in the SCHEDxx member of SYS1.PARMLIB.

```

/*****
/*          Program Properties table addition          */
/*          for the CICS program, DFHSIP              */
/*          */
/* The following defaults apply to this CICS entry:    */
/*          */
/* No affinity to a particular processor      (AFF(NONE)) */
/* Can be canceled                          (CANCEL)    */
/* Requires data set integrity              (DSI)        */
/* Not a privileged job                     (NOPRIV)     */
/* Password protection is required          (PASS)       */
/* Not a system task                       (NOSYST)      */
/* Protection key 8                        (KEY(8))      */
PPT PGMNAME(DFHSIP)      /* Add program name DFHSIP to the PPT */
  NOSWAP                 /* Non-swappable                      */
  NOPREF                 /* No preferred storage required     */
/*          */

```

Figure 5. Sample CICS PPT entry

For information about defining options in the PPT, see the *z/OS MVS Initialization and Tuning Guide*.

RACF password checking

If your installation has an MVS program properties table (PPT) entry for the DFHSIP program, ensure that the NOPASS option is *not* set in the PPT because this option bypasses password and RACF authorization checking.

For information about defining PPT entries for CICS in the SCHEDxx member of the SYS1.PARMLIB library, see [z/OS MVS Initialization and Tuning Reference](#).

Non-swappable CICS regions

For performance reasons, consider making your CICS regions nonswappable by specifying the NOSWAP option of the PPT entry in the SCHEDxx member of SYS1.PARMLIB. If you specify NOSWAP on the PPT entry in the SCHEDxx member of SYS1.PARMLIB, PPTNSWP is set to ON in the PPT internal control block.

If you specify NOPASS on the MVS PPT entry in the SCHEDxx member of SYS1.PARMLIB, PTNPAS is set to ON in the PPT internal control block.

MVS protection key for CICS

To use the storage protection facility of CICS, you must specify the system initialization parameter **STGPROT=YES** or allow the system initialization parameter to default to YES. When you operate CICS with storage protection, CICS observes the storage keys and execution keys that you specify in system and resource definitions.

Configuring MVS workload management for CICS systems

You can use z/OS workload management to manage sysplex resources across MVS subsystems, in parallel with the existing system resource management facilities.

For information about z/OS workload management and planning, see [z/OS MVS Planning: Workload Management](#).

The following topic ensures that CICS performance parameters correspond to the policies defined for z/OS workload management.

Matching CICS performance parameters to service policies

Ensure that the CICS performance parameters are compatible with the workload manager service policies used for the CICS workload.

In general, define CICS performance objectives to the MVS workload manager first and observe the effect on CICS performance. When the MVS workload manager definitions are working correctly, then consider

tuning the CICS parameters to further enhance CICS performance. However, use CICS performance parameters as little as possible.

Consider using these performance attributes:

- Transaction priority, passed on dynamic transaction routing. The priority assigned by the CICS dispatcher must be compatible with the task priority that is defined to MVS workload manager.
- Maximum number of concurrent user tasks for the CICS region.
- Maximum number of concurrent tasks in each transaction class.

Implementing MVS automatic restart management

You can exploit MVS automatic restart management to implement a sysplex-wide integrated automatic restart mechanism.

To use the MVS automatic restart manager facility, follow these steps:

1. Implement automatic restart management on the MVS images on which the CICS workload is to run.
2. Ensure that CICS startup JCL used to restart CICS regions is suitable for MVS automatic restart management.
3. Specify appropriate CICS START options.
4. Specify appropriate MVS workload policies.

Implementing MVS automatic restart management for CICS generally involves the following steps:

- Ensure that the MVS images available for automatic restarts have access to the databases, logs, and program libraries required for the workload.
- Identify those CICS regions for which you want to use automatic restart management.
- Define restart processes for the candidate CICS regions.
- Define ARM policies for the candidate CICS regions. Allow the RESTART_ORDER LEVEL to default to 2. CICS regions are generally level 2 in the ARM restart order, after Db2 and DBCTL.

For information about MVS automatic restart management, see [Automatic restart management](#) and [z/OS MVS Setting Up a Sysplex](#).

MVS cross-system MRO definitions

You can use CICS interregion communication (IRC) for multiregion operation (MRO) between CICS regions across MVS images in a sysplex.

IRC exploits the cross-system coupling facility (XCF) of MVS and removes the need for z/OS Communications Server to communicate between MVS images in the same sysplex.

In a sysplex, you must install DFHIRP from the highest release of CICS running in that MVS image.

Sysplex overview

A sysplex consists of multiple MVS systems, coupled together by hardware elements and software services. In a sysplex, MVS provides a platform of basic multisystem services that multisystem applications like CICS can use. As the workload grows, you can add MVS systems to the sysplex to meet the requirements of the greater workload.

To use XCF to communicate in a sysplex, you specify the XCF group name to CICS using the SIT XCFGROUP parameter. The default name is DFHIR000. If you specify a different group name, only members that specify that same group name can communicate using XCF. For more information about XCFGROUP, see [XCFGROUP system initialization parameter](#).

MVS XCF for MRO

When you format the primary and alternate couple data sets used by the XCF component of MVS, make sure that the following conditions are met:

- The value specified for the MAXMEMBER parameter must be large enough to handle the number of CICS regions and users of the EXCI in the CICS XCF group.

In a single sysplex, a theoretical maximum number of 1023 CICS regions can participate in XCF/MRO using the same XCF group. However, the maximum size of the XCF group is reduced if you set the MVS MAXMEMBER parameter, used to define XCF couple data sets, to a lower limit. When you calculate the maximum number of members in the CICS XCF group, allow one member for each of these items:

- Each CICS region to run on an MVS image in the sysplex.
- Each pipe that is allocated by a user of the external CICS interface (EXCI). For information about EXCI users and pipes, see [The EXCI CALL interface](#).

To list the members in the CICS XCF group, you can use the MVS DISPLAY command. The name of the CICS group is always DFHIR000, so you can use this MVS command:

```
DISPLAY XCF, GROUP, DFHIR000, ALL
```

- The value specified for the MAXGROUP parameter must be large enough for the CICS XCF group to be established.

MVS ASREXIT - SYMREC Authorization Exit

A CICS program might call the first failure symptoms component. The component uses the MVS SYMREC macro to write symptom records to the MVS SYS1.LOGREC data set, in addition to, or instead of, a job log.

The SYMREC authorization exit, ASREXIT, must be in effect to allow CICS to use the SYMREC macro call; otherwise, the call fails with return code 12, reason code 3868 (X'F1C').

When SYMREC is called by CICS, the ASREXIT routine issues a return code that permits the SYMREC to be successfully written.

The MVS sample exit programs ASREXT0 and ASREXT1, supplied in SYS1.SAMPLIB, are suitable for this purpose. For further information about these exits, see [z/OS MVS Installation Exits](#). The ASREXIT routine can determine whether CICS is the caller by testing EPLPNAME for the value DFHSIP, except in the following circumstances:

- When DFHSIP is renamed, in which case EPLPNAME contains the new name.
- When DFHSIP is the subject of an MVS LINK, in which case EPLPNAME contains the name of the program issuing the MVS LINK, unless it too is the subject of an MVS LINK.

If you choose this method, code your ASREXIT routine to allow for these exceptions.

An alternative method of coding the ASREXIT routine is shown in [Figure 6 on page 125](#). This method is not affected by the exceptions described earlier.

```

TITLE 'SYMREC SAMPLE EXIT'
ASREPL
PRINT NOGEN
IHAPSA
IKJTCB
PRINT GEN
DFHAFCD
EJECT
ASREXIT CSECT
ASREXIT AMODE 31

ASREXIT RMODE ANY
        USING *,R15                      Temporary addressability
        MODID BR=YES
        DROP R15
        STM R14,R12,12(R13)             Save the caller's registers
        LR R12,R15
        USING ASREXIT,R12
        L R3,0(,R1)                     Load the address of the EPL
        USING EPL,R3                    Get addressability
        LA R15,RCREJECT                 Preset "reject" return code
        USING PSA,0
        L R1,PSATOLD                    Point at current TCB
        USING TCB,R1
        L R1,TCBEXT2                    Point at TCB extension
        DROP R1
        USING TCBXTNT2,R1
        ICM R1,B'1111',TCBCAUF          Point at AFCB; is there one?
        BZ SETRC                        No, branch
        DROP R1
        USING DFHAFCB,R1
        CLC AFIDENT,=C'AFCX'            Is it a genuine CICS AFCB?
        BNE SETRC                       No, branch
        CLI AFVER,AFVER1                Is it at least Version 1?
        BL SETRC                        No, branch
        AH R1,AFLENG                    Add length of AFCB's DW
        DROP R1                         table.
        USING AFTSTART-AFPFXLEN,R1      Allow for AFCB prefix length
        ICM R1,B'1111',AFTAFCS          Point at AFCS; is there one?
        BZ SETRC                        No, branch
        DROP R1
        LA R15,RCWRITE                  Set "write" return code
        DS 0H
        ST R15,EPLRETC                  Store return code
        DROP R0
        DROP R3
        DROP R12
        EXIT LM R14,R12,12(R13)          Restore caller's registers
        BR R14                          Return
        LTORG *
R1      EQU 1                           Register 1
R3      EQU 3                           Register 3
R12     EQU 12                          Register 12
R13     EQU 13                          Register 13
R14     EQU 14                          Register 14
R15     EQU 15                          Register 15
RCREJECT EQU X'0C'                      Return code C
RCWRITE EQU X'00'                      Return code 0
END*   CONSTANTS

```

Figure 6. An example of coding the ASREXIT routine.

Setting up VSAM RLS support

These topics provide an overview of each task you need to perform if you want to use VSAM RLS to enable CICS regions to share VSAM data sets.

Defining the main coupling facility lock structure

VSAM RLS support requires the coupling facility to define a main lock structure, IGWLOCK00, for cross-system locking. You can also optionally create up to 256 secondary lock sets per sysplex.

See [z/OS DFSMSdfp Storage Administration](#) for information about calculating the size you require for the lock structure.

The amount of coupling facility space required depends on several characteristics of your hardware configuration and the applications that you run:

- The number of processors you have
- The power of your processors
- Your ratio of nonupdate activity to update activity
- Your ratio of recoverable updates to nonrecoverable updates
- Your ratio of sequential requests to direct requests

You define the lock structure in the CFRM policy with the IXCMIAPU utility.

As of z/OS 2.1, up to 256 secondary lock sets can be created in addition to the primary IGWLOCK00. Using a secondary lock structure can prevent workloads interfering with each other and improve system and application availability. Secondary lock structures help prevent locking constraints and allow isolation of workloads.

Defining coupling facility cache structures and cache sets

VSAM RLS support requires the coupling facility to define *cache structures* for cross-system buffer invalidation. You determine the number and size of cache structures you require.

The number you require depends on factors such as these:

- The number of coupling facilities you have
- The amount of space in each coupling facility
- The amount of data that is accessed through each coupling facility

See [z/OS DFSMSdfp Storage Administration](#) for information about calculating the amount of space you require for the cache structures. If you have previously used data sets in LSR mode, the total amount of coupling facility space allocated to cache structures will not be less than the amount of storage you were using for LSR pools, including any hiperspace buffers.

To achieve performance benefits:

- Make the size of the cache larger
- Divide cache structures appropriately across coupling facilities

You define cache structures in the CFRM policy with the IXCMIAPU utility.

Defining cache sets

You define cache sets with the ISMF control data set (CDS) application. A cache set maps onto one or more cache structures. If you specify more than one cache set, the data sets can be rebound to another cache structure in the set if a cache structure fails.

See [z/OS DFSMSdfp Storage Administration](#) for more information about cache sets.

Preparing for RLS access

You need to define SMS storage classes and alter data set attributes to prepare for VSAM RLS access.

Defining SMS storage classes for RLS access

Before you can use VSAM RLS, you require one or more storage classes that specify a non-blank cache set name. Use the ISMF storage class application to specify a cache set name when you define or alter a storage class, together with weighting parameters for tuning, such as CF DIRECT WEIGHT and CF SEQUENTIAL WEIGHT. See [z/OS DFSMSdfp Storage Administration](#) for more information about defining SMS storage classes.

Altering data set attributes for RLS access

Before you can use a data set in RLS access mode, you must ensure that it is eligible.

To be eligible for RLS, data sets must meet the following conditions:

- Data sets must reside in SMS managed storage.
- Data sets must specify a storage class that has a nonblank cache set name.
- Data set recoverability attributes must be defined in the ICF catalog and not in the CICS file control resource definition, where they are ignored for RLS.

You can specify a data set's attributes using the Access Method Services (AMS) `DEFINE CLUSTER` or `ALTER CLUSTER` commands.

Specifying a LOG parameter of NONE, UNDO, or ALL ensures that the recoverability of the data set is defined. You cannot open files in RLS mode if the LOG parameter of the associated data set is UNDEFINED. If you specify LOG(ALL), you must also specify a forward recovery log stream on the LOGSTREAMID parameter.

To use backup while open (BWO) for an RLS-accessed sphere, specify the BWO parameter. Specifying BWO(TYPECICS) means that backup while open can be used. All other values for BWO, including undefined, mean that backup while open is not allowed. BWO(TYPECICS) is valid only if LOG(ALL) and LOGSTREAMID are also specified.

- Data sets must not specify the IMBED attribute.

If you have some data sets that specify IMBED, you must remove the IMBED option before you can use the data sets in RLS mode. Redefine a new data set without IMBED and use the AMS REPRO function to copy the old data set to the new data set.

Note that RLS supports the REPLICATE cluster attribute. It does not provide any performance benefit. Removing RLS might save storage space.

Considerations for migrating from non-RLS to RLS access mode

When you are migrating from non-RLS to RLS access mode, to avoid potential AFCG abends, consider enabling the following feature toggle:

```
com.ibm.cics.rls.delete.ridfld=true
```

For details on why AFCG abends might occur and how this feature toggle can solve this issue, see [VSAM RLS](#).

Defining sharing control data sets

VSAM RLS requires sharing control data sets. These are used to maintain data integrity in the sharing environment. The sharing control data set is used sysplex-wide by all the SMSVSAM servers, and is always duplexed.

Two active, and at least one spare, sharing control data sets must be available at all times.

The size of these data sets depends on the number of MVS images in the sysplex and on the number of files that are expected to be open concurrently. For information about calculating the amount of space that is required for the sharing control data sets, see [z/OS DFSMSdfp Storage Administration](#).

Sharing control data sets are VSAM linear data sets that must be on volumes that have global connectivity. The data sets can have multiple extents, but only on the same volume. You define them using standard techniques for defining data sets. The names must have SYS1.DFPSHCDS as the first and second qualifiers. See [z/OS DFSMSdfp Storage Administration](#) for other rules relating to the definition of sharing control data sets.

You must not issue RESERVEs on any volumes that contain sharing control data sets. Convert any such RESERVEs to enqueue.

You can check that the data sets are available to the sysplex with the MVS DISPLAY SMS command, on any MVS image:

```
D SMS,SHCDS
```

This command shows the names of the two active data sets and the spare data set as shown in the following example:

Name	Size	%UTIL	Status	Type
ACTIVE1.VP2SS03 7920KB	7920KB	74%	Good	ACTIVE
ACTIVE2.VP2SS03 7920KB	7920KB	74%	Good	ACTIVE
SPARE.VP2SS03 7920KB	7920KB	74%	Good	SPARE

The DISPLAY command shows only the third and fourth qualifiers of the sharing control data set names; the first and second qualifiers are always SYS1.DFPSHCDS.

The first time an SMSVSAM server is started in the sysplex, you vary online the sharing control data sets with this command for the active data sets:

```
V SMS,SHCDS(shcdsname),NEW
```

Use this command for the spare data set or data sets.

```
V SMS,SHCDS(shcdsname),NEWSPARE
```

for the spare data set (or data sets). The server cannot start properly if you do not issue these commands.

Authorizing CICS user IDs for VSAM RLS support

Authorize each CICS user ID that is to use VSAM RLS support to have read access to a profile in the SUBSYSNM class that matches the APPLID.

See [“Authorizing access to an SMSVSAM server” on page 64](#) for more information.

You can restrict user access to the access method services SHCDS subcommands, AMS SHCDS LIST and REMOVE. See [z/OS DFSMS Access Method Services Commands](#).

Adding new parameters to SYS1.PARMLIB(IGDSMSxx)

To include RLS support in your system, specify the required parameters in the IGDSMSxx member of SYS1.PARMLIB.

- Specify RLSINIT(YES); otherwise, SMSVSAM does not initialize automatically when you IPL MVS. Alternatively, you can start SMSVSAM using the VARY SMS, SMSVSAM, ACTIVE command.
- Specify a value for the deadlock detection interval with the DEADLOCK_DETECTION parameter.
- Specify time intervals for the creation and synchronization of VSAM RLS SMF records with the CF_TIME and SMF_TIME parameters.
- Specify the maximum size of the SMSVSAM local buffer pool with the RLS_MAX_POOL_SIZE parameter.

See [z/OS DFSMSdfp Storage Administration](#) for information about these parameters.

Establishing new procedures for VSAM RLS support

You might require new operational procedures in a number of areas as a result of using VSAM RLS support, including integrity of data and management of coupling facilities and structures, use of RESERVEs, switching to non-RLS mode and forward recovery management.

- Integrity of data in coupling facility caches

To make sure that non-IBM products or user programs do not compromise the integrity of data in coupling facility caches when they modify the data on a volume, either vary the volume offline to each system in the sysplex, or CF-quietse the volume using the

```
V SMS,CFVOL(valid),QUIESCE
```

command before running such programs

- Management of the coupling facility and coupling facility structures
- Use of RESERVEs on volumes that contain sharing control data sets.

Make sure that RESERVEs are not used. Convert RESERVEs on other volumes into enqueues.

- Switching to non-RLS mode to run batch update jobs against recoverable data sets.
- Management of forward recovery and your forward recovery logs.

Forward recovery differs from forward recovery for non-RLS access:

- The forward recovery log stream must be in the ICF catalog.
- All forward recovery log records for a data set merge into the same log stream.
- Your forward recovery procedure must use the SHCDS FRSETRR, FRUNBIND, FRBIND, and FRRESETRR commands. CICS VSAM Recovery for z/OS automatically issues these commands.

Refer to the DFSMS/MVS documentation for more details.

Activating the coupling facility structures

After you have defined the main coupling facility lock structure and coupling facility structures in the CFRM policy, activate the structures using the SETXCF START POLICY command, specifying a TYPE of CFRM and the policy name.

Console messages

The message domain supports the use of MVS message routing codes in the range 1 to 16 for those messages that are sent to the console.

By default, if the issuing module specifies only CONSOLE (without a qualifying number) as the destination, CICS routes the message with MVS route codes 2 and 11. This support is available for all domain-type messages of the form DFHxxxxnn, where xx is the domain code and nnnn is the message number.

CICS issues other messages (of the form DFHnnnn) with either no route code or route codes other than 2 and 11.

The physical destination of these messages is controlled by the ROUTECODE parameter on the MVS console entries in a SYS1.PARMLIB member, CONSOLEnn. For further information about MVS console definitions, see the [z/OS MVS Initialization and Tuning Guide](#).

Defining the logger environment for CICS

CICS uses the MVS system logger for all its logging and journaling requirements.

Using services provided by the MVS system logger, the CICS log manager supports these logs:

- The CICS system log, which is used for these purposes:
 - Dynamic transaction backout
 - Warm and emergency restarts
 - Cold starts, but only if the log contains information required for resynchronizing indoubt units of work
- Forward recovery logs, autojournals, and user journals.

The MVS system logger is a component of MVS. It provides a programming interface to access records on a log stream. For information about the MVS system logger, see the following MVS publications:

- [z/OS MVS Setting Up a Sysplex for:](#)
 - General information about the MVS system logger.
 - Information about defining and formatting the LOGR couple data set.
 - Information about how to plan the system logger configuration, plan and set up a system logger application, and plan for recovery for system logger applications.
- [z/OS MVS Programming: Authorized Assembler Services Reference \(Volume 1\)](#) and [z/OS MVS Programming: Authorized Assembler Services Reference \(Volume 2\)](#) for the syntax of system logger services.
- [z/OS MVS Initialization and Tuning Reference](#) for information about the COUPLExx PARMLIB member.

Requirements planning and checklist for the logger environment

The requirements and the procedure to set up the CICS logging environment are outlined.

The procedure includes tasks for MVS system programmers, CICS system programmers, and security administrators, and close cooperation between all groups is required.

- Plan the logger environment.

Consider the possible storage options, and choose which of the three available hardware options to use:

- Nonvolatile coupling facility, where log stream data is duplexed in the MVS logger data space. Nonvolatile storage involves the use of battery backup or an uninterruptible power supply (UPS):
 - When using a UPS, you use a hardware console command to update coupling facility status.
 - When using battery backup, batteries must be online and charged.
- Volatile coupling facility, where log stream data is duplexed to a staging data set.
- DASD-only, where log stream data is duplexed in the MVS logger data space.

See [“Coupling facility or DASD-only?”](#) on page 135, to help you decide on one of these or a combination of both.

Make the following planning decisions:

- Determine the number CICS regions that require logger support and hence system log streams.
- Determine the number of user journals and autojournals that your regions use.
- Determine the number of forward recovery logs required for VSAM data sets.
- Determine whether any user journal or forward recovery log streams are to be shared between regions (to create merged data automatically). The system log streams, DFHLOG and DFHSHUNT, cannot be shared.

DASD-only log streams can be shared only in the same MVS image.

- Decide on the number and sizes of the coupling facilities to be used.

For information about types of coupling facility, see [z/OS MVS Setting Up a Sysplex](#). The minimum level of coupling facility supported by the MVS system logger is CFLEVEL=1, with the appropriate service level of the coupling facility control code that supports CFLEVEL=1.

- Determine the log stream sizes:
 - For coupling facility log streams, see [“Coupling facility log streams”](#) on page 136.
 - For DASD-only log streams, see [“DASD-only log streams”](#) on page 150.

- Apply any maintenance updates.

Ensure that all maintenance affecting the MVS system logger, and the CICS log manager and its utilities, is applied.

- Logger serviceability APARs that relate to the MVS system logger are identified with the LOGRSERVICE keyword.
- APARs that relate to the CICS log manager are identified with the CICSLOGR keyword.

- Create and format the LOGR couple data sets.

You need to know the number of log streams and, for coupling facility log streams, the number of structures. Each CICS region requires two system log streams and the following optional log streams:

- A log stream for the log of logs.
- One or more log streams for forward recovery logs.
- One or more log streams for autojournals.
- One or more log streams for user journals.

In consultation with your MVS system programmer, complete the following steps:

- Use MVS utility IXCL1DSU to create and format the primary and alternate LOGR couple data sets.
- Identify the LOGR couple data sets to the sysplex in the COUPLExx member in SYS1.PARMLIB.
- Make the LOGR couple data set available to the sysplex.

For more information, see ["Format the LOGR Couple Data Set and Make it Available to the Sysplex" in z/OS MVS Setting Up a Sysplex](#).

- Define the coupling facility structures.

If you are using the coupling facility for some or all of your log streams, update your CFRM policy and your LOGR couple data set with the required structure definitions. See ["Defining coupling facility structures" on page 136](#) for details, including a sample job.

- Establish the required security authorizations

Ensure that all the user IDs that are involved with running the system logger, or defining or accessing logger resources, are authorized, and that the required profiles are defined in the LOGSTRM general resource class.

- If the MVS system logger address space (IXGLOGR) is not given SAF privileged or trusted status, ensure that you give the required authorization to the user ID that runs IXGLOGR. For example, if the user ID that runs IXGLOGR (defined in the RACF started procedures table (ICHRIN03), or defined in the RACF STARTED class profile) is SYSTASK:
 - SYSTASK requires ALTER access to IXLSTR structure profiles in the FACILITY general resource class for access to log stream coupling facility structures.
 - SYSTASK requires ALTER access to the data set profiles (*hlq.data_set_name*) in the DATASET general resource class, for each DASD log stream and staging data set.
- To use the MVS system logger IXCMIAPU utility to define, update and delete entries in the LOGR couple data set, you must have appropriate authorizations to the relevant RACF profiles in the LOGSTRM and FACILITY general resource classes. See ["Authorizing users of IXCMIAPU" on page 65](#) for information and examples of how to do this.
- To enable CICS to create log streams dynamically, and to write to log streams, ensure that the CICS region user ID has the required authorizations. See ["Authorizations for CICS regions" on page 65](#) for information and examples of how to do this.

Ensure that all the user IDs that are involved with running the system logger, or defining or accessing logger resources, are authorized, and that the required profiles are defined in the LOGSTRM general resource class. For more information about authorizations for the system logger, see [z/OS MVS Setting Up a Sysplex](#).

- Check the sysplex definition in PARMLIB.

To use the MVS system logger, each MVS image must be a member of a sysplex. Ensure your sysplex definition, in PARMLIB member IEASYSxx, specifies either PLEXCFG(MONOPLEX), for a single-member sysplex, or PLEXCFG(MULTISYSTEM), for a multi-member sysplex. Also ensure that you define a COUPLExx, member in PARMLIB. The value specified on the SYSPLEX parameter in COUPLExx, forms part of DASD-only and staging data set names.

- Activate the LOGR subsystem.

Ensure that the LOGR subsystem is active so that the CICS log manager batch utility, DFHJUP, can format and print log data. The LOGR subsystem is defined by the following entry in IEFSSNxx PARMLIB member:

```
SUBSYS SUBNAME(LOGR) INITRTN(IXGSSINT)
```

- Plan staging data set requirements.

Staging data sets are used for both DASD-only and coupling facility log streams, and if specified are dynamically allocated by the MVS system logger:

- For DASD-only log streams, staging data sets are the primary (interim) storage.
- For coupling facility log streams, staging data sets are allocated by the system logger to safeguard log data in the event of the log data being in a volatile configuration; that is:
 - There is a loss of the coupling facility battery backup
 - A structure failure that results in the only copy of log data being in MVS local storage buffers.

Review the following parameters:

- STG_DUPLEX(YES) and DUPLEXMODE(COND) to cause the system logger to use staging data sets if the coupling facility is not failure independent (see [“Staging data sets for coupling facility logstreams”](#) on page 149 for more information).
 - STG_MGMTCLAS to specify the System Managed Storage (SMS) management class to be used for staging data set allocation (valid only when STG_DUPLEX(YES) or DASDONLY(YES) is specified)
 - STG_STORCLAS to specify the SMS storage class to be used for staging data set allocation (valid only when STG_DUPLEX(YES) or DASDONLY(YES) is specified).
 - STG_SIZE to specify the size of staging data sets.
 - SHAREOPTIONS(3,3) for log stream data sets and staging data sets (see "VSAM Share Options for System Logger" in [z/OS MVS Setting Up a Sysplex](#)).
- Plan DASD space and SMS environment for logger auxiliary storage.

System logger auxiliary storage comprises all log stream (offload) data sets. See [“Managing auxiliary storage”](#) on page 155 for information about size parameters and other attributes that relate to auxiliary storage.

- Define log streams and log stream models.

Define the specific log streams, and log stream models for dynamic creation of log streams, in the LOGR policy. Ensure that you complete the following steps:

- Set HIGHOFFLOAD no higher than 80% so that the offload function can be activated before structures reach the 90% level and to provide a buffer so that CICS can continue to write records without filling the logstream before offload completes.
- Set LOWOFFLOAD for DFHLOG and DFHSHUNT in the range 40–60%. For user journals and the log of logs, specify LOWOFFLOAD as 0.
- Specify HLQ for the high level qualifier for offload data sets. It is not part of the CICS log stream name. The default is IXGLOGR.
- Specify STG_DUPLEX(YES) and DUPLEXMODE(COND) for log streams in the coupling facility to ensure that staging data sets are used automatically if the coupling facility is volatile or failure dependent.
- Set STG_SIZE to control the size, in 4KB blocks, of staging data sets allocated by the system logger. For coupling facility log streams, the staging data set must hold at least as much data as the log stream in the structure, so that offloads are not triggered by the staging data sets. See [“Sizing coupling facility log streams”](#) on page 141 and [“Sizing for DASD-only log streams”](#) on page 152.
- Specify LS_DATACLAS and LS_SIZE, for the SMS data class and the number of 4KB allocation blocks respectively for log stream off load data sets. See [“Managing auxiliary storage”](#) on page 155.

- Specify MODEL(YES) to indicate that a log stream definition is a model only and not an actual log stream. See SDFHINST members DFHILG2 (coupling facility) and DFHILG5 (DASD-only) for samples of model log streams.

Use AUTODELETE(YES) with a suitable retention period (RETPD) for general logs but *not* for CICS system logs (DFHLOG and DFHSHUNT). See “[Defining coupling facility log streams](#)” on page 140 for some sample IXCMIAPU jobs, and [z/OS MVS Setting Up a Sysplex](#) for general information about updating LOGR policies.

- Define JOURNALMODEL resource definitions.

Define JOURNALMODEL resource definitions in the CICS system definition data set (CSD) to enable CICS to map CICS journal names to MVS system logger log stream names. See [JOURNALMODEL resources](#).

The MVS system logger log stream names must match the streamnames that are used in the JOURNALMODEL resource definitions, otherwise the following error messages might occur at CICS region startup:

- IXG231I IXGCONN REQUEST=CONNECT TO LOG STREAM *stream_name* DID NOT SUCCEED
- DFHLG0508 Log stream *stream_name* not defined to MVS
- DFHLG0731 A failure has occurred while opening the system log (DFHLOG).

- Review the **AKPFREQ** system initialization parameter.

This parameter represents the number of write operations (log records) by CICS log manager to the log stream buffer before an activity keypoint is taken, whereas under the old journal control program it specifies the number of consecutive blocks written to the system log data set.

The parameter has a significant effect on the size of system logger primary (interim) storage, affecting the log tail management that takes place during activity keypoint (AKP) processing. The system logger acts as follows:

- It deletes records that are no longer of interest to CICS
- It moves records to DFHSHUNT for those tasks that wrote log records in the last ACP.

- Evaluate the results after implementation.

After you implemented the procedure to use the MVS system logger for CICS log streams and journals, evaluate the results on a continual basis. You can use the following:

- CICS interval statistics. You can collect these at specified intervals and end-of-day to obtain CICS log manager statistics. You can also collect statistics using the DFHOSTAT sample program.
- SMF Type 88 records. These are produced by the MVS system logger, and can be printed using IXGRPT1, which is supplied in SYS1.SAMPLIB. You can also print these records using IXGRPT1J and IXGRPT1L.

Setting up the environment for CICS log manager

CICS system programmers must consult with their MVS system programmers to plan for the storage that is required by the log streams required by the many CICS log managers operating in the sysplex.

Each log stream is a sequence of blocks of data, which the MVS system logger internally partitions over three different types of storage:

1. Primary storage, which holds the most recent records that were written to the log stream. Primary storage can consist of either of these areas:
 - a. A structure in a coupling facility. Log data written to the coupling facility is also copied to either a data space or a staging data set.
 - b. A data space in the same MVS image as the system logger. Log data written to the data space is also copied to a staging data set.
2. Auxiliary storage. When the primary storage for a log stream becomes full, the older records automatically spill into auxiliary storage, which consists of data sets managed by the storage

management subsystem (SMS). Each log stream, identified by its log stream name (LSN), is written to its own log data sets.

3. Tertiary storage. A form of archive storage that is used as specified in your hierarchical storage manager (HSM) policy. Optionally, older records can be migrated to tertiary storage, which can be either DASD data sets or tape volumes.

See the different levels of log stream storage in [Figure 7 on page 134](#) and [Figure 8 on page 135](#).

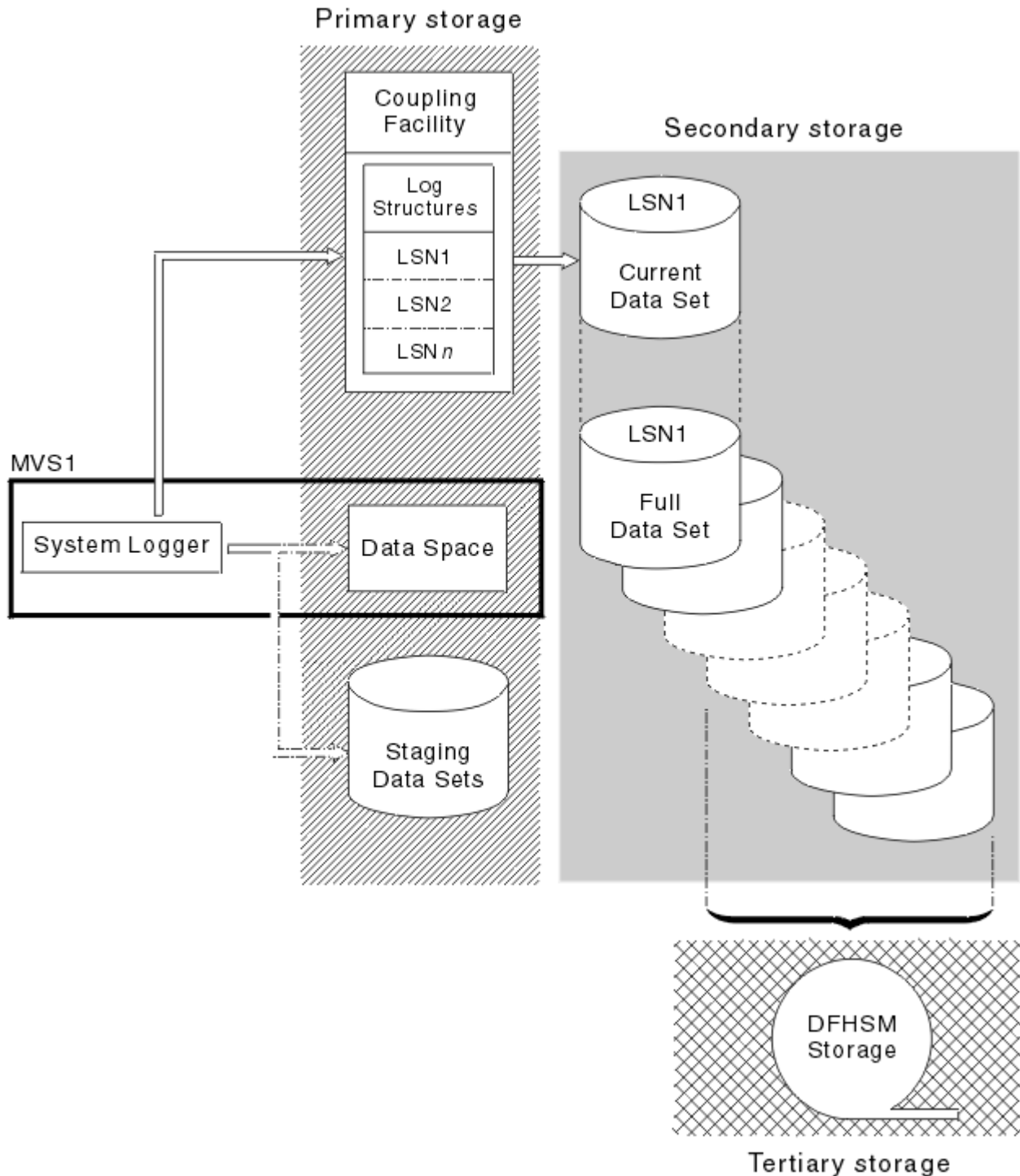


Figure 7. The types of storage used by the MVS system logger

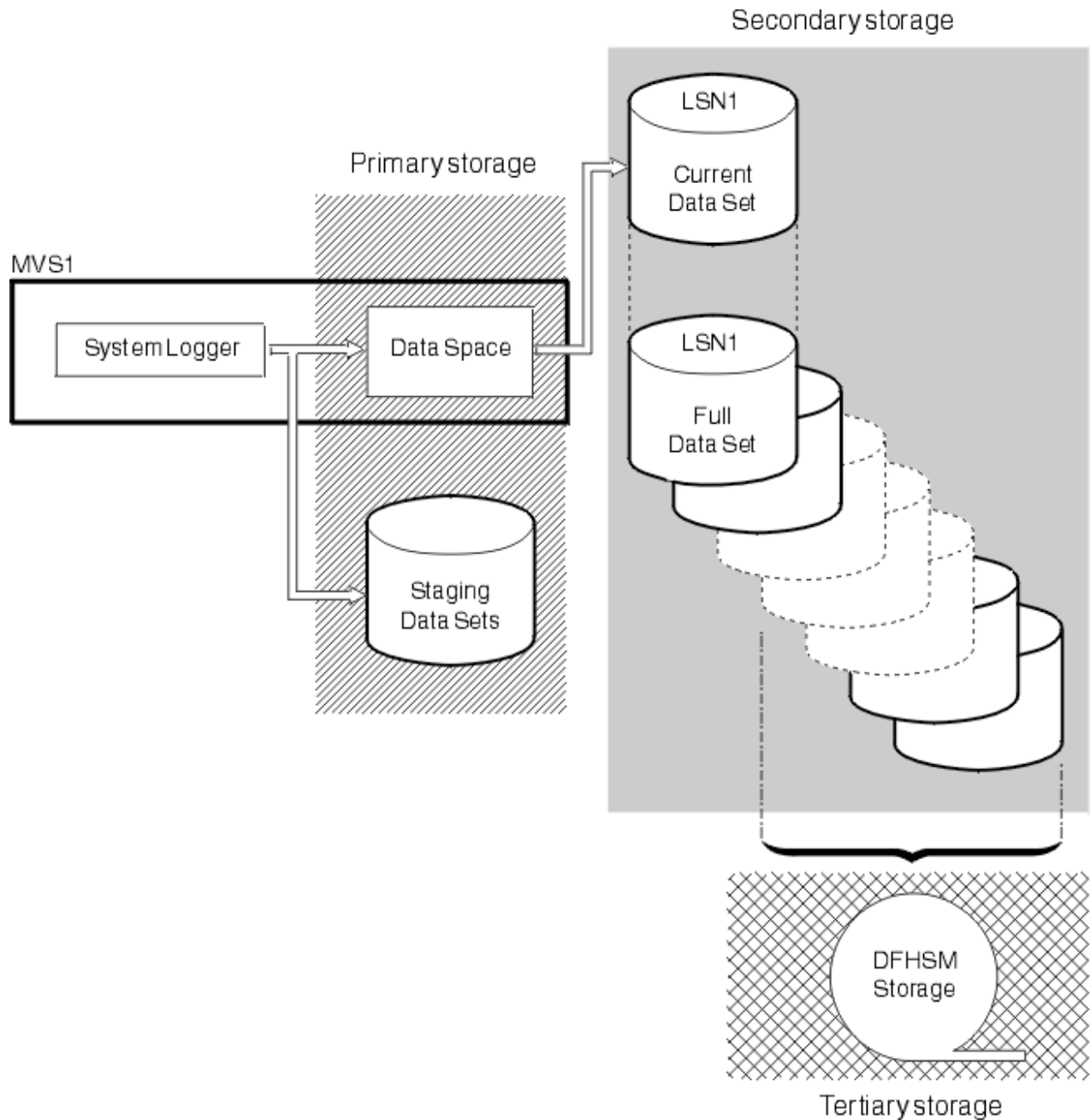


Figure 8. The types of storage used by the MVS system logger

Coupling facility or DASD-only?

The CICS log manager supports the DASD-only option of the MVS system logger. Individual CICS log streams can use either coupling facility log structures or DASD-only logging.

Take the following points into account when deciding which log streams you define to use the coupling facility and which to use DASD-only:

- A coupling facility log stream must be used if you want to allow simultaneous access from CICS regions running in different MVS images. (Simultaneous access to a DASD-only log stream is limited to CICS regions in the same MVS image.)

For example, assume that you are using RLS and have several CICS application-owning regions (AORs) running on different MVS images. Because the forward recovery log must be accessible from all the AORs, it must be defined as a coupling facility log stream. A CICS system log, on the other hand, is accessed only by a single CICS region, and can therefore always be defined as a DASD-only log stream.

Without a coupling facility, you cannot share general log streams across MVS images.

- For optimum availability, avoid defining all your CICS log streams to use structures in a single coupling facility. See [“Coupling facility log streams” on page 136](#) for more information.
- DASD-only log streams are easier to define and administer than coupling facility log streams.
- The CPU cost of a log write to a DASD-only log stream is greater than that of a write to a coupling facility log stream.
- If the amount of available coupling facility space is limited, you might want to define some DASD-only log streams to minimize the amount of space allocated to log structures.

Note that you can define a single-system sysplex, which must use a sysplex couple data set, with PLEXCFG=MONOPLEX. This definition is required for stand-alone MVS systems that use MVS system logger facilities. Also, define sysplexes that have two or more MVS images with PLEXCFG=MULTISYSTEM.

Coupling facility log streams

If you use a coupling facility, the most suitable environment is provided by two or more non-volatile coupling facilities that are failure-independent from any of the exploiting MVS images, using dedicated processor resources.

If one coupling facility fails, or requires maintenance, in such an environment, the system logger can rebuild its data in another coupling facility and continue. CICS systems that are running are minimally affected.

To define a higher availability Sysplex, review [z/OS MVS Setting Up a Sysplex](#).

If possible, do not run with a single coupling facility because its failure can cause the MVS system logger, and any other users of the coupling facility, to suspend normal operation until access to the coupling facility is restored. CICS is effectively unusable in such a situation.

Unless you specify that the system logger is to use staging data sets, the recovery of log stream data depends on the MVS images remaining active so that the system loggers can use copies of log records held in storage to repopulate the coupling facility when it is again available. If you must run with a single coupling facility, specify DUPLEXMODE(UNCOND) to force the use of staging data sets.

Defining coupling facility structures

If you use a coupling facility for your CICS log streams, define the coupling facility structures you require for the log streams in your CFRM policy, in the CFRM data set, and in the LOGR policy, in the LOGR data set.

Updating the CFRM policy

Coupling facility space is divided into structures using the coupling facility resource management (CFRM) policy defined in the CFRM data set. The CFRM policy allows you to define how MVS is to manage coupling facility resources, and you update this policy using the IXCMIAPU utility. See [Figure 9 on page 137](#) for a sample job to define coupling facility structures in the CFRM policy data set.

Updating the LOGR policy

You define structures in the MVS system logger LOGR policy in the system logger couple data sets using the DEFINE STRUCTURE specification of the IXCMIAPU utility. See [Figure 10 on page 138](#) for a sample job to define coupling facility structures in the LOGR policy data set.

Before attempting to run any of the IXCMIAPU jobs, ensure that the MVS system logger, IXGLOGR, is running. If IXGLOGR is not running, for example, if MVS is running in LOCAL mode, logstream definition jobs fail with rc=0814.


```

//CFRM      JOB (accounting-information),CLASS=A,MSGCLASS=A
//POLICY    EXEC PGM=IXCMIAPU
//STEPLIB   DD DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT  DD SYSOUT=*
//*****
/*
/* System logger structure definitions should be merged with
/* definitions for other structures required by the sysplex.
/*
/* Space values are for illustration only -- substitute values
/* appropriate to your number of logs and expected activity.
/*
/* NOTE: The values in this job are not matched with the other
/* sample jobs.
/*
//*****
//SYSIN     DD *
DATA TYPE(CFRM) REPORT(YES)
DELETE POLICY NAME(POL1)
DEFINE POLICY NAME(POL1)
    PREFLIST(cfname) REBUILDPERCENT(1)
/* Define coupling facilities */
CF NAME(cfname)
    TYPE(002097)
    MFG(IBM)
    PLANT(00)
    SEQUENCE(000000040032)
    PARTITION(3)
    CPCID(00)
    DUMPSPACE(8192)
/* Define other structures required by sysplex here . . . */
...
/* Define logger structures for CICS log streams.
/* - If a choice of facilities is available, use non-volatile
/* facilities if possible
/* - Specify low REBUILDPERCENT so that structures are rebuilt
/* in the event of connectivity failure
/* - INITSIZE gives initial coupling facility size (based on sizing calcs)
/* - SIZE should be larger to allow for rebuild to a larger
/* size if INITSIZE proves to be too small
/* SIZE and INITSIZE values are for illustration only -
/* substitute values appropriate for your intended usage.
/*

STRUCTURE NAME(LOG_DFHL0G_001) /* CICS system logs */
INITSIZE(10000) SIZE(16000)
PREFLIST(cfname) REBUILDPERCENT(1)

STRUCTURE NAME(LOG_DFHSHT_001) /* CICS secondary logs */
INITSIZE(10000) SIZE(16000)
PREFLIST(cfname) REBUILDPERCENT(1)

STRUCTURE NAME(LOG_USERJNL_001) /* CICS user journals */
INITSIZE(10000) SIZE(16000)
PREFLIST(cfname) REBUILDPERCENT(1)

STRUCTURE NAME(LOG_GENERAL_001) /* Forward recovery logs */
INITSIZE(10000) SIZE(16000)
PREFLIST(cfname) REBUILDPERCENT(1)

STRUCTURE NAME(LOG_CICSVR) /* CICS VR logs */
INITSIZE(10000) SIZE(16000)
PREFLIST(cfname) REBUILDPERCENT(1)
/*
//

```

Figure 9. Sample policy job to define logger structures to CFRM 1/2

Multiple log streams can write data to a single coupling facility structure. The log data is not merged; the log data stays separated according to log stream. You can specify the number of log streams that use the resources of a single coupling facility structure using the LOGSNUM parameter on the IXCMIAPU service to define a structure.

Each log stream is allocated a proportion of the structure space based on the number of currently connected log streams, up to the limit specified in LOGSNUM.

For example, you might define a structure to contain a maximum of, say, 30 log streams. If only 10 log streams are connected, each log stream can use one tenth of the space in the structure. As other log

streams are connected and disconnected, the MVS system logger adjusts the proportion of space to be used by each log stream.

You must plan carefully before specifying a value for LOGSNUM, because this parameter determines how much storage space in the structure is available to each log stream. A number in the range 10 to 20 is optimum in many environments.

The JCL in [Figure 10 on page 138](#) defines log stream coupling facility structures to the MVS system logger. It is meant for guidance only and you must substitute values appropriate to your requirements.

```
//DEFSTRUC JOB ...
//POLICY EXEC PGM=IXCMIAPU
//STEPLIB DD DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//*****
//*
//* Define log stream coupling facility structures to the MVS logger *
//*
//* AVGBUFSIZE and LOGSNUM values are just for illustration,      *
//* substitute values appropriate to your intended usage          *
//*
//*****
//SYSIN DD *
DATA TYPE(LOGR) REPORT(YES)

/* System logs */
DEFINE STRUCTURE NAME(LOG_DFHLOG_001) LOGSNUM(10)
MAXBUFSIZE(64000) AVGBUFSIZE(500)

/* Secondary system logs */
DEFINE STRUCTURE NAME(LOG_DFHSUNT_001) LOGSNUM(10)
MAXBUFSIZE(64000) AVGBUFSIZE(4096)

/* User journals with unforced writes */
DEFINE STRUCTURE NAME(LOG_USERJRNL_001) LOGSNUM(10)
MAXBUFSIZE(64000) AVGBUFSIZE(64000)

/* Fwd recovery logs and user jnl's that are forced */
DEFINE STRUCTURE NAME(LOG_GENERAL_001) LOGSNUM(10)
MAXBUFSIZE(64000) AVGBUFSIZE(2048)

/*
//
```

Figure 10. Sample JCL to define coupling facility structures to MVS system logger

For information about planning coupling facility configurations, see [z/OS MVS Programming: Assembler Services Guide](#).

Planning for the number of log structures

There are a number of points to consider when planning the definition of your coupling facility structures.

- The CFRM policy allows a maximum of 255 structures for all purposes.
- Allow a maximum of 512 log streams per structure.
- Smaller structures are more quickly allocated, rebuilt, and recovered than larger ones.
- It is good practice to keep the log streams for test CICS systems and other systems not in regular use in structures separate from the structures holding the log streams of production CICS systems. This separation avoids the structure space available to production CICS systems being affected by structure usage of the test CICS systems.
- It is good practice to keep the log streams for terminal-owning regions (TORs) in structures separate from those accommodating log streams for application-owning regions (AORs). In addition, keep log streams for file-owning regions in structures separate from those accommodating log streams for TORs and AORs.
- Share structures between MVS images. If an MVS image or logger address space fails, and a surviving MVS image is using the same log stream structures, although not necessarily the same log streams, the surviving image is notified of the failure and can initiate immediate log stream recovery for the failing MVS.

Recovery, otherwise, is delayed until the next time that a system attempts to connect to a log stream in the affected structures, or until the logger address space of the failing system is restarted.

For example, in a 4-way sysplex comprising MVSA, MVSB, MVSC, and MVSD, you might have the CICS regions that normally run on MVSA and MVSB use structure LOG_DFHLOG_001, and the regions that run on MVSC and MVSD use structure LOG_DFHLOG_002. Thus, each MVS image has a partner to recover its log streams if MVS fails. If a structure fails, the two MVS images using the other structure can take over the workload. Also, if you have more than one coupling facility, allocate the system log structures to different coupling facilities. See Figure 11 on page 139 for an illustration of this example.

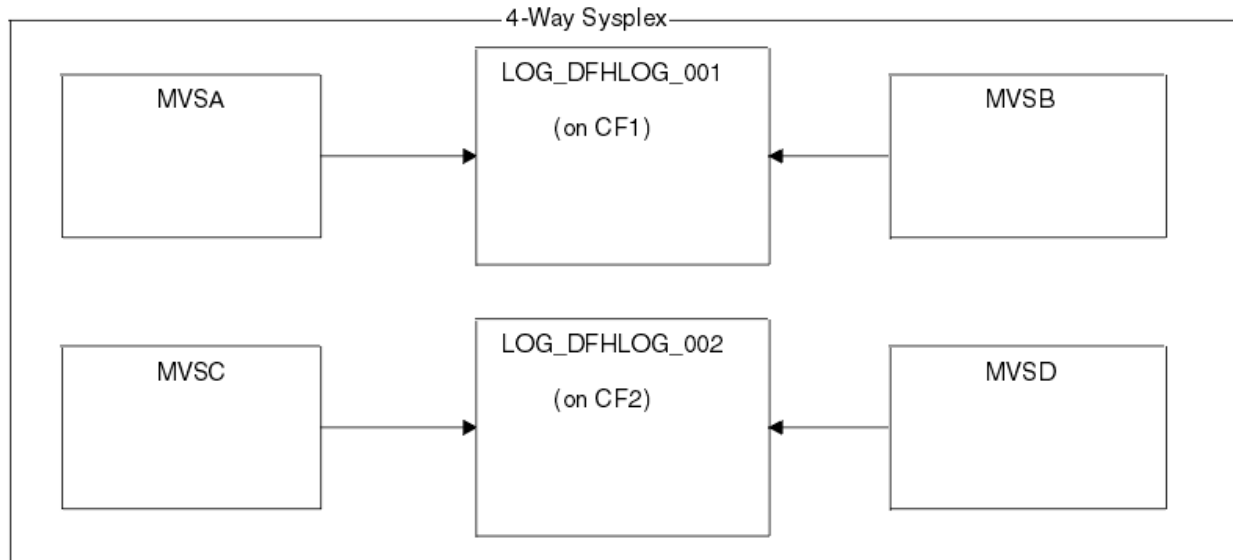


Figure 11. Sharing system logger structures between MVS images

- Use the appropriate buffer size. The average buffer size (AVGBUFSIZE) defined for a structure must be close to the actual buffer size of the log streams using the structure. If it is not, usable space might be exhausted long before the structure is full.

Important:

1. z/OS dynamically tunes the element and entry ratio, so the value you specify for AVGBUFSIZE is less important than it was on earlier releases of MVS.
 2. You cannot update AVGBUFSIZE, like other structure definition attributes such as MAXBUFSIZE and LOGSNUM, unless you first delete the log streams in the structure definition.
- Set MAXBUFSIZE to slightly less than 64 KB; for example, 64 000 is suitable. This number allows CICS to write the maximum size user record and allows coupling facility storage to be allocated in 256-byte units. If you allow MAXBUFSIZE to default, coupling facility storage is allocated in 512-byte units. This size can be wasteful of storage.

If you set MAXBUFSIZE lower than 64 000 you gain no significant advantage as far as the use of storage is concerned.

- Set a low value for the REBUILDPERCENT parameter in the CFRM policy for log structures used for CICS system logs.

Log structure naming conventions

Adopt a naming convention for your coupling facility structures that helps to identify the purpose of the structure.

Use a format such as LOG_purpose_nnn:

- *purpose* identifies the type of use of the structure.
- *nnn* is a sequence number to allow for more than one structure for each purpose.

Here are some examples:

LOG_DFHLOG_001

For the CICS primary system log. The structure must be large enough to avoid writing data to DASD. The average buffer size is small. See the sizing calculations in [“Structure size for system log usage”](#) on page 143.

LOG_DFHSUNT_001

For the CICS secondary system log. The structure must be small, but requires a large buffer size. A structure of 150 KB per log stream might be sufficient.

LOG_USERJRN_001

For user journals where block writes are not forced. The average and maximum buffer sizes of these structures must be the same.

LOG_GENERAL_001

For forward recovery logs and user journals in which block writes are forced periodically.

See also [Develop a naming convention for system logger resources in z/OS MVS System Management Facilities](#).

Defining coupling facility log streams

Use the MVS IXCMIAPU utility to define coupling facility log streams to the LOGR couple data set.

The following code sample shows the basic syntax to define a coupling facility log stream:

```
DEFINE LOGSTREAM NAME(log_stream_name)
    STRUCTNAME(structure_name)
    LOWOFFLOAD(low_offload) HIGHOFFLOAD(high_offload)
    STG_DUPLEX(YES|NO) DUPLEXMODE(COND|UNCOND)
```

For detailed information about the full range of log stream attributes, see the [z/OS MVS Setting Up a Sysplex](#) manual.

Figure 12 on page 140 shows example definitions for a pair of coupling facility log streams associated with a CICS system log. The definitions are for the CICS primary and secondary system log streams. The `region_userid` value is the RACF user ID under which the CICS address space is running. The `applid` value is the z/OS Communications Server APPL name of the CICS region (taken from the **APPLID** system initialization parameter).

The primary and secondary log streams are placed in different structures because of the large disparity in data volumes written to the primary and secondary system logs.

```
//DEFLOGS JOB ...
//LOGDEFN EXEC PGM=IXCMIAPU
//STEPLIB DD DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//*****
//*
//* Define coupling facility log streams for CICS system log.
//*
//* The LOWOFFLOAD value is for illustration only --
//* substitute a value appropriate for your environment.
//*
//*****
//SYSIN DD *
DATA TYPE(LOGR) REPORT(NO)
DEFINE LOGSTREAM NAME(region_userid.applid.DFHLOG)
    STRUCTNAME(LOG_DFHLOG_001)
    LOWOFFLOAD(40) HIGHOFFLOAD(80)
    STG_DUPLEX(YES) DUPLEXMODE(COND)
DEFINE LOGSTREAM NAME(region_userid.applid.DFHSUNT)
    STRUCTNAME(LOG_DFHSUNT_001)
    LOWOFFLOAD(40) HIGHOFFLOAD(80)
    STG_DUPLEX(YES) DUPLEXMODE(COND)
```

Figure 12. Example definitions of coupling facility log streams

Using model log streams

To avoid defining explicitly each log stream used by each of your CICS regions, you can use model log stream definitions.

Using models, log streams are defined to MVS dynamically, on their first usage. [Figure 13 on page 141](#) shows an example of coupling facility model definitions for CICS primary and secondary system log streams.

```
//DEFLOGS JOB ...
//LOGDEFN EXEC PGM=IXCMIAPU
//STEPLIB DD DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//*****
//*
//* Define coupling facility model log streams for CICS system log.*
//*
//* The LOWOFFLOAD value is for illustration only --
//* substitute a value appropriate for your environment.
//*
//*****
//SYSIN DD *
DATA TYPE(LOGR) REPORT(NO)
DEFINE LOGSTREAM NAME(sysname.DFHLOG.MODEL)
MODEL(YES)
STRUCTNAME(LOG_DFHLOG_001)
LOWOFFLOAD(40) HIGHOFFLOAD(80)
STG_DUPLEX(YES) DUPLEXMODE(COND)
DEFINE LOGSTREAM NAME(sysname.DFHSHUNT.MODEL)
MODEL(YES)
STRUCTNAME(LOG_DFHSHUNT_001)
LOWOFFLOAD(40) HIGHOFFLOAD(80)
STG_DUPLEX(YES) DUPLEXMODE(COND)
```

Figure 13. Example model definitions for coupling facility system log streams

For detailed information about using model log streams, see [Model log streams for CICS general logs](#). For information about the mapping of CICS journal definitions to log stream names, see [Mapping of general log streams](#).

When using model log streams, note these points:

- For coupling facility log streams, a model log stream definition determines the coupling facility structure in which the new log streams are created. On an MVS image that runs both CICS production and CICS test regions, take care that the system logs for the production regions are kept separate from the system logs for the test regions.
- You must take into account recovery when using model log streams to define CICS system logs. See [Recovery considerations](#) for more information.

Sizing coupling facility log streams

You must work out the sizes of the CICS primary and secondary system log streams, forward recovery logs, and user journals and autojournals.

- Primary system log stream.

Use the following guidelines when you size CICS primary system log stream, DFHLOG:

- Minimize the amount of data that is offloaded to auxiliary storage.

The MVS system logger begins the offload process when the high offload threshold (HIGHOFFLOAD) of the log stream is reached. The offload process has two steps:

1. The MVS logger physically deletes the data in the log stream that the CICS log tail deletion process has marked for deletion.
2. The MVS logger calculates how much data must be offloaded to auxiliary storage, based on the difference between HIGHOFFLOAD and LOWOFFLOAD, less the amount of data that was deleted since the last offload event.

To minimize the amount of data offloaded from the CICS primary system log:

- Define a suitably sized coupling facility structure.
- Ensure that the log tail deletion process is working effectively. For detailed information about the log tail deletion process, see [Log tail deletion](#).

– Avoid *structure full* events.

A structure full event occurs when the structure space of a log stream becomes full before the offloading of data has completed. For information about monitoring and avoiding structure full events, see [CICS logging and journaling](#).

- Secondary log stream.

You must size the secondary system log stream, DFHSHUNT, to avoid structure full events. However, typically some data is offloaded from DFHSHUNT to auxiliary storage.

- General logs.

You must size forward recovery logs, user journals, and autojournals to avoid structure full events. However, because CICS does not delete data from these log streams, data is typically offloaded to auxiliary storage.

Setting attribute values for structure, log stream and system definitions

You can use the z Systems Coupling Facility Structure Sizer tool (CFSizer) to calculate storage requirements for coupling facility log streams. CFSizer is a web-based application that communicates with a coupling facility at a current CFLEVEL to calculate storage requirements. See [System z Coupling Facility Structure Sizer Tool \(CFSizer\) web page](#).

You can use the CFSizer tool to calculate the following values:

- INITSIZE
- SIZE

For more information, see [Structure size for system log usage](#).

For information about the CPC support for different CFLEVELs and the function in each CFLEVEL, see [CF levels](#).

You can use the formulas provided in [Structure size for system log usage](#) to calculate the following values:

- INITSIZE
- AVGBUFSIZE
- SIZE
- LOWOFFLOAD

You can use the formula provided in [Staging data set size calculation](#) to calculate the STG_SIZE value.

You must base your calculations on the journaling requirements of your applications.

Table 13 on page 142 summarizes how you decide on the values for various attributes on the structure definition, log stream definition, and system definition.

Table 13. How to decide on the values of attributes		
Facility	Attribute	Method or Value
Structure	INITSIZE	Use CFSizer or the formula in INITSIZE calculation .
	SIZE	Use CFSizer or the formula in SIZE calculation .
	AVGBUFSIZE	Use the formula in AVGBUFSIZE calculation . Underestimate rather than overestimate.
	MAXBUFSIZE	64 000

Table 13. How to decide on the values of attributes (continued)

Facility	Attribute	Method or Value
Primary system log stream (DFHLOG)	HIGHOFFLOAD	80
	LOWOFFLOAD	Use the formula in LOWOFFLOAD calculation.
Secondary system log stream (DFHSHUNT)	HIGHOFFLOAD	80
	LOWOFFLOAD	0
General log stream	HIGHOFFLOAD	80
	LOWOFFLOAD	40 – 60
Log stream	STG_SIZE	Use the formula in Staging data set size calculation . Overestimate rather than underestimate.
CICS system	AKPFREQ	4 000

Startup might take longer than you experienced when using earlier releases. This is partly caused by the allocation and formatting of the staging data sets. Increased startup time depends on factors that include the following:

- Size of staging data set (STG_SIZE)
- DASD speed
- DASD contention

You can reduce startup time by avoiding large staging data sets.

Structure size for system log usage

You can use the z Systems Coupling Facility Structure Sizer tool (CFSizer), or the formulas provided, to calculate storage requirements for the primary system log, DFHLOG, and the secondary system log, DFHSHUNT.

The CFSizer calculations assume that the coupling facility is at the current CFLEVEL. The formulas provided assume the coupling facility is at CFLEVEL 11.

You use the results of these calculations to set the coupling facility resource manager (CFRM) **INITSIZE** and **SIZE** parameters in the CFRM policy.

The primary and secondary log streams must be placed in different structures because of the large disparity in data volumes written to the primary and secondary system logs.

Generally, the volume of data that CICS keeps in the primary system log at any one time is slightly greater than the amount written during one activity keypoint interval. This volume is determined by the activity keypoint frequency, which is measured in the number of write requests to the CICS system logstream output buffer, and defined on the **AKPFREQ** system initialization parameter. When you plan coupling facility structure sizes, review the value specified for the **AKPFREQ** parameter.

Using CFSizer to determine structure size

CFSizer is a web-based application that communicates with a coupling facility at a current CFLEVEL to calculate storage requirements. For primary and secondary system logs, it calculates suitable values for the **INITSIZE** and **SIZE** parameters. See [System z Coupling Facility Structure Sizer Tool \(CFSizer\) web page](#).

To use CFSizer to calculate storage requirements for system logs, enter the following information:

Maximum buffer size

The size, in bytes, of the largest log block that can be written to a logstream. The value must be between 1 and 65532. This value corresponds to the MAXBUFSIZE value in the definition of the logstream coupling facility structure. See [“Defining coupling facility structures”](#) on page 136.

Average buffer size

The average size in bytes of log blocks written to all logstreams. This input is used to determine the initial entry to element ratio of the list structure. This value corresponds to the AVGBUFSIZE value in the definition of the logstream coupling facility structure. See [“Defining coupling facility structures”](#) on page 136.

Also, this value is available in the system logger reports produced by IXGRPT1 and IXGRPT1J. See [“Analyzing system logger data \(SMF Type 88 records\)”](#) on page 154.

Highoffload

The high offload threshold of the logstream. When this percentage of the coupling facility structure is full, the system logger begins to offload data from primary storage to offload data sets. For CICS logstreams, this value is usually between 80 and 85. See [Checking logstream status](#) and [LOWOFFLOAD](#) and [HIGHOFFLOAD](#) parameters on log stream definition.

For more information about the offload threshold parameter, see [z/OS MVS Setting Up a Sysplex](#).

Number of write requests per interval

The number of log blocks written to the logstream (the total from all systems that use the logstream) during the data collection interval. This value is available in the CICS statistics reports. See [Logstream statistics](#) and [Logstreams reports](#).

Also, this value is available in the #WRITES INVOKED data in the system logger reports produced by IXGRPT1 and IXGRPT1J. See [“Analyzing system logger data \(SMF Type 88 records\)”](#) on page 154.

Length of interval in seconds

The time interval in seconds over which the number of writes was collected. This value is available in the CICS statistics reports. See [Statistics domain: Summary global statistics](#)

Longest running task in seconds

The duration in seconds of the longest running transaction during the data collection interval. This is not the average response time.

Log tail deletes

The number of times CICS performed log tail deletion during the data collection interval. This value is available in the CICS statistics reports. See [delete requests](#), [stream deletes](#) or [logstream deletes](#) in [Logstream statistics](#) and [Logstreams reports](#). You can also determine this value by using joblog of the region and counting the number of DFHLG0743 messages that were issued during the interval.

Calculation of INITSIZE for DFHLOG

To calculate the INITSIZE value to use in the CFRM policy for the primary system log, use the following formula:

$$\text{INITSIZE} = 310 + (\text{LOGSNUM} * A * B / 1024)$$

where:

- $A = 2000 + (\text{number_of_entries} + 5)$
- $B = (\text{AVGBUFSIZE} * 1.1289) + 195$

To calculate a value for `number_of_entries`, use the following formula:

$$\text{number_of_entries} = ((\text{akpintvl} + \text{trandur}) * \text{writespersec}) / 0.9$$

where:

- `akpintvl` is the interval between activity key points, which varies with workload. Calculate it as follows:

$$\text{akpintvl} = \text{AKPFREQ} / ((N1 * R1) + (N2 * R2) + (Nn * Rn))$$

where:

- $N1, N2, \dots, Nn$ is the transaction rate for each transaction (transactions per second).

- R1, R2 Rn is the number of log records written by each transaction.
- `trandur` is the execution time (between sync points) of the longest-running transaction that runs as part of the normal workload.

If this duration is longer than the `akpintvl` value, you have the following choices:

- Increase the value of the **AKPFREQ** parameter, thus increasing the value of `akpintvl`, provided that the value does not result in an unacceptably large coupling facility structure size.
- Change the application logic to cause more frequent sync points.
- Calculate a structure size based on a shorter transaction duration, and accept that DASD offloading occurs when the long-running transaction is used.
- `writespersec` = lesser of 25 or $((N1 * R1) + \dots (Nn * Rn))$, where:
 - N1, N2 Nn are the transaction frequencies (transactions per second) of the most frequently run transactions.
 - R1, R2 Rn is the number of log records written by each transaction.

To calculate the value of `AVGBUFSIZE` for the primary system log, `DFHLOG`, you can use the weighted average of the data logged by the most frequently run transactions in the system:

$$\text{AVGBUFSIZE} = (\text{bytespersec} / \text{writespersec}) + 48$$

where:

- `bytespersec` = $(N1 * D1) + (N2 * D2) + \dots (Nn * Dn)$
 - N1, N2 Nn are the transaction frequencies (transactions per second) of the most frequently run transactions.
 - D1, D2 Dn are the bytes of data logged by each transaction.

You can calculate the amount of data (Dn) written to the system log for each transaction:

$$Dn = Ns * \text{syncrclen} + Nfc * (\text{fcrechdr} + \text{fcreclen}) + Nts * (\text{tsrechdr} + \text{tsreclen}) + Ntd * (\text{tdrechdr} + \text{tdreclen}) + Nur * (\text{urrechdr} + \text{urreclen})$$

where:

- Ns is the number of sync points per transaction; usually 1.
- `syncrclen` is the sync point record length.
- Nfc, `fcrechdr`, `fcreclen` are the number of recoverable updates made, the length of the record headers, and the length of the records for file control, respectively.
Count only READ UPDATE and WRITE ADD records. `fcrechdr` is 144 (136 bytes of record header plus 8 bytes of file name).
- Nts, `tsrechdr`, `tsreclen` are for recoverable temporary storage updates.
Count only TS PUT and TS UPDATE records. For TS PUT records, `tsrechdr` is 108, and `tsreclen` is 88. For TS UPDATE records, `tsrechdr` is 108, and `tsreclen` is 52.
- Ntd, `tdrechdr`, `tdreclen` are for recoverable transient data updates.
`tdrechdr` is 108, and `tdreclen` is 380.
- Nur, `urrechdr`, `urreclen` are for user records written to `DFHLOG`.
`urrechdr` is 125.
- See [Writes per second calculation](#) for details of how to calculate `writespersec`.

If the result of the calculation shows a value for AVGBUFSIZE that is greater than the value defined for MAXBUFSIZE, the value defined for MAXBUFSIZE is taken as the value for AVGBUFSIZE, and writespersec is calculated as follows:

$$\text{writespersec} = \text{bytespersec} / (\text{MAXBUFSIZE} - 48)$$

Round the final result of the INITSIZE formula up to the next multiple of 256.

Calculation of SIZE for DFHLOG

To calculate the SIZE value to use in the CFRM policy for the primary system log, use the following formula:

$$\text{SIZE} = 480 + (\text{LOGSNUM} * \text{A} * \text{B} / 1024)$$

where:

- $\text{A} = 2500 + (\text{number_of_entries} + 5)$
- $\text{B} = (\text{AVGBUFSIZE} * 1.6821) + 289$

To calculate a value for number_of_entries, see [number of entries calculation](#).

Round the final result of the SIZE formula up to the next multiple of 256. The formula for SIZE gives a result that is approximately fifty percent greater than the INITSIZE value.

Calculation of LOWOFFLOAD for DFHLOG

You can calculate a suitable value for LOWOFFLOAD for DFHLOG using one of the following formulas:

- $\text{LOWOFFLOAD} = (\text{trandur} * 90) / (\text{akpintvl} + \text{trandur}) + 10$ (where RETPD=0 specified)
- $\text{LOWOFFLOAD} = (\text{trandur} * 90) / (\text{akpintvl} + \text{trandur})$ (where RETPD=dddd specified)

where:

- akpintvl is the interval between activity key points. See [activity key point interval calculation](#).
- trandur is the execution time, between sync points, of the longest-running transaction that runs as part of the normal workload.

If this duration is longer than akpintvl value, you have the following choices:

- Increase the value of the **AKPFREQ** parameter, thus increasing the value of akpintvl, provided that this value does not result in an unacceptably large coupling facility structure size.
- Change the application logic to cause more frequent sync points.
- Calculate a structure size based on a shorter transaction duration, and accept that DASD offloading occurs when the long-running transaction is used.

For a description of the low offload threshold for a logstream, see [LOWOFFLOAD and HIGHOFFLOAD parameters on log stream definition](#).

Calculation of INITSIZE and SIZE for DFHSHUNT

Generally, the secondary system logstream is only a fraction of the size of the primary logstream. To calculate coupling facility space for DFHSHUNT, use the following formulas:

- $\text{INITSIZE} = (150 * \text{LOGSNUM}) + 310$
- $\text{SIZE} = (230 * \text{LOGSNUM}) + 480$

where LOGSNUM is number of logstreams that can write data to the secondary logstream.

Structure size for forward recovery log usage

You can merge the forward recovery logs written by many CICS regions onto the same log stream. You can also use the same log stream for forward recovery data for multiple data sets.

You can calculate a value for number of entries for the INITSIZE and SIZE attributes in the following way:

```
no. entries = writespersec * 12.5
```

where:

```
writespersec = lesser of 25 or (N1 + ... + Nn)
```

where N1 Nn is the number of transactions per second writing to each data set.

You can calculate AVGBUFSIZE in the following way:

```
AVGBUFSIZE = (bytespersec / writespersec) + 36
```

where:

- bytespersec = (N1 * Wr1 * (D1 + rechdr) +.. (Nn * Wn * (Dn + rechdr)))
- writespersec = lesser of 25 or (N1 + ... + Nn), where:
 - N1 Nn is the number of transactions per second writing to each data set.
 - Wr1 . . . Wn is the number of write requests per transaction.
 - D1 Dn is the average record length for each data set.
- rechdr is the record header length of each record.

If the records are WRITE ADD, WRITE ADD COMPLETE, or WRITE ADD DELETE records, rechdr is 84 and is followed by the record key and the record data (including its key).

If the result of the calculation shows a value for AVGBUFSIZE that is greater than the value defined for MAXBUFSIZE, the value defined for MAXBUFSIZE is taken as the value for AVGBUFSIZE, and writespersec is calculated:

```
writespersec = bytespersec / (MAXBUFSIZE - 36)
```

Structure size for user journal and autojournal usage

You can use the z Systems Coupling Facility Structure Sizer tool (CFSizer), or the formulas provided, to calculate storage requirements for the user journal and autojournal.

The CFSizer calculations assume that the coupling facility is at the current CFLEVEL. The formulas provided assume the coupling facility is at CFLEVEL 11.

Using CFSizer to determine structure size

CFSizer is a web-based application that communicates with a coupling facility at a current CFLEVEL to calculate storage requirements. For the user journal and autojournal, it calculates suitable values for the **INITSIZE** and **SIZE** parameters. See [System z Coupling Facility Structure Sizer Tool \(CFSizer\) web page](#).

To use CFSizer to calculate storage requirements for the user journal and autojournal, enter the following information:

Writes per second

The number of log blocks written to the log stream per second from a single system. This value is available in the CICS statistics reports. See [Logstream reports](#).

Maximum buffer size

The size, in bytes, of the largest log block that can be written to a logstream. The value must be between 1 and 65532. This value corresponds to the MAXBUFSIZE value in the definition of the logstream coupling facility structure. See [“Defining coupling facility structures” on page 136](#).

Average buffer size

The average size in bytes of log blocks written to all logstreams. This input is used to determine the initial entry to element ratio of the structure. This value corresponds to the AVGBUFSIZE value in the definition of the logstream coupling facility structure. See [“Defining coupling facility structures”](#) on page 136.

Number of logstreams

The number of logstreams allocated to a structure. This must be a value from 0 to 512. In a CICS environment, the number of logstreams should be between 10 and 20.

For the user journal and autojournal, CFSizer uses the following default values:

Highoffload

80

Lowoffload

0

Calculation of structure size

Calculate values for the INITSIZE and SIZE attributes. See [“Calculation of INITSIZE for DFHLOG”](#) on page 144 and [“Calculation of SIZE for DFHLOG”](#) on page 146.

To calculate a value for number_of_entries, use the following formula:

```
number_of_entries = writespersec * 12.5
```

See the explanation of writespersec later in this topic.

For journals with log blocks that are not forced to the log stream, the average block size tends to be slightly less than the MAXBUFSIZE value defined for the coupling facility structure.

For journals with log blocks that are forced to the log, using the **EXEC CICS WAIT JOURNALNAME** or **EXEC CICS WAIT JOURNALNUM** commands, or using the WAIT option of the **EXEC CICS WRITE JOURNALNAME** or **EXEC CICS WRITE JOURNALNUM** commands, you can calculate AVGBUFSIZE from the weighted average of the data logged for each journal logging to the same log stream for a given CICS system.

```
AVGBUFSIZE = (bytespersec / (writespersec) + 36
```

where:

- $\text{bytespersec} = (N1 * Wr1 * (D1 + \text{rechdr}) + \dots (Nn * Wrn * (Dn + \text{rechdr})))$
- $\text{writespersec} = \text{lesser of } 25 \text{ or } ((N1 * Wa1) + \dots + (Nn * Wan))$ where:
 - $N1, \dots, Nn$ is the number of transactions per second writing to the journal.
 - $Wr1, \dots, Wrn$ is the number of write requests per transaction.
 - $Wa1, \dots, Wan$ is the number of wait requests per transaction.
 - $D1, \dots, Dn$ is the average record length of each journal record.
 - rechdr is the record header length of each record.

Autojournal records are issued from file control. They might be DATA SET NAME records, which consist of a 204-byte record header, and no further data. Alternatively, they might be READ ONLY, READ UPDATE, WRITE UPDATE, WRITE ADD, or WRITE ADD COMPLETE records. In this case, rechdr is 84 bytes and is followed by the file control record itself.

User journal records consist of a 68-byte record header, followed by the user prefix and the user data.

If the result of the calculation shows a value for AVGBUFSIZE that is greater than the value defined for MAXBUFSIZE, the value defined for MAXBUFSIZE is taken as the value for AVGBUFSIZE, and writespersec is calculated as follows:

```
writespersec = bytespersec / (MAXBUFSIZE - 36)
```

Coupling facility requirements in an RLS environment

When you move to an RLS environment from an environment where multiple AORs accessed data sets in an FOR, the logging activity of the FOR is distributed across the AORs. Therefore, the coupling facility structure size required by each AOR increases.

Calculating increased AOR coupling facility storage requirements

Use the following formulae to calculate the `avgbuFSIZE`, `number of entries` and `akpintvl` values for the AORs in the new RLS environment.

Calculate the AOR `AVGBUFSIZE` value required by the `INITSIZE` and `SIZE` formulae:

```
AOR AVGBUFSIZE = ( AOR_bytes + (FOR_bytes / no. of AORs) ) / (intvlen * 25)
```

where:

- `AOR_bytes` is the number of bytes written to the system log by an AOR in the sampling interval.
- `FOR_bytes` is the number of bytes written to the system log by an FOR in the sampling interval.
- `no of AORs` is the number of cloned AORs using the FOR.
- `intvlen` is the length in seconds of the sampling interval.

Calculate the AOR `no. entries` value required by the `INITSIZE` and `SIZE` formulae:

```
AOR no. entries = ((AOR_akpintvl + trandur) * 25) / 0.9
```

where:

- $AOR_akpintvl = (AKPFREQ * intvlen) / (AOR_recs + (FOR_recs / no. of AORs))$

where:

- `intvlen` is the length in seconds of the sampling interval.
- `AOR_rec`s is the number of records written to the system log by an AOR in the sampling interval.
- `FOR_rec`s is the number of records written to the system log by an FOR in the sampling interval.
- `no of AORs` is the number of cloned AORs using the FOR.
- `trandur` is the execution time between sync points of the longest-running transaction that runs as part of the normal workload.

If this value is greater than `AOR_akpintvl`, use `AOR_akpintvl` as the duration or consider increasing `AKPFREQ`.

After you calculate the values for `AOR AVGBUFSIZE` and `AOR no. entries`, determine the values of **INITSIZE** and **SIZE**, as described in [“Structure size for system log usage” on page 143](#).

Use the reports produced from the log stream statistics from CICS Transaction Server for z/OS that occur in a reporting interval for the AORs and the FOR to calculate the values:

- The number of log write operations
- The amount of data written

Staging data sets for coupling facility logstreams

MVS usually keeps a second copy of the data written to the coupling facility in a data space, for use when rebuilding a coupling facility log if an error occurs.

This backup is satisfactory if the coupling facility is failure-independent from MVS; that is, in a separate CPC and non-volatile.

When the coupling facility is in the same CPC, or uses volatile storage, the MVS system logger supports staging data sets for copies of logstream data that might otherwise be vulnerable to failures that affect both the coupling facility and the MVS images.

When defining logstreams, consider the following tasks:

- Define STG_DUPLEX(YES) and DUPLEXMODE(COND) for those logstreams that are associated with the system log. These definitions ensure that the MVS system logger automatically copies to staging data sets if it detects that the coupling facility is not failure-independent and is a single point of failure, and is therefore vulnerable to permanent log data loss. `

A connection to a logstream contains a single point of failure if the coupling facility is volatile or it resides on the same CPC as the MVS system connecting to it. For example, if there are two CPCs, CPC1 and CPC2, and CPC1 has an MVS LPAR and a coupling facility, but CPC2 has only MVS LPARs, the connections from the MVS LPAR in CPC1 to the coupling facility are failure-dependent. If you lose CPC1, you lose both MVS and its local buffers and the coupling facility. However, the connections from CPC2 are failure-independent, because the system logger local storage and buffers are in a physically separate CPC from the coupling facility, and the data is lost only if both fail. With DUPLEXMODE(COND), failure-dependent connections result in staging sets, but failure-independent connections are not allocated staging data sets.

- If you are operating with only a single coupling facility, define STG_DUPLEX(YES) and DUPLEXMODE(UNCOND) for those logstreams associated with the system log.
- Define STG_DUPLEX(YES) and DUPLEXMODE(COND) for those logstreams associated with forward recovery logs. If you do not, and a failure causes loss of data from the logstream, you need to take a new image copy of the associated VSAM data sets. For the consequent period of time until this operation is complete, the data sets are not fully protected.
- If you operate a nonvolatile, stand-alone coupling facility for normal logging, with a PR/SM LPAR configured as a coupling facility acting as backup, define all logstreams with STG_DUPLEX(YES) and DUPLEXMODE(COND).
- Define each staging data set to be at least the same size as the logstream share of the coupling facility, but round the average block size up to 4 KB.

For example, you can calculate the staging data set size that corresponds to the basic coupling facility space requirement for each CICS system logstream (DFHLOG) by using the following formula:

```
staging data set size = number_of_entries * rnd-avg-buf / 4096
```

where rnd-avg-buf = avgbuFSIZE (rounded up to 4K)

To calculate a value for number_of_entries, see [number of entries calculation](#).

DASD-only log streams

The CICS log manager supports the DASD-only option of the MVS system logger.

Individual CICS log streams can use either coupling facility log structures or DASD-only logging. You might define a log stream to use DASD-only logging for these reasons:

- You do not have a coupling facility.
- You want to preserve coupling facility space for other uses.
- You do not require the log stream to be shared across MVS systems. (The CICS system log can never be shared.)

See [Setting up the environment for CICS log manager](#) for advice about defining individual log streams to use coupling facility or DASD-only logging, based on their usage.

Defining DASD-only log streams

Use the MVS IXCMIAPI utility to define DASD-only log streams to the LOGR couple data set.

The following code sample shows the basic syntax to define a DASD-only log stream:

```
DEFINE LOGSTREAM NAME(log_stream_name)
    DASDONLY(YES)
    MAXBUFSIZE(max_buFSIZE)
    STG_SIZE(stg_size)
```

```
HIGHOFFLOAD(high_offload)
LOWOFFLOAD(low_offload)
```

Figure 14 on page 151 shows example definitions for a pair of log streams associated with a DASD-only system log.

```
//DEFLOGS JOB ...
//LOGDEFN EXEC PGM=IXCMIAPU
//STEPLIB DD DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//*****
//*
//* Define DASD-only log streams for CICS system log.
//*
//* The LOWOFFLOAD and STG_SIZE values are for illustration
//* only -- substitute values appropriate for your environment.
//*
//*****
//SYSIN DD *
DATA TYPE(LOGR) REPORT(NO)
DEFINE LOGSTREAM NAME(region_userid.applid.DFHLOG)
                DASDONLY(YES)
                MAXBUFSIZE(64000) STG_SIZE(3000)
                LOWOFFLOAD(40) HIGHOFFLOAD(80)
DEFINE LOGSTREAM NAME(region_userid.applid.DFHSHUNT)
                DASDONLY(YES)
                MAXBUFSIZE(64000) STG_SIZE(500)
                LOWOFFLOAD(40) HIGHOFFLOAD(80)
```

Figure 14. Example definitions of DASD-only log streams

Using model log streams

To avoid defining explicitly each log stream used by each of your CICS regions, you can use model log stream definitions.

Using models, log streams are defined to MVS dynamically, on their first usage. Figure 15 on page 151 shows example DASD-only model definitions for CICS primary and secondary system log streams.

```
//DEFLOGS JOB ...
//LOGDEFN EXEC PGM=IXCMIAPU
//STEPLIB DD DSN=SYS1.MIGLIB,DISP=SHR
//SYSPRINT DD SYSOUT=*
//*****
//*
//* Define DASD-only model log streams for CICS system log.
//*
//* The LOWOFFLOAD and STG_SIZE values are for illustration
//* only -- substitute values appropriate for your environment.
//*
//*****
//SYSIN DD *
DATA TYPE(LOGR) REPORT(NO)
DEFINE LOGSTREAM NAME(sysname.DFHLOG.MODEL)
                MODEL(YES)
                DASDONLY(YES)
                MAXBUFSIZE(64000)
                STG_SIZE(3000)
                LOWOFFLOAD(40) HIGHOFFLOAD(80)
DEFINE LOGSTREAM NAME(sysname.DFHSHUNT.MODEL)
                MODEL(YES)
                DASDONLY(YES)
                MAXBUFSIZE(64000)
                STG_SIZE(500)
                LOWOFFLOAD(40) HIGHOFFLOAD(80)
```

Figure 15. Example model definitions for DASD-only system log streams

For information about the mapping of CICS journal definitions to log stream names, see [Setting up CICS log streams](#).

When using model log streams, note that, if you specify a STG_SIZE on the model definition, all new log streams created from the model have the same-sized staging data set.

Sizing for DASD-only log streams

You must work out the sizes of the CICS primary and secondary system log streams, forward recovery logs, and user journals and autojournals.

Sizing DFHLOG

To size the CICS primary system log stream, DFHLOG:

- **Minimize the amount of data that is offloaded to secondary storage**

The MVS system logger begins the offload process when the high offload threshold (HIGHOFFLOAD) of the log stream is reached. The offload process consists of two steps:

1. The MVS logger physically deletes the data in the log stream that has been marked for deletion by the CICS log-tail deletion process.
2. The MVS logger calculates how much data must be offloaded to auxiliary storage, based on the difference between HIGHOFFLOAD and LOWOFFLOAD, less the amount of data that has been deleted since the last offload event.

To minimize the amount of data offloaded from the CICS primary system log, you must:

- Define a suitably sized staging data set. You can alter the size of a staging data set without deleting the log stream. To do so, use the UPDATE LOGSTREAM request of the MVS IXCMIAPI utility to change the value of the STG_SIZE parameter.
- Ensure that the log-tail deletion process is working effectively. For detailed information about the log-tail deletion process, see [Log tail deletion](#).

- **Avoid "staging-data-set-full" events**

A staging-data-set-full event occurs when a log stream's staging data set becomes full before the offloading of data has completed.

For advice on monitoring and avoiding staging-data-set-full events, see [Monitoring the logger environment](#).

Sizing DFHSHUNT

You must size the secondary system log stream, DFHSHUNT, to avoid staging-data-set-full events. However, typically some data is offloaded from DFHSHUNT to secondary storage.

Sizing general logs

You must size forward recovery logs, user journals, and autojournals to avoid staging-data-set-full events. However, because CICS does not delete data from these log streams, data is typically offloaded to auxiliary storage.

Setting attribute values for log stream and system definitions

[Table 14 on page 153](#) summarizes how you decide on the values for various attributes on the log stream definition, and system definition.

Table 14. How to decide on the values of attributes

Facility	Attribute	Method or Value
Primary system log stream (DFHLOG)	HIGHOFFLOAD	80
	LOWOFFLOAD	Use the formula in “Sizing your DASD log streams” on page 153 to calculate a suitable value.
	MAXBUFSIZE	64000
	STG_SIZE	Use the formula in “Sizing your DASD log streams” on page 153 to calculate a suitable value.
Secondary system log stream (DFHSHUNT)	HIGHOFFLOAD	80
	LOWOFFLOAD	40 – 60
	MAXBUFSIZE	64 000
	STG_SIZE	500 (4 KB blocks)
General log stream	HIGHOFFLOAD	80
	LOWOFFLOAD	0
	MAXBUFSIZE	64 000
	STG_SIZE	Use the formula in “Sizing your DASD log streams” on page 153 to calculate a suitable value
CICS system	AKPFREQ	4 000

Startup might take longer than you experienced when using earlier releases. This is caused in part by the allocation and formatting of the staging data sets. Increased startup time depends on factors like these:

- Size of staging data set (STG_SIZE)
- DASD speed
- DASD contention

You can reduce startup time by avoiding large staging data sets.

Sizing your DASD log streams

If you are capacity planning for new applications, you must calculate your space requirements.

The formulae provided help you to calculate values for LOWOFFLOAD and STG_SIZE.

You must base your calculations on the journaling requirements of your applications. These requirements provide the starting point for the following formulae.

You can calculate LOWOFFLOAD for DFHLOG using the following formula:

$$\text{LOWOFFLOAD} = (\text{trandur} * 90) / (\text{akpintvl} + \text{trandur}) + 10 \quad (\text{where RETPD}=0 \text{ specified})$$

or

$$\text{LOWOFFLOAD} = (\text{trandur} * 90) / (\text{akpintvl} + \text{trandur}) \quad (\text{where RETPD}=dddd \text{ specified})$$

where:

- akpintvl is the interval between activity key points. Calculate it using the following formula:

$$\text{akpintvl} = \text{AKPFREQ} / ((N1 * R1) + (N2 * R2) + (Nn * Rn))$$

where:

- N1, N2 Nn is the transaction rate for each transaction (transactions per second).

- R1, R2 Rn is the number of log records written by each transaction.
- `trandur` is the execution time (between sync points) of the longest-running transaction that runs as part of the normal workload.

If this duration is longer than the `akpintvl` value, you have these choices:

- Increase the value of `AKPFREQ`, thus increasing the value of `akpintvl`, provided that this value does not result in an unacceptably large staging data set size.
- Change the application logic to cause more frequent sync points.
- Calculate a staging data set size based on a shorter transaction duration, and accept that offloading to auxiliary storage occurs when the long-running transaction is used.

You can calculate `STG_SIZE` for DFHLOG using the following formula:

```
Staging DS size    = (AKP duration) * No. of log writes per second
for system log (no. of 4k blocks)
where AKP duration = (CICS TS 390 AKPFREQ)/(No. buffers per second)
```

You can take the values for the number of log writes per second and buffer puts per second from your statistics. The value for log writes per second must not exceed 30.

Converting a DASD-only log stream to use a coupling facility

You can upgrade a DASD-only log stream to use a coupling facility structure, without having to delete and redefine the log stream.

1. Make sure that there are no connections, neither active nor failed, to the log stream.
2. Use the `UPDATE LOGSTREAM` request of the MVS `IXCMIAPU` utility. Specify the `STRUCTNAME` keyword, and let the `DASDONLY` keyword default to `NO`. For example:

```
//LOGUPDT JOB ...
//LOGUPDT EXEC PGM=IXCMIAPU
//SYSPRINT DD SYSOUT=A,DCB=RECFM=FBA
//*****
//*
//* Convert DASD-only log stream to coupling facility log stream.*
//*
//*****
//SYSIN DD *
DATA TYPE(LOGR) REPORT(NO)
UPDATE LOGSTREAM NAME(region_userid.applid.DFHLOG)
                  STRUCTNAME(LOG_DFHLOG_001)
                  STG_DUPLEX(YES) DUPLEXMODE(COND)
```

Figure 16. Converting a DASD-only log stream to use a coupling facility structure

1. To upgrade a DASD-only log stream to a coupling facility log stream that does *not* use a staging data set, you must explicitly specify `STG_DUPLEX(NO)`, because the DASD-only log stream by definition uses a staging data set; unless you specify `STG_DUPLEX(NO)`, this storage data set is retained by the coupling facility log stream.
2. You cannot use `UPDATE LOGSTREAM` to convert a log stream that uses a coupling facility structure to one that uses DASD-only. To do that, you must delete and redefine the log stream.

Analyzing system logger data (SMF Type 88 records)

When you review the output from the system logger reports produced by `IXGRPT1`, `IXGRPT1J`, and `IXGRPT1L`, look at the following key fields for CICS system logs.

Check the following key fields:

- The number of bytes deleted from primary storage is close to the number of bytes written
- The number of bytes deleted from the system log after writing to offload data sets is usually very low:
 - If this number is high, resources are being used to move data to the offload data set only to be later deleted.

- This number is a key indicator that log-tail deletion is not working effectively.
- Check the MVS system log for any DFHRM0205 and DFHLG0743 messages from the affected CICS region.
- Look for long-running tasks (using CICS monitoring data or a system monitoring package), or check if AKPFREQ is too high.
- In general offloads are acceptable, but offloads triggered by NTRY FULL indicators are not a good sign:
 - NTRY FULL indicates that the entry-to-element ratio is too high.
 - Such offloads are probably the result of unlike logstreams defined in the same structure.
 - The offloads are being triggered by all the entries being used rather than triggered by the HIGHOFFLOAD value.
- TYPE3 I/O counts must not appear in the statistics for coupling facility log streams, because they indicate that I/O is being initiated when over 90% of the elements for the log stream are in use.
- Average buffer size is important for these reasons:
 - If over 4 KB, the writes are asynchronous, rather than synchronous.
 - Buffer size is used to determine the entry to element ratio.
 - If MAXBUFSIZE specified on the log stream definition is *lower than* 65 532-bytes, the element size is 256-bytes.
 - If MAXBUFSIZE is 65 532-bytes, the element size is 512-bytes.
 - The entry-to-element ratio is calculated as (average-buffer-size plus 4 divided by 4).

Managing auxiliary storage

Use System Managed Storage (SMS) to manage log stream data sets.

Managing log data sets

You can specify the SMS characteristics of log data sets in a number of ways, depending on your installation.

Using automatic class selection (ACS) routines

You can use installation-written automatic class selection (ACS) routines to assign log data sets to SMS classes.

Using the LOGR policy

When you define or update a log stream definition in the LOGR policy, you can assign the SMS storage class, data class, and management class for both the DASD log data sets and staging data sets.

- Use LS_DATACLAS to specify the SMS data class to be used for log stream data set allocation.
- Use LS_STORCLAS to specify the SMS storage class to be used for log stream data set allocation.
- Use LS_SIZE to specify the size, in 4 KB blocks, of the log stream DASD data sets. Specify a size so that each data set can contain multiple offloads of the primary storage: this capability is particularly important if *all* the data is offloaded for a log stream, as in the case of user journals and forward recovery logs. The MVS system logger issues message IXG256I if you specify less than 64 KB.

If you omit the size parameter, the size is taken from the ALLOCxx member of PARMLIB. The default is 2 tracks, which leads to a high number of new data set allocations. Specify a size that is large enough to avoid a high frequency of new data set allocations; aim for a new data set to be allocated less frequently than once an hour.

SHAREOPTIONS(3,3)

Always define logger data sets with SHAREOPTIONS(3,3), whether the system is a part of a multiple-member sysplex or a monoplex. The common symptom of not having SHAREOPTIONS(3,3) is return code 84A or 403 from the logger.

For more information about managing log data sets, see [z/OS MVS Setting Up a Sysplex](#).

Managing the system log

CICS manages the system log by deleting records, for completed units of work, during activity keypoint processing (log-tail deletion). With an appropriately sized log stream, the system log data remains in primary storage, thus avoiding data spilling to DASD.

Note that:

- The CICS system log must be used only for short-lived data required for recovery purposes. For example, do not write audit trail user records to the system log.
- Allow CICS to manage the size of the system log.

However, if historically you have used the system log for such things as audit trails, you might want to preserve system log data beyond the time it is typically deleted by CICS. You can use the RETPD MVS parameter to preserve system log data. Define DFHLOG and DFHSHUNT to MVS with AUTODELETE(NO) and RETPD(dddd). The default values are AUTODELETE(NO) and RETPD(0). By specifying AUTODELETE(NO), CICS, rather than MVS, retains control of the log-tail trimming process; dddd is the number of days for which data is to be retained. In this way, the MVS logger physically deletes an entire log data set when *all* of the data in the data set has been marked for deletion by the CICS log-tail trimming process and is older than the retention period specified for the log stream.

You can view log data that has been marked for deletion by CICS, but not yet physically deleted by MVS, with the DFHJUP utility program or the VIEW=ALL option of the MVS IXGBRWSE macro.

Managing general logs

The number of data sets per log stream recognized by the MVS logger is several million. Therefore, in general, do not be concerned about the limit being reached. You can cause redundant data to be deleted from log streams automatically, after a specified period. To arrange deletion for general log streams, define the logs to MVS with AUTODELETE(YES) and RETPD(dddd), where dddd is the number of days for which data is to be retained. This definition causes the MVS system logger to delete an entire log data set when all the data in it is older than the retention period (RETPD) specified for the log stream.

Unicode data conversion by z/OS

z/OS Unicode services provides support for the conversion of Unicode data (either UTF-8 or UTF-16) to any of the EBCDIC CCSIDs currently supported by CICS.

For more information on Unicode services, see [z/OS Unicode Services User's Guide and Reference](#).

The three CCSIDs for UTF data are as follows:

- CCSID 1202 indicates UTF-16LE.
- CCSID 1201 indicates UTF-16BE.
- CCSID 1200, which indicates that the data carries a byte order marker that must be examined to see whether the following form is big-endian or little-endian.

When handling CCSID 1200, CICS respects the byte order marker for inbound conversions, but cannot retain that information when handling a related outbound conversion. All outbound data for CCSID 1200 is UTF-16BE. Application programmers must know about this CCSID restriction, and perform their own BE to LE conversions if necessary.

Applying service to CICS Transaction Server for z/OS

Service material for CICS Transaction Server for z/OS is distributed as APAR fixes and PTFs.

An APAR (Authorized Program Analysis Report) is raised when you and your IBM programming service representative agree that a CICS problem exists. You might be given an APAR fix. When the problem has been analyzed, all users are sent a PTF (Program Temporary Fix) to correct the problem permanently on the current release. PTFs are incorporated into any future CICS release. Both types of change are called SYSMODs (system modifications).

Using SMP/E control statements, you can process SYSMODs in three stages:

1. The **RECEIVE** control statement moves the SYSMOD into the PTF temporary store (PTS) data set. This operation is reversed by the **REJECT** control statement.
2. The **APPLY** control statement moves the SYSMOD into the target libraries. This operation is reversed by the **RESTORE** control statement.

At this point you can test the modified system.

3. The **ACCEPT** control statement moves the SYSMOD into the distribution libraries. This operation is not easily reversed.

When you are dealing with APAR fixes, apply the SYSMOD, but do not accept it. If you later obtain a PTF that solves the problem in a different way, you might be asked to restore (that is, remove) the APAR fix and apply the PTF instead.

When you are dealing with PTFs, apply the SYSMOD and then test it. Afterward you can accept it.

For background information about SMP/E operations, see the [SMP/E for z/OS User's Guide](#). For more detailed information, see [SMP/E for z/OS Reference](#).

Software maintenance strategy for CICS TS and its embedded components

This information helps you plan for updates to CICS Transaction Server for z/OS (CICS TS) and related software components such as WebSphere® Liberty and CICS Explorer. Installing APAR fixes (PTFs) using Recommended Service Upgrades (RSUs) as early as possible can help avoid problems that could result in a service call, and as long as you test appropriately, help reduce risks to your business.

Components in CICS TS

CICS TS provides the following embedded runtime components:

Equinox

Equinox is an implementation of the OSGi core framework specification and is provided for both OSGi JVM servers and for Liberty JVM servers. Service for these components is provided when and if required.

Apache Axis2

Axis2 is an open source web services engine from the Apache Software Foundation and is used within CICS to provide the web service Java engine for the Axis2 JVM server environment. Service for this component is provided when and if required.

WebSphere Liberty

Liberty fix packs are provided on a regular basis through the CICS TS APAR delivery process and provide an update mechanism to keep the embedded CICS Liberty runtime synchronized with the fix packs provided by WebSphere Application Server. Keeping current helps you resolve problems faster for two reasons:

- Liberty APAR fixes (known as ifixes) can only be delivered on the latest Liberty fix pack that has been supplied by the CICS APAR process.
- CICS APAR fixes can require other fixes such as WebSphere Liberty fix packs.

CICS Explorer and z/OS Explorer

The CICS Software Development Kit (SDK) for Java includes components from CICS for developing CICS Java applications. The CICS SDK for Servlet and JSP support includes Java Enterprise Edition (Java EE) and Liberty API libraries, and automatically installs required components such as WebSphere Developer Toolkit (WDT). Service for these components is provided through the CICS Explorer fix pack process when and if required.

Update recommendations

Recommended Service Upgrade (RSU)

IBM recommends that you install all RSU PTFs as preventive software maintenance for CICS TS. An RSU is a predefined collection of PTFs identified with a SOURCEID of RSUyy mm , where yy indicates the last 2 digits of the year and mm indicates the month. You can order an RSU from [Shopz](#).

Equinox and Apache Axis2

There is unlikely to be much maintenance for Equinox and Axis2. Service for these components is provided when and if required, meaning that it will be on a case by case basis and in most cases this will be to resolve service requests.

WebSphere Liberty

See the section Fixes for WebSphere Application Server Liberty in [Changes to feature toggles](#) for a list of APARs that you will need to update CICS TS to the latest version of WebSphere Liberty.

CICS Explorer

IBM recommends that you upgrade to the latest release of CICS Explorer to obtain the most recent functionality and service. Old releases of CICS Explorer are not generally updated with new function, but new CICS Explorer releases are compatible with all earlier versions of CICS TS. Therefore, CICS Explorer for z/OS Aqua V3.2 (Fix Pack 5.5.0.9 and higher) is compatible with all versions of CICS TS including CICS TS 5.6. For further details, see [Latest version of CICS Explorer and CICS Tools plug-ins for CICS Explorer](#).

Note: CICS Explorer V5.5 is now referred to as CICS Explorer for Aqua V3.2 and CICS Explorer V5.4 is now referred to as CICS Explorer for Aqua V3.1 (where Aqua refers to IBM Explorer for z/OS Aqua). For details, see the [CICS Transaction Server for z/OS V5.6 announcement letter](#).

Load library secondary extents

CICS supports load library secondary extents that are created while CICS is running.

If you define libraries in the DFHRPL or dynamic LIBRARY concatenation with primary and secondary extents, and secondary extents are added while CICS is running, as a result of link-editing into the load library, the CICS loader detects the occurrence and closes and then reopens the library. In this way, you can introduce new versions of programs by using the CEMT NEWCOPY command, even if the new copy of the program has caused a new library extent.

However, do not attempt to apply service to data sets that are used by CICS TS components that are running.

The CICS TS-supplied SMP/E procedure

DFHSMPE is a CICS TS-supplied procedure for applying service to the CICS and CICSplex SM components of CICS TS.

This procedure is tailored to your environment and stored in the *hlq*.XDFHINST library when you run the DFHISTAR job.

For information about how to apply corrective service with SMP/E, see the [SMP/E for z/OS User's Guide](#).

Whenever you do any SMP/E processing on CICS or CICSplex SM software, and you use any of the examples quoted in the SMP/E for z/OS User's Guide, specify DFHSMPE as the name of the SMP/E procedure on the EXEC statement; that is, in place of SMPPROC, as used in the examples. The DFHSMPE procedure includes the following DD statement for supplying SMP/E control statements:

```
//SMPCTL DD DSN=&&SETBDY,DISP=(OLD,DELETE)
//      DD DDNAME=DFHSMPIN
```

The ZNAME parameter of the DFHSMPE procedure generates a SET BDY command for the zone that is identified by the parameter. The command is stored in the temporary data set, SETBDY. The ZNAME parameter is set to the value of *zonename* that you specify for the TZONE parameter. If you do not specify any value for *zonename* for the TZONE parameter of the DFHISTAR job, *zonename* and the ZNAME value default to TZONE.

The ZNAME parameter also generates a SET BDY command in DFHAUPLE, the CICS TS procedure supplied for assembling and link-editing CICS control tables.

If you supply an *override* SMP_CNTL DD statement in the job that runs DFHSMPE, it must come *before* any DD statements that are additional to the procedure. Furthermore, if you provide an override, you receive the following MVS system message:

```
IEF686I DDNAME REFERRED TO ON DDNAME KEYWORD IN PRIOR STEP WAS NOT RESOLVED
```

You receive this message because the DD statement for DFHSMPIN is missing because of the SMP_CNTL DD override. However, the message is not a JCL error, and does not prevent the step from running successfully with a return code of 0.

If you supply any SMP/E control statements in your job using the DFHSMPIN ddname, they are prefixed by a SET BDY command for the zone that you specify on the ZNAME parameter. If you run SMP/E with a command that does not require this SET BDY statement, the way your job runs is not affected.

APARs and PTFs

An APAR is a temporary fix created for individual customers. PTFs are intended for all users to install to avoid possible problems.

APARs

Generally, do *not* ACCEPT APAR fixes into distribution libraries. Subsequent PTFs might not include the APAR fix, and you might be required to reapply the APAR fix.

If two APAR fixes depend on each another, and each is a prerequisite of the other, you must apply them both in the same SMP/E APPLY processing step.

PTFs

A PTF can contain fixes for several different problems, so several APAR fixes reported in RETAIN might be superseded by the more permanent PTF:

- A PTF provides card-image changes that are functionally equivalent to those in the APAR fix.
- A PTF contains object-module replacements for preassembled CICS TS programs.

For further information about using SMP/E to apply service, see [SMP/E for z/OS User's Guide](#).

Mapping CICS releases from APARs and PTFs to CICS external versions

CICS APARs include internal releases in the **Applicable component levels** section of the APAR on the web. The internal release numbers are different from the external, dotted form of CICS versions such as V5.6. This information helps you map the internal release numbers that IBM uses for CICS APARs and PTFs to the corresponding external versions of CICS TS.

When you view the web page for a CICS APAR such as [PH15887](#), the external, dotted form of the release for which an APAR is created is listed under **Software version** on the top right of the web page. However, as you can see in this example, an APAR can have PTFs for multiple releases of the product. If you view the **Applicable component levels** section, PH15887 includes the following three internal releases:

- R000 PSY UI67166
- R100 PSY UI67167
- R200 PSY UI67165

An entry in **Applicable component levels** follows the pattern *Rnnn PSY Uxxxxnn*, where

Rnnn provides a 3-digit number that indicates an internal release.

Uxxxxnn is the PTF number.

You can use the mapping tables below to map the internal release in an APAR for a PTF to the external version and release.

In the case of APAR PH15887, the APAR is created for CICS TS V5.3 but it has PTFs for CICS TS V5.3, V5.4, and V5.5.

Release mapping reference

Select from one of the following products to locate the corresponding table containing the release mappings for each version. Note that the mapping reference for CICS Service Flow Runtime and REXX is provided in [Table 20 on page 163](#) and is applicable to all CICS versions.

- [“CICS Transaction Server for z/OS V5.6” on page 160](#)
- [“CICS Transaction Server for z/OS V5.5” on page 161](#)
- [“CICS Transaction Server for z/OS V5.4” on page 161](#)
- [“CICS Transaction Server for z/OS V5.3” on page 162](#)
- [“CICS Transaction Server for z/OS V5.2” on page 162](#)
- [“CICS Service Flow Runtime and REXX” on page 163](#)

CICS Transaction Server for z/OS V5.6

Table 15. Mapping reference for CICS Transaction Server for z/OS V5.6			
Name	Component ID	Release	FMID
CICS Transaction Server for z/OS V5.6 - Base	5655Y0400	300	HCI7300
COBOL Language parts	5655Y0400	301	JCI7301
PL/I Language parts	5655Y0400	302	JCI7302
C Language parts	5655Y0400	303	JCI7303
WebSphere Liberty	5655Y0400	30L	JCI730L
CICSplex System Manager	5655Y0400	30M	JCI730M
WS-Security	5655Y0400	30W	JCI730W
CICS Explorer	5655Y0401	550	Not applicable
UrbanCode Plugin	5655Y0402	100	Not applicable
CICS TS Build Toolkit	5655Y0403	550	Not applicable
CICS TS Enabler / Activation	5655Y0400	560	HCTS560
CICS JCICSX development feature for Java	5655Y0404	100	Not applicable
Value Unit Edition	5722DFJ00	300	HCV730
Developer Trial Activation Module	5655Y3000	300	HCD730

See also [“CICS Service Flow Runtime and REXX” on page 163](#).

CICS Transaction Server for z/OS V5.5

Table 16. Mapping reference for CICS Transaction Server for z/OS V5.5			
Name	Component ID	Release	FMID
CICS Transaction Server for z/OS V5.5 - Base	5655Y0400	200	HCI7200
COBOL Language parts	5655Y0400	201	JCI7201
PL/I Language parts	5655Y0400	202	JCI7202
C Language parts	5655Y0400	203	JCI7203
WebSphere Liberty	5655Y0400	20L	JCI720L
CICSplex System Manager	5655Y0400	20M	JCI720M
WS-Security	5655Y0400	20W	JCI720W
CICS Explorer	5655Y0401	550	Not applicable
CICS TS Build Toolkit	5655Y0403	550	Not applicable
CICS TS Enabler / Activation	5655Y0400	550	HCTS550
Value Unit Edition	5722DFJ00	200	HCV7200
Developer Trial Activation Module	5655Y3000	200	HCD7200

See also “CICS Service Flow Runtime and REXX” on page 163.

CICS Transaction Server for z/OS V5.4

Table 17. Mapping reference for CICS Transaction Server for z/OS V5.4			
Name	Component ID	Release	FMID
CICS Transaction Server for z/OS V5.4 - Base	5655Y0400	100	HCI7100
COBOL Language parts	5655Y0400	101	JCI7101
PL/I Language parts	5655Y0400	102	JCI7102
C Language parts	5655Y0400	103	JCI7103
IIOP/Java	5655Y0400	10D	JCI710D
WebSphere Liberty	5655Y0400	10L	JCI710L
CICSplex System Manager	5655Y0400	10M	JCI710M
WS-Security	5655Y0400	10W	JCI710W
CICS Explorer	5655Y0401	540	Not applicable
CICS TS Build Toolkit	5655Y0403	540	Not applicable
CICS TS Enabler / Activation	5655Y0400	540	HCTS540
Value Unit Edition	5722DFJ00	100	HCV7100

Table 17. Mapping reference for CICS Transaction Server for z/OS V5.4 (continued)

Name	Component ID	Release	FMID
Developer Trial Activation Module	5655Y3000	100	HCD7100

See also “CICS Service Flow Runtime and REXX” on page 163.

CICS Transaction Server for z/OS V5.3

Table 18. Mapping reference for CICS Transaction Server for z/OS V5.3

Name	Component ID	Release	FMID
CICS Transaction Server for z/OS V5.3 - Base	5655Y0400	000	HCI7000
COBOL Language parts	5655Y0400	001	JCI7001
PL/I Language parts	5655Y0400	002	JCI7002
C Language parts	5655Y0400	003	JCI7003
Java	5655Y0400	00D	JCI700D
WebSphere Liberty	5655Y0400	00L	JCI700L
CICSplex System Manager	5655Y0400	00M	JCI700M
WS-Security	5655Y0400	00W	JCI700W
CICS Explorer	5655Y0401	530	Not applicable
Activation Module	5655Y0400	530	HCTS530
Value Unit Edition	5722DFJ00	000	HCV7000
Developer Trial Activation Module	5655Y3000	000	HCD7000

See also “CICS Service Flow Runtime and REXX” on page 163.

CICS Transaction Server for z/OS V5.2

Table 19. Mapping reference for CICS Transaction Server for z/OS V5.2

Name	Component ID	Release	FMID
CICS Transaction Server for z/OS V5.2 - Base	5655Y0400	900	HCI6900
COBOL Language parts	5655Y0400	901	JCI6901
PL/I Language parts	5655Y0400	902	JCI6902
C Language parts	5655Y0400	903	JCI6903
IIOP/Java	5655Y0400	90D	JCI690D
WebSphere Liberty	5655Y0400	90L	JCI690L
CICSplex System Manager	5655Y0400	90M	JCI690M

Table 19. Mapping reference for CICS Transaction Server for z/OS V5.2 (continued)

Name	Component ID	Release	FMID
WS-Security	5655Y0400	90W	JCI690W
CICS Explorer	5655Y0401	520	Not applicable
Activation Module	5655Y0400	520	HCTS520
Value Unit Edition	5722DFJ00	900	HCV6900
Developer Trial Activation Module	5655Y3000	900	HCIC900

See also “CICS Service Flow Runtime and REXX” on page 163.

CICS Service Flow Runtime and REXX

Table 20. Mapping reference for Service Flow Runtime and REXX

Name	Component ID	Release	FMID
Service Flow Runtime	5655M1502	300	HCIZ300
CICS REXX Runtime Facility	565508700	110	H0B5110
CICS REXX Development System	565508600	110	H0B7110
CICS REXX Common	565511200	110	H0Z2110

Applying service to CICS modules

If you use the CICS TS-supplied SMP/E usermod to install a module into the LPA (for example, into the *hlq.SDFHLPA* library), and later apply service to that module, the LPA-resident version of the module is serviced. If you have not used the SMP/E usermod to install the module into the LPA, the original version in the *hlq.SDFHAUTH* library or *hlq.SDFHLOAD* library is serviced.

After you have installed CICS, and before you start the postinstallation tasks, change the *TEMPLIB* parameter and the *SYSPROC DD* statement of the *DFHISTAR* job to refer to the *hlq.SDFHINST* library. This reference ensures that, if you apply service to any of the skeleton jobs, the changes applied to the *hlq.SDFHINST* library are used in subsequent runs of *DFHISTAR*. In any such subsequent runs of *DFHISTAR*, you can use the *SELECT* parameter to select any jobs that are affected by service.

1. If *DFHISTAR* is serviced, add the service changes to your *DFHISTAR* module in the *hlq.TDFHINST* library to preserve your current installation parameters, or respecify your current installation parameters in the serviced *DFHISTAR* module, which you can copy from the *hlq.SDFHINST* library to the *hlq.TDFHINST* library.
2. Linkage editor messages IEW0461, IEW2454, IEW2646, IEW2651, and IEW2689 are produced during the *APPLY* stage for unresolved external references. These messages are issued, giving a return code of 4, when some CICS load modules are link-edited during PTF installation. You can ignore these IEWxxxx messages because they are produced for component object modules of executable CICS load modules.
3. HCI7300 and JCI730E PTFs to ship Java service are often significantly larger than those for the base CICS product and might require more system resources during *APPLY* processing. To avoid errors caused by insufficient storage, do not restrict the region size for the SMP/E *APPLY* step for such PTFs. If a region size limit is used and the *APPLY* fails with errors relating to insufficient storage, it might be necessary to increase or remove the limit for the SMP/E job. In some cases, you might require a region size of 500 MB or more.

Applying service to CICSplex SM

Service material for CICSplex SM is distributed as corrective or preventive service.

Both types of changes are called system modifications (SYSMODs). SYSMODs are processed using SMP/E control statements.

For background information on SMP/E operations, see [SMP/E for z/OS User's Guide](#). For more detailed information, see [SMP/E for z/OS Reference](#). For information about how to apply corrective service using SMP/E, see [SMP/E for z/OS User's Guide](#).

CICS Transaction Server for z/OS-supplied SMP/E procedure

For all CICS Transaction Server regions, the procedure for applying service is DFHSMPE.

You can customize this procedure with the DFHISTAR job that is stored in the CICSTS56.CICS.XDFHINST library.

For full details about applying service to the CICSplex SM component of CICS TS, see [“Applying service to CICS Transaction Server for z/OS” on page 156](#).

Applying PTFs to CICSplexes

This procedure outlines general good practices for applying any CICSplex SM maintenance. Further instructions may be issued through ++HOLD ACTION data within an individual PTF.

Important: When you apply service to CICSplex SM, PTFs that are applied to the ESSS are not intended to be downward-compatible with earlier maintenance levels at the same release. This means that all CMASes, MASes, WUI Server regions and API programs must run at the same maintenance level as the ESSS for their release. Otherwise, abends, data corruption and unexpected results might occur. See [Designing your CICSplex SM environment](#) for additional considerations about the configuration of CICSplex SM components. When you apply PTFs to CICSplex SM, it is essential that all ++HOLD ACTION items associated with the SMP/E maintenance are followed carefully.

1. First apply the PTF to the maintenance point for the CICSplex.
2. If the PTF modifies a repository record, upgrade the repository at the maintenance point before restarting the maintenance point CMAS.
3. After the maintenance point CMAS is restarted, apply the PTF to any local MASs connected to the maintenance point CMAS. You can restart these local MASs one at a time. You do not have to restart them all together.
4. After the maintenance point CMAS is restarted, apply the PTF to any other CMASs in the CICSplex. You can do this at the same time as the MASs connected to the maintenance point are being updated. You can update the non-maintenance point CMASs one at a time and you do not have to restart them all together.
5. If the PTF modifies a repository record, upgrade the repository for the non-maintenance point CMAS before restarting it.
6. After a non-maintenance point CMAS is restarted, you can apply the PTF to any local MASs connected to that CMAS and restart them. You can restart these local MASs one at a time. You do not have to restart them all together.

To summarize, as a best practice, update the maintenance point CMAS first, then the other CMASs, and update a MAS only after its owning CMAS has been updated. This procedure is important if the PTF modifies a CICSplex SM resource table. A PTF might contain additional documentation, giving further instructions specific to that PTF.

If you have multiple CMASes connected together in a network with more than one CMAS being a maintenance point for different CICSplexes, you must apply PTFs to all CMASes simultaneously that function as a maintenance point.

For example, consider the following configuration:

- CMAS1 is a maintenance point for PLEX1
- CMAS2 is a maintenance point for PLEX2
- CMAS3 helps manage PLEX1 and PLEX2
- CMAS4 helps manage PLEX2

There are CMAS-to-CMAS connections between each CMAS in this list.

If you apply PTF maintenance to CMAS1, you must also apply it to CMAS2 at the same time, because both CMASs function as maintenance points for various CICSplexes. Failure to do so results in CMAS isolation.

Applying service to CICSplex SM modules

If you use the CICS TS-supplied SMP/E USERMOD to install modules into the LPA (for example, into the CICSTS56.CPSM.SEYULPA library), and later apply service to that module, the LPA-resident version of the module is serviced. If you have not used the SMP/E USERMOD to install the module into the LPA, the original version in the CICSTS56.CPSM.SEYUAUTH library or CICSTS56.CPSM.SEYULOAD library is serviced.

After you have applied CICSplex SM service, ensure that all CICSplex SM regions are running with a consistent set of CICSplex SM libraries. Failure to do so can cause unpredictable results.

For more information about applying service to CICSplex SM, see [“CICS Transaction Server for z/OS-supplied SMP/E procedure” on page 164](#).

Chapter 8. Getting ready to run CICS

Before you can run CICS, you must customize the CICS TS supplied skeleton jobs and enable services that are required by CICS.

For detailed information on configuring CICS, including setting up the data sets, connectivity, and configuring CICSplex SM see [Configuring](#).

Tailoring the CICS-supplied skeleton jobs

You tailor the CICS-supplied skeleton jobs to your CICS TS environment.

If you used CBPDO or the ServerPac to install CICS TS, edit and run DFHISTAR to tailor the CICS-supplied skeleton jobs that create the CICS TS data sets and run the CICS TS IVPs.

If you used the distribution tape to install CICS TS you have typically tailored the skeleton jobs already, and you can now start “Jobs for creating the CICS data sets” on [page 170](#).

Whichever method you used to install CICS TS, you can edit and run DFHISTAR several times, to create different copies of the skeleton jobs or subsequently change them. For example, to create several copies of DFHDEFDS to define data sets unique to several CICS regions, or if you have to apply service to any of the installation-related jobs. In this way, you can tailor the jobs to your CICS environment after you have loaded the CICS software into the SMP/E-supported CICS libraries.

The CICS installation libraries

When you use CBPDO to install CICS TS, you use the TDFHINST, XDFHINST, ADFHINST, and SDFHINST installation libraries.

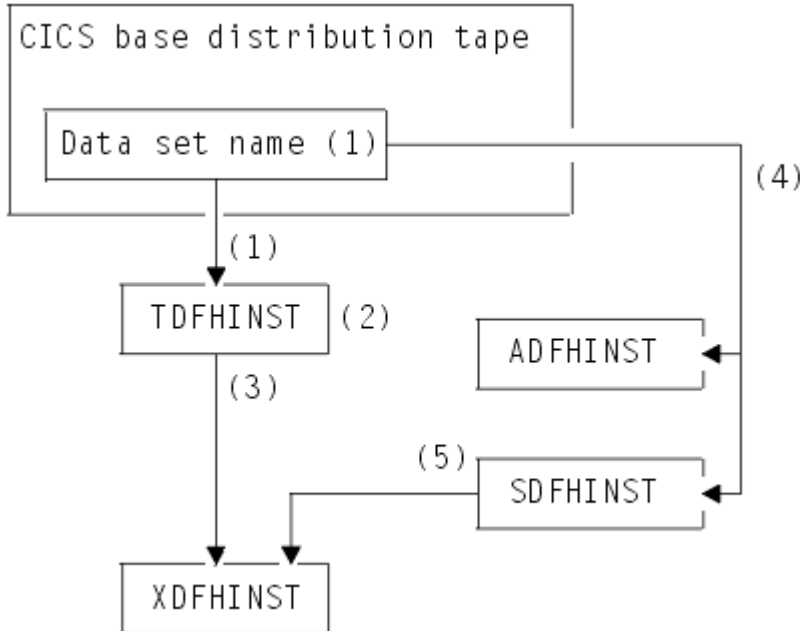


Figure 17. Installation libraries for this release

The names of the CICS installation libraries in [Figure 17 on page 167](#) and their use is explained in the following notes which refer to the numbers in the figure:

1. Skeleton installation-related jobs are copied from data set HCI 640.F2 on the distribution tape into *hlq.TDFHINST*. *hlq.TDFHINST* is used to store the DFHISTAR that you edit and run to tailor the skeleton

installation-related jobs to your CICS environment. Until you have installed the CICS software into the SMP/E-supported CICS libraries, this library also stores the skeleton jobs to be tailored.

2. You edit DFHISTAR in the *hlq.TDFHINST* library, to specify CICS installation parameters specific to your CICS environment.
3. When you run DFHISTAR, the tailored copies of the skeleton jobs are copied from the *hlq.TDFHINST* library to the *hlq.XDFHINST* library. *hlq.XDFHINST* is used to store the tailored, executable, copies of the skeleton jobs that are to be run.
4. To install CICS, you run the tailored copies of the CICS-supplied installation jobs to transfer the CICS software from the distribution tape to the *hlq.ADFHINST* and *hlq.SDFHINST* libraries. *hlq.ADFHINST* is the SMP/E-supported distribution installation library. *hlq.SDFHINST* is the SMP/E-supported target installation library. After you have installed the CICS software into this library and other SMP/E-supported libraries, named SDFHxxxx and ADFHxxxx, the skeleton jobs that you use on any later runs of DFHISTAR are stored in the SDFHINST library.

The names of the TDFHINST and XDFHINST libraries, and the prefix for those and other CICS libraries, are defined in DFHISTAR, which you edit as described in this topic.

Running DFHISTAR

When you have edited DFHISTAR with the values for installation parameters for your CICS environment, submit DFHISTAR.

When you run DFHISTAR, it tailors the skeleton jobs selected by the SCOPE or SELECT parameter in the DFHISTAR input to your environment and adds them to the library that you specified on the LIB parameter (by default, *hlq.XDFHINST*). If necessary, DFHISTAR creates the library that is specified on the LIB parameter.

You must specify the full name of the installation library from which the skeleton jobs are obtained, on the TEMPLIB parameter and SYSPROC DD statement of DFHISTAR (by default, *hlq.TDFHINST*). For the postinstallation tasks that are described, specify TEMPLIB SDFHINST.

DFHISTAR produces a job log and, if necessary, an error code:

- The output job log lists the values that were used for the parameters of DFHISTAR.
- If an error occurs when running DFHISTAR, an error code of 4 or 12 is returned. For error code 4, the skeleton jobs are tailored and added to the *hlq.XDFHINST* library. For error code 12, the skeleton jobs are not tailored or copied. To resolve the cause of the error, examine the output job log and, if necessary, edit and submit DFHISTAR again.

CICSplex SM postinstallation members

A number of CICSplex SM postinstallation members are delivered, as skeletons, in the TDFHINST library. When you run DFHISTAR, the postinstallation members are customized and saved in the XDFHINST library.

Use these members to create a basic CICSplex SM configuration that consists of a CMAS, a WUI, and a managed CICS system (MAS).

Postinstallation members are split into three areas as shown in [Table 21 on page 168](#), [Table 22 on page 169](#), and [Table 23 on page 169](#).

Table 21. Postinstallation members for a CMAS	
Member	Description
EYUCMASJ	JCL to start a CMAS. It runs EYUCMASP.
EYUCMASP	Starts a CMAS. This CMAS uses the CICS-supplied sample table, DFHSIT6\$, but appropriate override values are supplied in the job.
EYUCMSOP	EYUPARM parameters for a CMAS.

Table 21. Postinstallation members for a CMAS (continued)

Member	Description
EYUCMSDS	JCL to create and initialize the data sets for a CMAS.
EYUCMSSP	CICS SIT overrides for a CMAS.

For more information on CMAS data set customization, see [Creating and customizing CMAS data sets](#).

Table 22. Postinstallation members for a WUI

Member	Description
EYUJWREP	JCL to delete and define a WUI data repository. This function is also included in EYUWUIDS.
EYUWUIOP	EYUPARM parameters for a WUI.
EYUWUIDS	JCL to create and initialize the data sets for a WUI.
EYUWUIIN	EYUWUI parameters for a WUI.
EYUWUIJ	JCL to start a WUI. It runs EYUWUIP.
EYUWUIP	Starts a WUI. This WUI uses the CICS-supplied sample system initialization table, DFHSIT6\$, but appropriate override values are supplied in the job.
EYUWUISP	CICS SIT overrides for a WUI.

For more information on WUI customization, see [Creating and customizing the WUI data set](#).

Table 23. Postinstallation members for a managed CICS system (MAS)

Member	Description
EYUCSYDS	JCL to create and initialize the data sets for a managed CICS system.
EYUCSYSJ	JCL to start a managed CICS system. It runs EYUCSYSP.
EYUCSYSP	Procedure to start a managed CICS system. The MAS uses the CICS-supplied sample system initialization table, DFHSIT6\$, but appropriate override values are supplied in the job.
EYUJHIST	JCL to delete and define a pair of history data sets.
EYULMSOP	EYUPARM parameters for a managed CICS system.
EYULMSSP	CICS SIT overrides for a managed CICS system.
EYULPMOD	JCL to apply the USERMOD function, EYU\$UM01, that moves some MAS load modules to a link pack area (LPA) library.

For more information on managed CICS system customization, see [Creating and customizing MAS data sets](#).

Jobs for creating the CICS data sets

After you have installed CICS, and applied any necessary service, you can run the DFHCOMDS, DFHDEFDS, DFHCMACI, DFH0JCUS, and DFH0JHLP jobs to create CICS data sets. These data sets are required if you want to run the IVP.

When you run DFHISTAR, these jobs are tailored to your environment and stored in the library that you specify on the **LIB** parameter of DFHISTAR (by default, *hlq.XDFHINST*). If you have not yet run DFHISTAR, do so before running any of the CICS postinstallation jobs.

You can generate several copies of these jobs by rerunning DFHISTAR, selecting the jobs that you want to copy. To generate new copies of these jobs, edit DFHISTAR to specify new values for the **DSINFO** and **SELECT** parameters. Only those jobs that you name in the **SELECT** parameter are regenerated.

The supplied level of DFSMS changes the way that the index control interval size (CISIZE) is calculated for the VSAM data sets that have index components. As a result, expect a large default CISIZE, which can cause open errors during CICS startup. The change in CISIZE applies to both CICS data sets and your own application data sets.

CICS supplies the following jobs to create the required data sets:

DFHCOMDS

Deletes and re-creates data sets common to all CICS regions.

DFHDEFDS

Deletes and re-creates copies of data sets that are used only by one CICS region. You can run a separate copy of this job to create the data sets for each CICS region.

DFHCMACI

Deletes and re-creates the CICS messages data set, dsindex.DFHMACD, and loads it with the data from the CICS-supplied file, DFHMACD, in the *hlq.SDFHMSG* target library.

Note: This job is optional if you are installing CICS using the DFHISTAR job, but is used in [Defining the CMAC messages data set](#).

DFH0JCUS

Deletes and re-creates the sample applications details data set, dsindex.SAMPLE.DFHCTCUS and its associated alternate index and path, and loads it with the data from the CICS-supplied file, DFH0DCUS, in the *hlq.ADFHAPD2* library.

DFH0JHLP

Deletes and re-creates the sample applications help data set, dsindex.SAMPLE.DFHCTHLP, and loads it with the data from the CICS-supplied file, DFH0DHLP, in the *hlq.ADFHAPD1* library.

Data set naming conventions

No restrictions apply to the data set names you choose for CICS data sets, other than z/OS constraints.

To learn about the rules and constraints for z/OS data set names, see [DSNAME parameter](#) in the [z/OS MVS JCL Reference](#).

In the examples in this information, *hlq* is the high-level qualifier, and the DD name is the lowest level. If you are running multiple CICS regions, you can use the CICS APPLID as a third-level qualifier.

The *CTGI* naming convention is an example of a naming convention that you can use for CICS 4-character names, and is based on the 4-character *CTGI* symbol:

- *C* identifies an entire CICSplex.
- *T* identifies the type of region.
- *G* identifies a group of regions.
- *I* identifies iterations of regions in a group.

Use the *CTGI* naming convention. For example, if CICSHTH1 is the APPLID, the following data set name for the CICS system definition data set (CSD) is correct:

```
DFHCSD DD DSN=CICSTS56.CICS.CICSHTH1.DFHCSD,DISP=SHR
```

Where names are allowed to be up to eight characters long, as for CICS APPLIDs, the letters CICS are used for the first four characters, particularly for production regions.

DFHCOMDS job for common data sets

The DFHCOMDS job deletes and re-creates the data sets that are common to CICS regions. The common data sets are DFHCSD and SYSIN.

For detailed information on creating these data sets for your CICS regions, see [Setting up the CICS system definition data set](#).

The DFHCOMDS job creates one of each of these data sets common to all CICS regions. If you use separate copies of any of these data sets for each CICS region, move and edit the appropriate statements into the DFHDEFDS job. For further information about creating multiple copies of these data sets, see [“Creating several copies of the DFHCSD and SYSIN data sets” on page 171](#).

The DFHCOMDS job comprises five job steps:

1. DELETE deletes the data sets.
2. DEFCSDD defines the VSAM cluster for the CICS system definition data set, dsindex.DFHCSD, where *dsindex* is defined by the **DSINFO** parameter of DFHISTAR.
3. INITCSD initializes the CICS system definition data set.
4. DEFRRPIDC defines the VSAM cluster for ONC RPC.
5. DEFSYSIN creates the SYSIN PDS and copies the following modules from the *hlq.SDFHSAMP* library:
 - DFH\$SIPA
 - DFH\$SIPD
 - DFH\$SIPT
 - DFH\$SIP1
 - DFH\$SIP2
 - DFH\$SIP5
 - DFHRCNO
 - DFHRCYES

Creating several copies of the DFHCSD and SYSIN data sets

The CICS-supplied DFHCOMDS job creates one of each of the DFHCSD and SYSIN data sets common to all CICS regions.

To use separate copies of any of these data sets for each CICS region:

- Move the statements that define the data set from the DFHCOMDS job to the DFHDEFDS job.
- Edit the statements in the DFHDEFDS job to specify the symbol ®NAME for the region qualifier in the name of the data set.

Move and edit the appropriate data set statements before you create copies of the DFHDEFDS job for each CICS region. When you run DFHISTAR to create the new copies of the DFHDEFDS job, it substitutes your values for the CICS region qualifier (®NAME) and index (&INDEX) into the data set names.

For example: If you intend using a copy of the DFHCSD data set for each CICS region, copy the job steps DELCSD, DEFCSDD, and INITCSD from the DFHCOMDS job to the DFHDEFDS job. Also, add the symbol ®NAME for the qualifier to the name of the DFHCSD data set to give

&DSINDEX.CICS®NAME.DFHCSID.DFHDEFDS. If you edit DFHISTAR to select the DFHDEFDS job to be copied, and specify the following DSINFO parameter:

```
DSINFO userid.CICSTS56.CICS H3P060 3390 IDA .
```

when you run the DFHDEFDS job, it creates the DFHCSID data set called `userid.CICSTS56.CICS.CICSIDA.DFHCSID` for the CICS region identified by the qualifier `IDA`. If you change the `SELECT` and `DSINFO` parameters of `DFHISTAR`, to specify an appropriate new job name and qualifier for another CICS region, you can create several copies of the `DFHDEFDS` job to create `DFHCSID` and `SYSIN` data sets for each CICS region.

DFHDEFDS job for CICS region data sets

The `DFHDEFDS` job deletes and re-creates copies of data sets that are required for each CICS region.

For details about creating these data sets, see [Defining data sets](#).

Table 24. Data sets created by DFHDEFDS	
Data set	Description
DFHAUXT	Non-VSAM auxiliary trace (A) data set
DFHBRNSF	Bridge
DFHBUXT	Non-VSAM auxiliary trace (B) data set
DFHDMPA	Non-VSAM dump (A) data set
DFHDMPB	Non-VSAM dump (B) data set
DFHDPFMB	The debugging profiles base data set
DFHDPFMP	The debugging profiles path data set
DFHDPFMX	The debugging profiles path data set
DFHGCD	CICS global catalog
DFHHTML	HTML template data set
DFHINTRA	Intrapartition transient data set
DFHLCD	CICS local catalog
DFHLRQ	BTS local request queue
DFHPIDIR	WS-AT directory data set
DFHTEMP	Temporary storage data set
FILEA	Sample program file

Use `DFHISTAR` to create a copy of the `DFHDEFDS` job for each CICS region. Edit `DFHISTAR`, specifying the parameters **DSINFO** and **SELECT**, and run it once for each region.

In `DFHISTAR`, specify the following parameters:

- **SELECT DFHDEFDS newname** to specify the new name by which the copy of the `DFHDEFDS` job is to be known.
- **DSINFO** to specify the following details of the data sets for each CICS region:
 - The high-level index (*dsindex*)
 - The serial number of the volume (*volume*)
 - The unit type of the volume (*disktype*)
 - The region qualifier (*qualifier*)

The format of the data set names is:

```
dsindex.CICSqualifier.dsname
```

dsindex

Is the high-level index for the data sets, specified on the DSINFO parameter of DFHISTAR. The default is *hlq*.

qualifier

Is the region qualifier for the data sets that are used by this CICS region, specified on the DSINFO parameter of DFHISTAR. The default is no qualifier.

dsname

Is the name of the data set being defined.

For example, the default name for the CICS local catalog is *hlq.CICS.DFHLCD*.

The DFHDEFDS job comprises the following job steps:

1. **DELETE** deletes any existing copies of the data sets.
2. **DEFINE** defines the clusters for the data sets.
3. **INITDP** initializes the debugging profiles base data set.
4. **DEFAULT** defines the alternate index for the debugging profiles data set.
5. **BLDDP** builds the alternate index for the debugging profiles data set.
6. **INITGCD** initializes the CICS global catalog.
7. **INITLCD** initializes the CICS local catalog.
8. **DEFTRACE** defines the trace data sets.
9. **DEFDUMP** defines the dump data sets.
10. **DEFHTML** defines the CICS default HTML data set.
11. **LOADFILE** loads the sample data into the FILEA data set.
12. **LOADHTML** loads HTML templates for supplied transactions.

DFHCMACI job for creating the messages data set

The DFHCMACI job deletes and re-creates the CICS messages data set DFHCMACD. This data set is used by the CMAC transaction.

The DFHCMACI job comprises the following job steps:

1. **CMACDEF** deletes any existing copies of the DFHCMACD data set and defines the VSAM cluster for the CICS message data set dsindex.DFHCMACD, where dsindex is defined by the DSINFO parameter of DFHISTAR.
2. **CMACLOAD** loads the CICS message data set with data from the CICS-supplied file, DFHCMACD, in the *hlq.SDFHMSG* target library.

For detailed information about setting up this data set, see [Defining data sets](#).

Defining the sample applications data sets

CICS provides a range of samples that you can use to help develop your own applications and test various CICS functions; for example, as an aid to verifying that CICS has installed correctly.

Before you can use some of these samples, you must create the data sets that they use, and make them available to your CICS region, as described in these topics. Do not create these data sets, unless you intend using the associated sample applications.

The CUA text-level application

You can use this sample application to demonstrate BMS support for the Common User Access (CUA) interface. The application uses an action bar, with associated pull-downs, pop-ups, and help panels.

The application programs demonstrate how to code COBOL programs to display, overlay, and remove CUA style windows.

Creating the data sets for the CUA text-level application

To create the data sets that the CUA text-level application requires, submit the DFH0JCUS and DFH0JHLP jobs, which are installed in the *hlq.XDFHINST* library.

Making the data sets available to CICS

You can cause CICS to dynamically allocate the files for these data sets and open them after CICS initialization by installing the sample resource definitions in the group DFH\$CTXT.

If no DD statement exists for these data sets in the CICS startup job stream, the files are allocated to the data sets with DSNAMES that are specified in the resource definitions: *hlq.SAMPLE.DFHCTCUS*, *hlq.SAMPLE.DFHCTHLP*, and *hlq.SAMPLE.DFHCTAIX*, for the data sets and the alternate index.

Alternatively, you can add DD statements for the data sets to your CICS startup job, which causes CICS to use the DSNAMES specified on the DD statements instead of those in the resource definitions.

The FILEA sample application programs

The FILEA sample application programs comprise four sets of command-level application programs that operate on the sample VSAM file, FILEA.

Each of the four programming languages that are supported (Assembler, C, COBOL, and PL/I) has its own set. These programs show basic functions, such as inquire, browse, add, and update, that can serve as a framework for your own first programs. They were written before publication of the Common User Access guidelines.

Creating the data set for FILEA

When you submit the DFHDEFDS job, a copy of the data set that the FILEA application requires is created. This data set is installed in the *hlq.SDFHINST* library.

Making the data set available to CICS

When you tailor the CICS installation-related jobs, a DD statement for the FILEA data set is added to the CICS IVP jobs and the DFHSTART procedure. If you want CICS to dynamically allocate the data set and open the file, remove the DD statement and install a FILE resource definition with an appropriate DSNAMES; for example, as supplied in the group DFH\$FILA. For more information about CICS installation-related jobs, see [“Tailoring the CICS-supplied skeleton jobs” on page 167](#).

The CICS Application Programming Primer sample application

You can use this sample application to demonstrate the design and programming of a standard CICS application.

It provides online inquiry and maintenance for a sample customer credit file in a department store. The application uses VSAM files and 3270 display and printer terminals. It was written before publication of the Common User Access guidelines, and provides similar function, without CUA support, as the CUA sample application.

Creating the data sets for the Primer application

To create the data sets that the Primer sample application requires, edit and submit the sample job.

```

//DEFACCTF JOB (accounting parameters),MSGCLASS=A,MSGLEVEL=(1,1),
//          CLASS=A,NOTIFY=userid
//*
//*****
//*          CICS sample jobs to define ACCT files
//*
//* This job deletes and defines the following data sets for the
//* ACCT sample described in the CICS Application Programming Primer
//*
//* STEPS:
//*   . DELETE AND DEFINE
//*     - DELETE/DEFINE THE CLUSTERS FOR:
//*       . CICSTS56.CICS.ACCTFILE
//*       . CICSTS56.CICS.ACIXFILE
//*
//* THE HIGH-LEVEL-QUALIFIER(S) OF THE DATASETS: CICSTS56.CICS
//* THE VOLUME SERIAL                          CICS41
//* THE UNIT TYPE                              3390
//*
//*****
//DELETE    EXEC PGM=IDCAMS,REGION=1M
//SYSPRINT DD SYSOUT=*
//SYSIN     DD *
DELETE CICSTS56.CICS.ACCTFILE
DELETE CICSTS56.CICS.ACIXFILE
SET MAXCC=0
/*
//DEFINE    EXEC PGM=IDCAMS,REGION=1M
//SYSPRINT DD SYSOUT=*
//SYSIN     DD *
/*          */
DEFINE CLUSTER(NAME(CICSTS56.CICS.ACCTFILE) -
              KEYS(5 0) -
              INDEXED -
              RECORDSIZE(383 383) -
              REC(80) -
              SHR(2 3) -
              VOLUMES(CICS41)) -
DATA(NAME(CICSTS56.CICS.ACCTFILE.DATA) -
UNIQUE) -
INDEX(NAME(CICSTS56.CICS.ACCTFILE.INDEX) -
UNIQUE)
/*          */
DEFINE CLUSTER(NAME(CICSTS56.CICS.ACIXFILE) -
              KEYS(17 0) -
              INDEXED -
              RECORDSIZE(63 63) -
              REC(80) -
              SHR(2 3) -
              VOLUMES(CICS41)) -
DATA(NAME(CICSTS56.CICS.ACIXFILE.DATA) -
UNIQUE) -
INDEX(NAME(CICSTS56.CICS.ACIXFILE.INDEX) -
UNIQUE)
/*
//*

```

Figure 18. Example JCL to create the Primer sample data sets

Making the data sets available to CICS

You can cause CICS to dynamically allocate the files for these data sets and open them on first reference by installing the sample resource definitions in the group DFH\$ACCT.

If no DD statement exists for these data sets in the CICS startup job stream, the files are allocated to the data sets with DSNAMES that are specified in the resource definitions: *hlq.ACCTFILE* and *hlq.ACIXFILE*. Alternatively, you can add DD statements for the data sets to your CICS startup job, which causes CICS to use the DSNAMES specified on the DD statements instead of those in the resource definitions.

CICS resource definitions for CICSplex SM

You must review or change some definitions when you upgrade the CICS system definition file (CSD) for CICSplex SM.

Dynamic creation of CICS resource definitions

CICS can dynamically define the CICS resource definitions required to allow the CICSplex SM CMAS, MAS agent, or WUI server to be started.

This includes the COLM and COVC transactions. If these transactions are invoked, or the equivalent **CPSMCONN** system initialization parameter is set, CICSplex SM dynamically creates further definitions as required.

The dynamically created resource definitions and their attributes can be found in the following members of the SEYUSAMP sample library:

- EYU\$CDEF contains the default resource definitions for a CMAS.
- EYU\$MDEF contains the default resource definitions for a MAS.
- EYU\$WDEF contains the default resource definitions for a WUI server.

If the dynamically created CICS resource definitions need to be changed, see [“Overriding the dynamically created CICS resource definitions for CICSplex SM” on page 176.](#)

The CICS system definition data set (CSD) in use needs to be appropriate for the release of CICS in use. For example, upgrade the CSD to the latest release and depending on the CICS release of the region, add the CICS DFHCOMPx CSD compatibility groups to the CICS group list as necessary. See the appropriate CICS Upgrading documentation for details.

Overriding the dynamically created CICS resource definitions for CICSplex SM

You do not have to update your CICS system definition file (CSD) to obtain the default resource definitions supplied by CICSplex SM.

You do not typically run user transactions in a CMAS. However, if you do choose to define your own transactions to the CMAS, be aware that transaction IDs used by CICSplex SM in the CMAS have no specific format. To avoid conflict between your names and those that are used by CICSplex SM, review the transactions that are defined in EYU\$CDEF. For a list of these transactions, see [Chapter 11, “Default CICS resource definitions for CICSplex SM,” on page 225.](#)

If you do want to modify a definition, you can do so using the required definition from the EYU\$CDEF (CMAS), or EYU\$WDEF (WUI) samples.

For MAS definitions use the following:

- EYU\$M690: This sample includes the default CICS resource definitions used for a MAS agent at the CICS TS 5.6 level and the CICS region at CICS R690.
- EYU\$M700: This sample includes the default CICS resource definitions used for a MAS agent at the CICS TS 5.6 level and the CICS region at CICS R700.
- EYU\$M710: This sample includes the default CICS resource definitions used for a MAS agent at the CICS TS 5.6 level and the CICS region at CICS R710.
- EYU\$M720: This sample includes the default CICS resource definitions used for a MAS agent at the CICS TS 5.6 level and the CICS region at CICS R720.
- EYU\$MDEF: This sample includes the default CICS resource definitions used for a MAS agent at the CICS TS 5.6 level and the CICS region at CICS R730.

Copy the required definitions from the appropriate sample into a DFHCSDUP job, and make the required changes, including specifying a CSD group. Specify this CSD group in an appropriate GRPLIST for the CMAS, MAS, or WUI server. The region is then COLD or INITIAL started, as required, to process the change.

Some CICS resource definitions are enforced. If an inappropriate definition is found, a message is produced during CICSplex SM initialization. If CICSplex SM cannot dynamically correct the error, CICSplex SM initialization can fail.

You can update the CSD for CICSplex SM if CMAS journaling is required; see [CMAS journaling](#).

Another reason you might want to update the CSD for CICSplex SM is if your WUI servers require additional import or export TDQs to make copying WUI view or menu definitions easier. You can use the COVI and COVE definitions from EYU\$WDEF as a model. For example, if the WUI servers can access the same MVS data sets, make your updates in this way:

1. Copy the COVI and COVE definitions from EYU\$WDEF into a DFHCSD job.
2. Rename the COVE TDQ; for example, to MYEX. This TDQ definition must have the data set name specified. Also, this data set must be preallocated using data set characteristics.
3. Include the MYEX definition in a group that is to be used by the exporting WUI.
4. Install the MYEX TDQ, after the definition has been made on the CSD in one of these ways:
 - A COLD start of the exporting WUI, if the group is in the exporting WUI's group list, or
 - CEDA INSTALL, if the WUI server is running.
5. Rename the COVI TDQ; for example, to MYIM. This TDQ definition must have the same data set name specified on the MYEX TDQ definition.
6. Include the MYIM definition in a group that is to be used by the importing WUI.
7. Install the MYIM TDQ, after the definition has been made on the CSD in one of these ways:
 - A COLD start of the importing WUI, if the group is in the importing WUI's group list, or
 - CEDA INSTALL, if the WUI server is running.
8. Use COVC, on the exporting WUI, to export the required definitions to the MYEX TDQ. Wait for the Export to complete.
9. Use COVC, on the importing WUI, to import the required definitions to the MYIM TDQ.

If the WUI servers cannot access the same data sets, the export data set must be copied from the exporting system (after the COVC Export is complete) to the importing system (before the COVC Import is run).

The following DFHCSDUP SYSIN control statements contain the default definitions, copied from EYU\$WDEF, with the modification to the default definitions, to allow the same data set to be used to export to a copy of the COVE output TD queue and the COVI input TD queue to import into another WUI. The modification is in bold type.

```
//CSDUP EXEC PGM=DFHCSDUP
//STEPLIB DD DSN=cics.index.SDFHLOAD,DISP=SHR
//DFHCSD DD DSN=cics.dfhcscd,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
*-----*
* Create TDQUEUE definitions for import/export *
*-----*
DELETE GROUP(group_name)
DEFINE TDQUEUE(MYIM) GROUP(group_name)
DESCRIPTION(CPSM WUI IMPORT DATASET)
TYPE(EXTRA) DATABUFFERS(1) DDNAME(EYUCOVI) DSNAME(hlq.wui.screens)
ERROROPTION(IGNORE) OPENTIME(DEFERRED) TYPEFILE(INPUT)
RECORDSIZE(32000) BLOCKSIZE(0) RECORDFORMAT(VARIABLE)
BLOCKFORMAT(BLOCKED) DISPOSITION(SHR)
*
DEFINE TDQUEUE(MYEX) GROUP(group_name)
DESCRIPTION(CPSM WUI EXPORT DATASET)
TYPE(EXTRA) DATABUFFERS(1) DDNAME(EYUCOVE) DSNAME(hlq.wui.screens)
ERROROPTION(IGNORE) OPENTIME(DEFERRED) TYPEFILE(OUTPUT)
RECORDSIZE(32000) BLOCKSIZE(0) RECORDFORMAT(VARIABLE)
BLOCKFORMAT(BLOCKED) DISPOSITION(SHR)
ADD GROUP(group_name) LIST(list_name)
```

where:

STEPLIB

Identifies `cics.index.SDFHLOAD` as the CICS load library that contains the DFHCSDUP module.

DFHCSD

Identifies `cics.dfhcscd` as the CICS CSD file to be updated.

SYSIN

Shows the SYSIN input for import and export transient data queues.

group_name

Identifies the name of your group.

list_name

Identifies the name of a CSD list, which you include on the GRPLIST system initialization parameter for WUI server WUI A.

For information about TD queue definitions see [Transient data queue definitions for the WUI](#).

A return code of 4 is expected from this run of DFHCSDUP because, before adding the designated group to the CSD, the job attempts to delete any group with the same name.

You can amend the supplied definitions for a CMAS and MAS in the same way. Run DFHCSDUP including the modified resource definition from the sample EYU\$CDEF, for a CMAS, or EYU\$MDEF, for a MAS, in a CSD list that you use when starting your system.

Amending log stream names in a CMAS

If the log stream names used by the CICS-supplied DFHLGMOD group are not appropriate for your environment, copy group DFHLGMOD to a new group, where you can make your amendments. Add the new group to a CSD list and ensure that the list is included for your CMAS, using the SIT GRPLIST system initialization parameter.

See [Defining the logger environment for CICS](#) for details on how to define log streams.

Do not operate the CMAS with log streams that are defined as DUMMY. Dummy log streams can cause problems when recovering the CSD or CICSplex SM data repository (EYUDREP). See [CMAS journaling](#) for details about the various CMAS journaling options that you can activate.

Adding CICS support for programming languages

You must add runtime support for the programming languages used with the CICS command level (EXEC) programming interface before you can install your application programs.

About this task

To write CICS application programs that request CICS services through the command-level application programming interface (API), you can use Assembler language, C and C++, COBOL, or PL/I.

CICS provides the support required to run application programs written in Assembler language, and Language Environment provides the required support for all the other languages.

The programming guidance documentation expects that your CICS system is using the services of Language Environment, which provides a common runtime environment for IBM implementations of Assembler and those high-level languages (HLLs) supported by CICS, namely COBOL, PL/I, C, and C++. Supported compilers are listed in [Changes to CICS support for application programming languages](#).

Installing Language Environment support

Language Environment support is provided by runtime libraries that establish a common runtime environment for application programs compiled by high-level languages. All programs compiled by a

high-level language, whether by a Language Environment-conforming compiler or not, must be run under CICS-Language Environment support.

About this task

The CICS-Language Environment interface is initialized automatically when CICS performs the following tasks:

1. Loads the Language Environment interface modules, CEECCICS, CEEPIPI, and CEECTCB, from STEPLIB.
2. Successfully calls the CEECCICS module to initialize the interface.

Language Environment initialization takes place during CICS startup, when CICS issues the message DFHAP1203I *applid* Language Environment is being initialized. The CEECCICS module is loaded, followed by a partition initialization call, before the start of second phase PLT processing. If Language Environment cannot successfully complete the initialization of all languages supported by CICS, or can initialize only some of them, it issues messages to the MVS console. If Language Environment initialization fails completely, it might be because the CEECCICS module was not loaded, or something failed during the loading of a particular language routine.

Installing CICS support for Language Environment

To enable Language Environment support to be installed correctly by CICS, specify storage requirements and ensure that you have the required modules and resource definitions.

Procedure

1. Specify enough storage for the extended read-only dynamic storage area (ERDSA) to run CICS and Language Environment together. They require a minimum of 3,500 KB. To this minimum, add an amount of storage sufficient for your own requirements.
2. Ensure the CICS-Language Environment interface module, CEECCICS, and the Language Environment modules CEEPIPI and CEECTCB are installed in an APF-authorized library defined in the STEPLIB concatenation in the CICS startup JCL. You can do this by including the Language Environment SCEERUN library in an APF-authorized library in the STEPLIB concatenation of your CICS startup job, for example, in the CICSTS56.CICS.SDFHAUTH library, or in an APF-authorized library in the MVS LNKLSTnn concatenation.
3. Ensure that the CEE group is removed from any CICS startup group list named in the **GRPLIST** system initialization parameter. If CEE group definitions have been converted to bundle definitions or CICSplex SM BAS definitions, these should also be removed. This is because with the CICS system autoinstall function, CICS dynamically installs Language Environment definitions when required, rather than installing a large number of definitions during startup.

Note: Language Environment requires some definitions to be preloaded, such as the event handlers for Language Environment, which are named CEEEVnnn. So, during CICS initialization, Language Environment attempts to load all its 18 event handlers. As a result, all the 18 Language Environment event handler definitions are automatically installed; but later in initialization, before PLT phase 2, CICS discards, where possible, event handlers that are not required in a CICS environment.

The following event handlers are retained because they are required in a CICS environment:

- CEEEV003 for C and C++
- CEEEV004 and CEEEV005 for COBOL
- CEEEV006 for Debug Tool
- CEEEV010 and CEEEV011 for PL/I

In addition, two unwanted handlers, CEEEV007 for Fortran and CEEEV012 for OE sockets, will remain because Language Environment has successfully loaded the two event handler programs.

4. Define the Language Environment transient data destinations, CESE and CESO (DD names CEEMSG and CEEOUT).

The CICS-supplied resource definition group DFHDCTG, in the CICS system definition data set (CSD), contains entries for CESE and CESO.

For information about the attributes required for Language Environment transient data destinations, see the [z/OS Language Environment Programming Guide](#).

5. Define the Language Environment runtime libraries on the CICS STEPLIB and DFHRPL DD statements:

a) Add the following libraries to STEPLIB or to a library in the MVS LNKSTnn concatenation:

- SCEERUN, which contains CEECCICS and CEECTCB
- SCEERUN2, which contains support that is required for the IBM Java Virtual Machine (JVM) and also support for other programming languages

Both the libraries, SCEERUN and SCEERUN2, must be APF-authorized.

b) Add the SCEECICS, SCEERUN2, and SCEERUN libraries to DFHRPL.

For example:

```
/*          CICS APF-authorized libraries
//STEPLIB DD DSN=hlq.CICS.SDFHAUTH,DISP=SHR
//          DD DSN=hlq.LE.SCEERUN2,DISP=SHR
//          DD DSN=hlq.LE.SCEERUN,DISP=SHR
/*          CICS load libraries
//DFHRPL  DD DSN=hlq.CICS.SDFHLOAD,DISP=SHR
//          DD DSN=hlq.LE.SCEECICS,DISP=SHR
//          DD DSN=hlq.LE.SCEERUN2,DISP=SHR
//          DD DSN=hlq.LE.SCEERUN,DISP=SHR
```

Use only these Language Environment runtime libraries for *all* your high-level language application programs.

Language Environment support for COBOL

Language Environment is a prerequisite for application programs written in COBOL.

[Changes to CICS support for application programming languages](#) lists the COBOL compilers that are supported by CICS Transaction Server for z/OS, Version 5 Release 6 , and their service status on z/OS. For information about Language Environment, see [z/OS Language Environment Customization](#).

To run COBOL application programs:

- Install support for Language Environment, ensuring that CICS can initialize Language Environment during startup.
- Install resource definitions for your programs with the LANGUAGE attribute specified as LANGUAGE(COBOL), or leave the language blank.

For your application programs, CICS can create and install program resource definitions automatically, or you can create them specifically in the CICS system definition file (CSD), and install them by using the **GRPLIST** system initialization parameter or **CEDA INSTALL** command. For more information about installing program resource definitions, see [Resource management transaction CEDA commands](#).

Language Environment support for C and C++

Language Environment is a prerequisite for application programs compiled using IBM C/C++ for MVS or SAA AD/Cycle C/370 compilers. Language Environment incorporates the runtime libraries required for both these C language compilers.

For information about Language Environment, see [z/OS Language Environment Customization](#).

To run C application programs in CICS:

- Install support for Language Environment, ensuring that CICS can initialize the Language Environment environment during startup.
- Install resource definitions for your programs with the LANGUAGE attribute specified as LANGUAGE(C) or leave the language blank.

For information about installing program resource definitions, see [Resource definition installation](#).

CICS supports application programs written in C++ that meet these requirements:

- Are compiled using the IBM C/C++ for MVS compiler (5655-121)
- Run with the Language Environment runtime libraries

If you use Version 3 Release 2, or later, of the C/C++ compiler to compile a C++ program, specify the CXX parameter when options are passed to the compiler; otherwise, the C compiler is invoked. Do not specify CXX if a C program is to be compiled. For further information, see [Customized JCL and the CXX format in z/OS XL C/C++ Compiler and Runtime Migration Guide for the Application Programmer](#).

Language Environment support for PL/I

Language Environment is a prerequisite for application programs compiled using the IBM Enterprise PL/I for z/OS compiler. Language Environment incorporates the runtime libraries required for PL/I compilers.

PL/I support is also required if you use the web services support in CICS; in particular, it is required if you use the supplied SOAP 1.1 and SOAP 1.2 message handler programs.

To run CICS PL/I application programs:

- Install support for Language Environment, ensuring that CICS can initialize the Language Environment environment during startup.
- Install resource definitions for the programs with the LANGUAGE attribute specified as LANGUAGE(PLI) or leave blank.

For information about Language Environment, see [z/OS Language Environment Customization](#). For information about installing program resource definitions, see [Resource definition installation](#).

Language Environment support for Java

Language Environment is a prerequisite for Java programs that run in JVMs in CICS. However, unlike the other languages, Java programs do not require the CICS-Language Environment interface.

Java programs run with Language Environment support using MVS services, not CICS services. Java programs require the Language Environment support provided by the SCEERUN and SCEERUN2 libraries only, which can either be defined in the CICS STEPLIB or included in the MVS linklist. The SCEERUN and SCEERUN2 libraries must also be added to DFHRPL.

Enabling TCP/IP in a CICS region

The CICS sockets domain provides TCP/IP support, with network services supplied by z/OS.

About this task

The sockets domain provides listener support and outbound sockets support.

The listener

The listener monitors specified TCP/IP ports for incoming requests. You configure the listener with a TCPIPSERVICE resource definition to listen on a specific TCP/IP port and to attach a specified request receiver transaction to handle each connection. When the connection has been established between a client program and a particular request receiver, all subsequent requests from the client program over that connection flow to the same request receiver. The listener supports user applications initiated by TCP/IP services for the following protocols:

External Call Interface (ECI)

The ECI allows client applications to use a TCP/IP connection directly to a CICS region. The External Presentation Interface (EPI) and External Security Interface (ESI) are not supported.

Hypertext Transfer Protocol (HTTP)

HTTP messages are received and sent over the Internet, using CICS web support. See [HTTP request and response processing for CICS\(r\) as an HTTP client](#) for information about the transmission of HTTP messages on the web.

IP interconnectivity protocol (IPIC)

[Intercommunication facilities available using IPIC](#)

Outbound sockets

Outbound socket support allows CICS to open sockets and communicate using one of the supported TCP/IP networking protocols, for example, HTTP or IPIC.

Using TCP/IP in a CICS region

To use TCP/IP in a CICS region, install Communications Server, set and define system initialization parameters, and provide resource definitions. Additional tasks are required if you use SSL authentication.

1. Install Communications Server. Make ports belonging to Communications Server available for use by the CICS region involved.
2. Set the **TCPIP** system initialization parameter to YES.
3. Provide TCPIPSERVICE resource definitions for each active port and the type of service associated with it. The CICS TCP/IP listener is activated for the specified ports when the TCPIPSERVICE is installed, if you also specify **TCPIP(YES)**.
4. If Secure Sockets Layer (SSL) authentication is used, you must define the KEYRING system initialization parameter, to identify the RACF key ring containing the keys and X.509 certificates used in the SSL handshake.

The TCPIPSERVICE resource definitions are for use only with the CICS-provided TCP/IP services, and are not related to the z/OS Communications Server IP CICS Sockets interface. The TCP/IP Socket Interface for CICS is supplied with z/OS Communications Server, which is an integral part of z/OS and does not use the CICS Sockets domain.

A TCPIPSERVICE supports either one specific IP address or all IP addresses (INADDR_ANY). Therefore, if two CICS regions are required to listen on the same port at the same IP address, you must use a form of network load balancing, for example, TCP/IP port sharing.

Installing MRO support

CICS multiregion operation (MRO) enables CICS regions that are running in the same z/OS image, or in the same z/OS sysplex, to communicate with each other. MRO does not support communication between a CICS system and a non-CICS system such as IMS.

The external CICS interface (EXCI) uses a specialized form of MRO link to support communication between z/OS batch programs and CICS.

MRO does not require ACF/Communications Server or SNA networking facilities. The CICS support that enables region-to-region communication is called interregion communication (IRC). IRC is implemented in three ways:

1. Through support in CICS terminal control management modules and by use of a CICS-supplied interregion program, DFHIRP, loaded in the z/OS link pack area. DFHIRP is invoked by a type 3 supervisory call (SVC).
2. By z/OS cross-memory services, which you can select as an alternative to the CICS type 3 SVC mechanism. Here, DFHIRP only opens and closes the interregion links.
3. By the cross-system coupling facility (XCF) of z/OS. XCF/MRO is required for links between CICS regions in different z/OS images of an z/OS sysplex. CICS selects XCF/MRO dynamically for such links, if available.

For information about the design and implementation of interregion communication, and about the benefits of cross-system MRO, see [Introduction to CICS intercommunication](#).

To install support for MRO, complete the following steps:

1. Define CICS as a z/OS subsystem. Multiregion operation with CICS requires z/OS Subsystem Interface (SSI) support. To obtain this support, you define CICS as an operating system subsystem.
2. Install the current versions of the DFHIRP and DFHCSVC modules in the LPA.
3. Specify appropriate system initialization parameters to enable MRO for each CICS region startup. To help you get started with MRO, a CICS job and a CICS startup procedure are supplied on the CICS distribution volume. For each MRO region, you must also create the CICS system data sets needed.

To use cross-system MRO (XCF/MRO):

4. Install the required sysplex hardware and software.
5. Define the z/OS images as systems in an XCF sysplex.

To use the MRO support:

6. Define and install the MRO connections appropriate to your CICS environment.

Provided you complete these steps, you can use MRO to communicate with all supported levels of CICS.

If MRO is used to communicate between different releases of CICS, the function provided on any connection is that of the lower-level release.

Installing the modules DFHIRP and DFHCSVC in the LPA

You must install the DFHIRP and DFHCSVC modules to enable your regions to communicate by MRO.

Procedure

1. Install the current versions of the DFHIRP and DFHCSVC modules into the LPA, as described in [“Installing CICS modules in the MVS link pack area”](#) on page 98.

Notes:

- If you are running CICS with MRO at different release levels, all communicating CICS regions must use the latest DFHIRP module and the latest SVC module, DFHCSVC, on the same z/OS image. If a previous version of CICS is already installed on this z/OS image, do not use the dynamic LPA function to replace DFHIRP unless you have a strategy whereby all users of DFHIRP on the z/OS image that is being upgraded can be quiesced. For more information about dynamically updating DFHIRP, see [Upgrading MRO](#). Failure to shut down all users of DFHIRP during the upgrade process can cause incompatibility between control blocks that results in abends that require an IPL of the z/OS image.
 - Multiregion operation requires the CICS interregion communication modules to run in supervisor state to transfer data between different regions. CICS achieves this by using a normal supervisor call to this startup SVC routine, which is in the pregenerated system load library (CICSTS56.CICS.SDFHLOAD).
2. Define the SVC module, DFHCSVC, to z/OS, as described in [“Installing the CICS SVCs”](#) on page 84.

Installation requirements for XCF/MRO

For MVS images to communicate using MRO between CICS regions on different z/OS images, the z/OS images must be joined in a sysplex.

For the hardware and software that are required for z/OS systems in a sysplex, see the [“Program Directories”](#) on page 2.

A sysplex consists of multiple MVS images, coupled together by hardware elements and software services. In a sysplex, MVS images provide a platform of basic services that multisystem applications like CICS can exploit. As the workload for an installation grows, additional MVS images can be added to the sysplex so that the installation can meet the needs of the greater workload.

Usually, a specific function (one or more modules or routines) of the MVS application subsystem (such as CICS) is joined as a *member* (a member resides on one MVS image in the sysplex), and a set of related members is the *group* (a group can span one or more of the MVS images in the sysplex). A group is a

complete logical entity in the sysplex. To use XCF to communicate in a sysplex, each CICS region joins an XCF group as a member, using services provided by DFHIRP.

For information about installing and managing MVS systems in a sysplex, see [z/OS MVS Setting Up a Sysplex](#).

Generating XCF/MRO support

There are a number of steps you must complete to generate XCF/MRO support.

Procedure

1. Ensure that the version of DFHIRP in the extended link pack area (ELPA) is at the required level for all the MVS images containing CICS systems to be linked.

Depending on the versions of CICS installed in the MVS images participating in XCF/MRO, the versions of DFHIRP installed in the images can be different. The DFHIRP module must be from the most current CICS release in the image, or higher.

The CICS TS for z/OS, Version 5.6 DFHIRP module, which is required for multiple XCF group support, can be used only with z/OS, Version 2 Release 3 or later.

2. Ensure that each CICS APPLID is unique within the sysplex.
3. Ensure that the value of the **MAXMEMBER** MVS parameter, used to define the XCF couple data sets, is high enough to cater for the largest CICS XCF group.
The maximum size of any XCF group within a sysplex is limited by this value. The theoretical maximum size of any XCF group is 2047 members.

External CICS interface (EXCI) users that use an XCF/MRO link also join an XCF group. You must therefore set the value of **MAXMEMBER** high enough to allow all CICS regions and EXCI XCF/MRO users in the largest CICS XCF group to join the group concurrently.

To list the CICS regions and EXCI users in an XCF group, use the MVS **DISPLAY** command. For example, to list the CICS regions and EXCI users in the DFHIR001 XCF group, use the command:

```
DISPLAY XCF, GROUP, DFHIR001, ALL
```

Do not rely on the default value of **MAXMEMBER**, which might be too low to allow all the CICS regions and EXCI users in the largest XCF group to join the group. This is especially important if you have only a few CICS XCF groups.

Likewise, do not set a value much larger than you need, because this results in large couple data sets for XCF. The larger the data set, the longer it takes to locate entries.

Make the value of **MAXMEMBER** 10-15 greater than the combined number of CICS regions and EXCI users in the largest CICS XCF group.

Results

Each CICS region joins an XCF group when it logs on to DFHIRP. Its member name is its APPLID (NETNAME) used for MRO partners. The XCF group name is specified on the **XCFGROUP** system initialization parameter. If **XCFGROUP** is not specified, the XCF group name defaults to DFHIR000.

At connect time, CICS invokes the IXCQUERY macro to determine whether the CICS region being connected to resides in the same MVS image. If it does, CICS uses IRC or XM as the MRO access method, as defined in the connection definition. If the partner resides in a different MVS image, CICS uses XCF as the access method, regardless of the access method defined in the connection definition.

CICS regions can use MRO or XCF/MRO to communicate only with regions in the same XCF group. Members of different XCF groups cannot communicate using MRO, or XCF/MRO, even if they are in the same MVS image.

Defining z/OS images as systems in an XCF sysplex

To use XCF/MRO, you define all participating z/OS images as part of the same sysplex.

In a parallel sysplex, where MRO communication between z/OS images is by XCF/MRO, the DFHIRP programs installed in the different z/OS images can be at different release levels. However, DFHIRP must be installed from the highest release of CICS running in an z/OS image. For example, a CICS TS V5.6 DFHIRP can communicate with another DFHIRP across XCF/MRO, but the CICS regions running in the z/OS with the CICS TS V5.6 DFHIRP cannot be later than CICS TS V5.6 .

For more information, see [“MVS cross-system MRO definitions” on page 123](#).

Enabling MRO for CICS startup

For each CICS region that is to use MRO, you must specify ISC=YES to include the intersystem communication program, DFHISP.

If you want a CICS region to establish MRO communication during startup, also specify YES on the IRCSTART system initialization parameter.

Alternatively, after your CICS region is running, you can establish MRO communication by using the CEMT SET IRC OPEN command or the EXEC CICS SET IRC OPENSTATUS(cvda) command.

Either method establishes MRO communication with every CICS region that meets the following criteria:

1. The MRO connection is currently active.
2. The MRO connection is defined to your region by CONNECTION and SESSIONS definitions that are installed from the CICS system definition file (CSD). To establish MRO communication between two CICS regions, the installed CONNECTION definition must specify INSERVICE (YES) in both regions.

Adding communications support between systems

You can include communications to other systems in a CICS region by using either ISC over SNA or IP interconnectivity (IPIC).

MRO can be used only for CICS-to-CICS connections in the same z/OS image or z/OS sysplex. If you connect CICS to both CICS and non-CICS systems, and the remote systems can be inside or outside the local z/OS sysplex, CICS offers intersystem communication over SNA (ISC over SNA) and intersystem communication over TCP/IP. Intersystem communication over TCP/IP is known as IP interconnectivity (IPIC).

Unlike MRO, no special z/OS operating system requirements apply to these methods of communication.

Activating intersystem communication over z/OS Communications Server

To provide the necessary protocols to support communication between CICS regions that are in different z/OS images, or in different z/OS sysplexes, ISC over SNA uses the ACF/Communications Server access method.

You can also use ISC over SNA in the same CPC, through the application-to-application facilities of ACF/Communications Server.

You must include the following management programs in your CICS regions, by specifying the system initialization parameters that are given in parentheses:

- DFHISC – the intersystem communication program (ISC=YES).
- DFHTCP – the terminal control program (TCP=YES is the default).

Establishing ISC over SNA

Intersystem communication over SNA requires z/OS Communications Server support. CICS regions cannot communicate until they have established the z/OS Communications Server connection.

Specify VTAM=YES as a system initialization parameter.

If the z/OS Communications Server is running during CICS initialization, CICS opens the z/OS Communications Server ACB.

If the z/OS Communications Server is started after CICS, opening the z/OS Communications Server ACB fails, and you must open it using the z/OS Communications Server CEMT SET VTAM OPEN command when the z/OS Communications Server is available.

ACF/SNA definition for CICS

When you define your CICS system to ACF/SNA, you need to include a number of operands in the SNA APPL statement.

About this task

MODETAB=logon-mode-table-name

This operand names the SNA logon mode table that contains your customized logon mode entries. See “ACF/Communications Server LOGMODE table entries for CICS” on page 187 for more information. You can omit this operand if you choose to add your MODEENT entries to the IBM default logon mode table, without renaming it.

AUTH=(ACQ,SPO,VPACE[,PASS])

ACQ is required to allow CICS to acquire LU type 6 sessions. SPO is required to allow CICS to issue the MVS MODIFY *snaname* USERVAR command. VPACE is required to allow pacing of the intersystem flows.

PASS is required if you intend to use the **EXEC CICS ISSUE PASS** command, which passes existing terminal sessions to other SNA applications.

VPACING=number

This operand specifies the maximum number of normal-flow requests that another logical unit can send on an intersystem session before waiting to receive a pacing response.

Take care when selecting a suitable pacing count. Too low a value can lead to poor throughput because of the number of line turnarounds required. Too high a value can lead to excessive storage requirements.

EAS=number

This operand specifies the number of network-addressable units that CICS can establish sessions with. The number must include the total number of parallel sessions for this CICS system.

PARSESS=YES

This option specifies LU type 6 parallel session support.

SONSCIP=YES

This operand specifies session outage notification (SON) support. SON enables CICS, in particular cases, to recover a failed session without requiring operator intervention.

APPC=NO

APPC=NO is required for CICS. This setting is the default. If you do not use APPC=NO, you receive message DFHZC2400E, referencing the SNA return code 1013.

For more information about the SNA APPL statement, see [z/OS Communications Server: SNA Resource Definition Reference](#).

ACF/Communications Server LOGMODE table entries for CICS

For APPC sessions, you can use the MODENAME option of the CICS **DEFINE SESSIONS** command to identify a z/OS Communications Server logmode entry that in turn identifies the required entry in the z/OS Communications Server class-of-service table.

Every modename that you supply, when you define a group of APPC sessions to CICS, must be matched by a z/OS Communications Server LOGMODE name. You need to create entries in the z/OS Communications Server LOGMODE table using the following format:

```
MODEENT LOGMODE=modename
MODEEND
```

An entry is also required for the LU services manager modeset (SNASVCMG):

```
MODEENT LOGMODE=SNASVCMG
MODEEND
```

If you plan to use autoinstall for single-session APPC terminals, additional information is required in the MODEENT entry.

For CICS-to-IMS links that are cross-domain, you must associate the IMS LOGMODE entry with the CICS APPLID, using the **DLOGMOD** or **MODETAB** parameters.

ACF/SNA definition for IMS

When the IMS system is defined to SNA, the following operands should be included on the SNA APPL statement.

About this task

AUTH=(ACQ,VPACE)

ACQ is required to allow IMS to acquire LU type 6 sessions. VPACE is required to allow pacing of the intersystem flows.

VPACING=number

This operand specifies the maximum number of normal-flow requests that another logical unit can send on an intersystem session before waiting to receive a pacing response. An initial value of 5 is suggested.

EAS=number

The number of network addressable units must include the total number of parallel sessions for this IMS system.

PARSESS=YES

This operand specifies LU type 6 parallel session support.

For more information, see [z/OS Communications Server: SNA Programming](#).

ACF/SNA LOGMODE table entries for IMS

IMS allows the user to specify some BIND parameters in a z/OS Communications Server logmode table entry. The CICS logmode table entry must match that of the IMS system.

IMS uses, in order of priority, the mode table entry specified in the following places:

1. The MODETBL parameter of the TERMINAL macro
2. The mode table entry specified in CINIT
3. The DLOGMODE parameter in the VTAMLST APPL statement or the MODE parameter in the IMS / OPNDST command
4. The ACF/SNA defaults.

[Figure 19 on page 188](#) shows an IMS logmode table entry:

```

LU6NEGPS  MODEENT LOGMODE=LU6NEGPS,  NEGOTIABLE BIND
          PSNDPAC=X'01',              PRIMARY SEND PACING COUNT
          SRCVPAC=X'01',              SECONDARY RECEIVE PACING COUNT
          SSNDPAC=X'01',              SECONDARY SEND PACING COUNT
          TYPE=0,                     NEGOTIABLE
          FMPROF=X'12',              FM PROFILE 18
          TSPROF=X'04',              TS PROFILE 4
          PRIPROT=X'B1',              PRIMARY PROTOCOLS
          SECPRROT=X'B1',             SECONDARY PROTOCOLS
          COMPROT=X'70A0',            COMMON PROTOCOLS
          RUSIZES=X'8585',            RU SIZES 256
          PSERVIC=X'060038000000380000000000' SYSMMSG/Q MODEL
MODEEND

```

Figure 19. Example IMS logmode table entry

IMS system definition for intersystem communication

The IMS ISC-related macros that are used in IMS system definition are the COMM, NAME, SUBPOOL, TERMINAL, TYPE, and VTAMPOOL macros.

The COMM macro

The COMM macro identifies the IMS system.

APPLID=name

Specifies the APPLID of the IMS system. This APPLID is usually the name that you specify on the NETNAME option of DEFINE CONNECTION when you define the IMS system to CICS.

However, consider the following points:

- If APPLID on the COMM macro is coded as NONE, the CICS NETNAME option should specify the label on the EXEC statement of the IMS startup job.
- If the IMS system is started as a started task, NETNAME should specify the started task name.
- For an IMS system with XRF, the CICS NETNAME option should specify the USERVAR (that is, the generic applid) that is defined in the DFSHSBxx member of IMS.PROCLIB, not the applid from the COMM macro.

RECANY=(number,size)

Specifies the number and size of the IMS buffers that are used for SNA “receive any” commands. For ISC sessions, the buffer size has a 22-byte overhead. It must therefore be at least 22 bytes larger than the CICS buffer size specified in the SENDSIZE option of DEFINE SESSIONS.

This size applies to all other ACF/SNA terminals attached to the IMS system, and must be large enough for input from any terminal in the IMS network.

EDTNAME=name

Specifies an alias for ISCEDT in the IMS system. For CICS-to-IMS ISC, an alias name must not be longer than four characters.

The NAME macro

The NAME macro defines the logical terminal names associated with the subpool. Multiple LTERMs can be defined per subpool.

COMPT={1|2|3|4}

Specifies the output component associated with this session. The component specified determines the protocol that IMS ISC uses to process messages. An output component defined as SINGLE1 is strongly recommended.

ICOMPT={1|2|3|4}

Specifies the input component associated with this session. When IMS receives a message, it determines the input source terminal by finding the NAME macro that has the matching input component number. A COMPT1 input component must be defined for each session that CICS uses to send START commands.

EDIT={[NO|YES]][,]{ULC|UC}]

The first parameter specifies whether the user-supplied logical terminal edit routine (DFSCNTEO) is to be used.

The second parameter specifies whether the output is to be translated to uppercase (UC) or not (ULC) before transmission.

The SUBPOOL macro

A SUBPOOL macro is required for each session to the remote system.

NAME=subpool-name

Specifies the IMS name for this session. A CICS-to-IMS session is identified by a “session-qualifier pair” formed from the CICS name for the session and the IMS subpool name.

The CICS name for the session is specified in the SESSNAME option of the DEFINE SESSIONS command for the session.

The IMS subpool name is specified to CICS in the NETNAMEQ option of the DEFINE SESSIONS command.

The TERMINAL macro

The TERMINAL macro identifies the remote CICS system to IMS. It is comparable to DEFINE CONNECTION in CICS.

NAME=name

Identifies the CICS node to IMS. It must be the same as the APPLID of the CICS system.

OUTBUF=number

Specifies the size of the IMS output buffer. It must be equal to or greater than 256, and should include the size of any function management headers sent with the data. It must not be greater than the value specified in the RECEIVESIZE option of the DEFINE SESSIONS commands for the intersystem sessions.

SEGSIZE=number

Specifies the size of the work area that IMS uses for deblocking incoming messages. We recommend that you use the size of the longest chain that CICS may send. However, if IMS record mode (VLVB) is used exclusively, you could specify the largest record (RU) size.

MODETBL=name

Specifies the name of the z/OS Communications Server mode table entry to be used. You must omit this parameter if the CICS system resides in a different SNA domain.

OPTIONS=[NOLTWA|LTWA]

Specifies whether Log Tape Write Ahead (LTWA) is required. For LTWA, IMS logs session restart information for all active parallel sessions before sending a syncpoint request. LTWA is recommended for integrity reasons, but it can adversely affect performance. NOLTWA is the default.

OPTIONS=[SYNCESS|FORCESS]

Specifies the message resynchronization requirement following an abnormal session termination. SYNCESS is the default. It requires both the incoming and the outgoing sequence numbers to match (or CICS to be cold-started) to allow the session to be restarted. FORCESS allows the session to be restarted even if a mismatch occurs. SYNCESS is recommended.

OPTIONS=[TRANSRESP|NORESP|FORCRESP]

Specifies the required response mode.

TRANSRESP

Specifies that the response mode is determined on a transaction-by-transaction basis. This is the default.

NORESP

Specifies that response-mode transactions are not allowed. In CICS terms, this means that a CICS application cannot initiate an IMS transaction by using a SEND command, but only with a START command.

FORCRESP

Forces response mode for all transactions. In CICS terms, this means that a CICS application cannot initiate an IMS transaction by using a START command, but only by means of a SEND command.

TRANSRESP is recommended.

OPTIONS=[OPNDST|NOPNDST]

Specifies whether sessions can be established from this IMS system. OPNDST is recommended.

{COMPT1|COMPT2|COMPT3|COMPT4}={SINGLEn|MULn}

Specifies the IMS components for the IMS ISC node. Up to four components can be defined for each node. The input and output components to be used for each session are then selected by the ICOMPT and COMPT parameters of the SUBPOOL macro.

The following types of component can be defined:

SINGLE1

Used by IMS for asynchronous output. One output message is sent for each SNA bracket. The message may or may not begin the bracket, but it always ends the bracket.

SINGLE2

Each message is sent with the SNA change-direction indicator (CD).

MULT1

All asynchronous messages for a given LTERM are sent before the bracket is ended. The end bracket (EB) occurs after the last message for the LTERM is acknowledged and dequeued.

MULT2

The same as MULT1, but CD is sent instead of EB.

SESSION=number

Specifies the number of parallel sessions for the link. Each session is represented by an IMS SUBPOOL macro and by a CICS DEFINE SESSIONS command.

EDIT=[{NO|YES}][,{NO|YES}]

Specifies whether user-supplied physical output and input edit routines are to be used.

The TYPE macro

The TYPE macro specifies the terminal type. Parameters of the TERMINAL macro can also be specified in the TYPE macro if they are common to all the terminals defined for this type.

UNITYPE=LUTYPE6

Must be specified for ISC.

The VTAMPOOL macro

The VTAMPOOL macro heads the list of SUBPOOL macros that define the individual sessions to the remote system.

Defining ISC over SNA connections

Before you can use ISC over SNA, you must define and install connections with attributes appropriate to your CICS and SNA environment.

Activating IP interconnectivity (IPIC) connections

IP interconnectivity requires CICS TCP/IP services to be activated.

System initialization parameters

To activate IPIC at CICS startup, specify TCP/IP=YES and ISC=YES as system initialization parameters. The default value of the **TCP/IP** and **ISC** parameters is NO.

Defining IPIC connections

Before you can use IPIC, you must:

- Define and install IPCONN with attributes appropriate to your CICS environment.
- Define and install a TCIPSERVICE definition with the PROTOCOL attribute set to IPIC.
- Review your MAXSOCKETS system initialization parameter settings. Ensure that you allocate enough sockets to support IPIC connections and other traffic that requires IP sockets.

Verifying Java components checklist

The CICS components required to support Java applications are included in the base product. You must also install the IBM 64-bit SDK for z/OS, Java Technology Edition, on z/OS UNIX, to provide Java support. Before you begin to set up and configure Java support in your CICS regions, verify that the Java components are correctly installed on your system.

IBM 64-bit SDK for z/OS, Java Technology Edition installation

Check that the IBM 64-bit SDK for z/OS, Java Technology Edition is installed in z/OS UNIX. CICS Transaction Server for z/OS, Version 5 Release 6 uses Version 8 of the IBM 64-bit SDK for z/OS, Java Technology Edition for Java support. CICS TS for z/OS, Version 5.6 supports only the 64-bit version of the SDK and not the 31-bit version.

The **JAVADIR** parameter of the DFHISTAR installation job for CICS specifies the location for the IBM SDK for z/OS. This parameter is used to customize the sample JVM profiles and Java security policy, so that they point to the SDK installation. The value in the JVM profiles determines the SDK that is used for Java support in a CICS region.

The default for the **JAVADIR** parameter is `java/J8.0_64`. This value is appended to `/pathprefix/usr/lpp/` to specify the default location of the IBM 64-bit SDK for z/OS, Java Technology Edition.

Files supplied by CICS

The CICS components for Java are installed during the installation process. Confirm that they are all present in the specified locations. z/OS UNIX System Services must be active in full function mode during the installation process to enable files to be stored in its file system.

In all of the directory paths shown here, `cicsts56` is a user-defined value, specified on the **USSDIR** parameter in the DFHISTAR installation job. This value determines the installation directory for CICS files on z/OS UNIX. This value is used as the `uss_path` variable in the DFHIJVMJ job, which creates the customized sample JVM profiles and Java security policy as z/OS UNIX files in the directories listed in the following table.

Table 25. Java components checklist		
Java component	Location	Comments
Directory containing JAR files supplied by CICS	z/OS UNIX directory: <code>/pathprefix/usr/lpp/cicsts/cicsts56/lib</code>	This directory is on the base library path and class path for all JVMs in CICS.
Directory containing the WebSphere Application Server Liberty	z/OS UNIX directory: <code>/pathprefix/usr/lpp/cicsts/cicsts56/wlp</code>	This directory contains the Liberty profile for running web servlets in a JVM server.

Table 25. Java components checklist (continued)		
Java component	Location	Comments
Sample programs	z/OS UNIX directory: <code>/pathprefix/usr/lpp/cicsts/cicsts56/samples</code>	The sample programs demonstrate the use of output redirection, the JCICS classes, JDBC, pipelines, and web services.
JVM profiles	z/OS UNIX directory: <code>/pathprefix/usr/lpp/cicsts/cicsts56/JVMProfiles</code>	JVM profiles specify options used in creating JVMs. These sample JVM profiles are customized for your system during the installation process. You can edit them using CICS Explorer or any text editor. The JVM profiles must always be available to CICS.

For all z/OS UNIX files, case is important. CICS does not automatically convert the name of these files to uppercase. When you use the name of a JVM profile anywhere in CICS, you must enter it using the same combination of uppercase and lowercase characters that is present in the z/OS UNIX file name. The file extension `.jvmprofile` must always be lowercase.

Defining DL/I support

CICS can provide DL/I database support by using the IBM IMS Database Manager.

For information about appropriate versions and releases, see [Overview of Database Control \(DBCTL\)](#). As they become available, versions of IMS newer than those versions listed are also compatible.

You can use DL/I support with CICS through these methods:

- Database control (DBCTL)
- CICS remote DL/I support, also known as *function shipping*

The IMS libraries referred to in the job streams are identified by `IMS.libnam`; for example, `IMS.PGMLIB`. If you use your own naming convention for IMS libraries, rename the IMS libraries accordingly.

CICS provides a CICS-DBCTL interface which enables DBCTL, IMS, or IMS DM/TM to satisfy DL/I requests that are issued from the CICS region. This method is simpler to install than local DL/I and provides additional function. Details of installing and using DBCTL are in [Installing and generating DBCTL](#).

CICS support for access to DL/I databases using the IBM Information Management System (IMS) product is included in the base product. No specific installation is required.

For more information about storage protection, see [Storage protection](#).

Program specification blocks (PDIR)

A directory of program specification blocks (PDIR) is a list of program specification blocks (PSBs) that define, for DL/I, the use of databases by application programs.

Your CICS region requires a PDIR to access a database owned by a remote CICS region. Your CICS region does not require a PDIR to access a DL/I database owned by DBCTL. For information about accessing DL/I databases owned by DBCTL, see the [Overview of Database Control \(DBCTL\)](#).

The modules that provide remote DL/I support are automatically loaded by CICS during startup when a DL/I PSB directory is specified with the `PDIR=` system initialization parameter.

Adding remote DL/I support

Remote DL/I support is included in CICS Transaction Server for z/OS, and works with one of the supported levels of IMS.

For information about appropriate IMS versions and releases, see [Overview of Database Control \(DBCTL\)](#).

Usually, you use remote DL/I support, with either MRO or ISC connections, to access databases owned by another CICS region. You can also use CICS remote DL/I support to access, through another CICS region connected to DBCTL, databases owned by DBCTL. CICS regions accessing databases owned by DBCTL (that is, connected to DBCTL) must be running on the same MVS image as the DBCTL system. An overview is given in [Figure 20 on page 193](#).

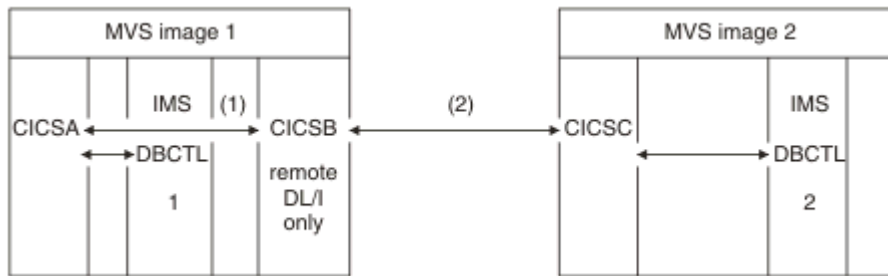


Figure 20. Using CICS remote DL/I support to access DBCTL databases

1. CICSB uses remote DL/I to access, through CICSA, databases owned by DBCTL 1 in MVS image 1. This support is only required if CICSB is not connected to DBCTL 1.
2. CICSB uses remote DL/I to access, through CICSC, databases owned by DBCTL 2 in MVS image 2.

For information about accessing DL/I databases owned by DBCTL, see the [Overview of Database Control \(DBCTL\)](#).

To add support in CICS for remote database access:

1. Code, assemble, and link-edit a program specification blocks directory (PDIR).
2. Code the PDIR CICS system initialization parameter for remote DL/I support.

Defining a PSB directory

You code entries in a program specification block directory (PDIR), to indicate the identity of the remote CICS region, or regions, to which you want CICS to function ship DL/I requests.

You do this by coding the SYSIDNT parameter in DFHDLPSB TYPE=ENTRY macros, which you assemble and link-edit to create a PDIR. You must also code the MXSSASZ parameter. You can, optionally, code the RMTNAME parameter to define the name by which the PSB is known in the remote CICS region. For information about creating PDIRs, see [Defining resources](#).

Coding CICS system initialization parameters for remote DL/I support

The following summary of the DL/I parameters specifies the parameters that you can, or must, code as CICS system initialization parameters:

- PDIR={YES|xx} Suffix of PSB directory (mandatory for remote DL/I)
- PSBCHK={NO|YES} Security check on remote terminals starting transactions
- XPSB={YES|name|NO} PSB entries to be checked by RACF

Enabling REXX for CICS

The REXX Development System for CICS and the REXX Runtime Facility for CICS are two licensed programs collectively referred to as REXX for CICS. With REXX for CICS, you can write and run REXX programs in a CICS region.

These programs have access to most EXEC CICS API commands, the CICS CEDA and CEMT transactions, and Db2 databases through the EXEC SQL interface.

See [Configuring REXX support](#).

Chapter 9. Renewing a Developer Trial license

The Developer Trial has an fixed expiry date to allow at least 90 days use from the date that it is ordered. A console message showing the expiry date is issued each day. After that time, your system terminates and, if you try to initialize CICS TS Developer Trial, the console message DFHT10201 is issued. If you want to use the CICS Transaction Server for z/OS Developer Trial, Version 5 Release 6 after the expiry date, you can follow this procedure to renew a Developer Trial license. When installed, the Developer Trial is the latest version, with a new expiry date.

Before you begin

You must have a Developer Trial that is installed with a license that is expired.

Procedure

1. Download the latest version of the Developer Trial from [Shopz](#).
This downloaded package will give you a number of FMIDs. If you have already installed the Developer Trial, you need only the activation module to renew the license.
2. Install the latest level of the enablement module using the supplied activation module installation jobs. Ensure that you use the REDO operand on the SMP/E APPLY and ACCEPT step.

Results

You have now renewed your Developer Trial license and the Developer Trial is now ready for use.

Chapter 10. Verifying the CICS installation

After you have installed CICS and applied any necessary service, you can use the DFHIVPBT and DFHIVPOL CICS-supplied installation verification procedures (IVPs) to confirm that CICS is operational.

You must have installed both the base component and activation module before you run the IVP jobs.

Preparing to run the IVPs

You must perform a number of tasks to prepare CICS for running the CICS installation verification procedures.

Creating and defining activities for the IVP jobs

In preparation for running the IVP jobs, create the CICS data sets, install the CICS SVC, define and activate the CICS applids, and define log streams.

Creating the CICS data sets for the IVP jobs

Before you can run any of the CICS-supplied IVP jobs, create the data sets that they use. For further information about creating the data sets for the IVP jobs, see [“Jobs for creating the CICS data sets” on page 170](#).

Installing the CICS SVC for the IVP jobs

All the IVP jobs require the CICS Type 3 SVC, which must be installed in the LPA.

If you have not already installed the CICS SVC in the LPA, described in [“Installing CICS modules in the MVS link pack area” on page 98](#), do so now before you attempt to run any of the IVP jobs. The IVP jobs do not use the Type 6 SVC (DFHHPSVC).

Defining and activating the CICS applids

If you want to use SNA with a CICS region started by any of the CICS IVP jobs, create and activate an SNA APPL definition for the CICS region application identifier (applid).

The applid defined to SNA must match the applid that is specified on the APPLID system initialization parameter that is used by the IVP job. For example, to be able to log on to the CICS region that is started by the DFHIVPOL job, you must perform one of the following actions:

- Create and activate an APPL definition for your own applid, which you specify on the APPLID parameter of the DFH\$SIP1 member of the SYSIN data set.
- Define and activate an APPL definition for the default applid DBDCCICS, which you specify on the APPLID parameter of the DFH\$SIP1 member of the SYSIN data set.

For more information about creating and activating SNA APPL definitions for CICS, see [“Defining CICS regions as applications to SNA” on page 89](#) and [“Data set naming conventions” on page 170](#).

Further, if you want to use SNA cross-domain services to communicate between CICS regions on separate MVS images, you must create and activate SNA CDRSC definitions in both MVS images involved in the communication. For more information about creating and activating SNA CDRSC definitions for CICS, see [“Defining cross-domain services when using SNA” on page 91](#).

Defining log streams

CICS automatically attempts to connect to its system log stream, unless you define a journal model resource definition to define the log stream as TYPE(DUMMY). You decide whether you want to run the IVPs with system logs or to run with dummy logging.

If you decide to run with real log streams, see [Defining the logger environment for CICS](#) for information about defining log streams.

Alternatively, you can define a CICS JOURNALMODEL resource definition with TYPE(DUMMY) to avoid having to define log streams. To run the IVPs with the minimum effort:

- Define JOURNALMODEL resource definitions in the CICS system definition file (CSD) for the primary and secondary system logs, DFHLOG and DFHSHUNT respectively, specifying TYPE(DUMMY); see [Figure 21 on page 198](#) for a sample job.
- Add the CSD group that contains your dummy system log journal models to your own group list, and include your group list on the GRPLIST system initialization parameter.

Note that your group list must *follow* the IBM-supplied list DFHLIST. DFHLIST includes group DFHLGMOD, which contains DFHLOG and DFHSHUNT JOURNALMODEL definitions. Concatenating your list after DFHLIST ensures that your DUMMY definitions replace the IBM definitions.

```
//CSDUP EXEC PGM=DFHCSDUP,REGION=1M,PARM='CSD(READWRITE)'  
//STEPLIB DD DSN=CICSTS56.SDFHLOAD,DISP=SHR  
//DFHCSD DD DSN=CICSTS56.DFHCSD,DISP=SHR  
//SYSPRINT DD SYSOUT=*  
//SYSABOUT DD SYSOUT=*  
//SYSABEND DD SYSOUT=*  
//SYSUDUMP DD SYSOUT=*  
//SYSIN DD *  
*  
* DEFINE JOURNAL MODELS FOR CICS LOG STREAMS AS DUMMY *  
DEFINE JOURNALMODEL(DFHLOG) GROUP(LOGTEST)  
DESCRIPTION(DEFINE SYSTEM LOG AS DUMMY)  
JOURNALNAME(DFHLOG) TYPE(DUMMY)  
ADD GROUP(LOGTEST) LIST(mylist)  
/*  
//
```

Figure 21. Sample job to define DUMMY JOURNALMODELS for CICS system logs

Reviewing and defining IVP security

You can run the IVP jobs with or without external security. To run the IVP jobs with external security, you must define to RACF an IVP default CICS user ID that has authority to run the transactions used as part of the IVP jobs.

Reviewing security requirements

As supplied, the system initialization parameters specify that external security is on. However, the IVP jobs have been set up with SEC=NO, indicating that external security is not on. The system initialization parameters also specify that the IVP jobs are subject to transaction security (XTRAN=YES), resource security (Xyyy=YES), and command security (XCMD=YES).

As supplied, the DFH\$SIP2 member of the SYSIN data set used by the DFHIVPBT job specifies SEC=NO, so that you can run this job without external security.

To run the IVP jobs with external security:

1. Define CICS resource profiles to RACF.
2. Define an IVP default CICS user ID to RACF.
3. Specify the IVP user ID on the DFLTUSER=userid system initialization parameter.

You must also give the IVP user ID sufficient authority to use transactions and resources that are required to run the IVP jobs:

1. Authorize the IVP user ID to run the transactions that are used as part of the IVP jobs. (See [Table 26 on page 199](#).) To do so, add the IVP user ID, with READ access, to the access list of the RACF profiles for the transaction member class (TCICSTRN) or the transaction group class (GCICSTRN).
2. If you define the transactions as prefixed resources, you must also specify the system initialization parameter SECPRFX={YES |[prefix](#)} for the IVP jobs.

3. Authorize the IVP user ID to access the resources that are used by the transactions. To do so, you add the IVP user ID, with appropriate authority, to the access list for the resource class profiles.
4. Authorize the IVP user ID to issue SP-type commands using the CEMT main terminal transaction. To do so, you add the IVP user ID, with appropriate authority, to the access list of the RACF profiles for the resource member class (CCICSCMD) or the resource group class (VCICSCMD). You must give the IVP user ID UPDATE access for the SHUTDOWN resource class; otherwise, the user ID cannot end the IVP jobs. Give the IVP user ID UPDATE access for the DUMPDS and SYSTEM resource classes, if the DFHIVPBT job is to be run with external security.

For information about implementing external security, see [CICS TS security](#). Alternatively, you can run the IVP jobs with limited security, for example:

- Without command security (XCMD=NO), the IVP user ID runs the IVP jobs without requiring authority to use the CEMT SP-type commands and the resources that they access.
- With transaction security only (Xyyy=NO including XCMD=NO), the IVP user ID runs the IVP jobs if authorized to use only the transactions used as part of the IVP jobs.

Authorizing the IVP user ID

To run the IVP jobs with external security, you must define to RACF an IVP default CICS user ID that has authority to run the transactions used as part of the IVP jobs.

These transactions include the CICS-supplied transactions that are listed in [Table 26 on page 199](#). The level of authority that is required by the IVP user ID depends on the security that you want to use for the IVP jobs. On a production system, the default user must not have access to any CICS-supplied transactions except those that you require in your CICS environment. Make the resource access authorizations that you give to the default user clearly limited to those resources that you intend to be universally available, and therefore not restricted in any way.

For information about the security requirements for CICS-supplied transactions, and about CICS security in general, see [Security for CICS-supplied transactions](#).

<i>Table 26. Transactions used as part of the IVP jobs</i>	
Application	Transactions
DFH\$BTCH	CWTO, CEMT, CEOT, CSFE
FILEA samples	
DFH\$MNU	AMNU, MENU, PMNU, DMNU
DFH\$ALL	AINQ, INQY, PINQ, DINQ AADD, ADDS, PADD, DADD AUPD, UPDT, PUPD, DUPD
DFH\$xBRW	ABRW, BRWS, PBRW, DBRW
DFH\$REN	AORD, OREN, PORD, DORD
DFH\$xCOM	AORQ, OREQ, PORQ, DORQ
DFH\$REP	AREP, REPT, PREP, DREP
Other functions	CETR, CEDA, CMAC, CMSG, CSGM

Specifying system initialization parameters for the IVP jobs

All the IVP jobs use the system initialization parameters that are specified in the associated DFH\$SIPn member of the SYSIN data set.

The DFH\$SIPn members, as supplied by CICS, use system initialization defaults, and the resources defined to CICS are adequate only for a basic CICS region. For example, in the case of the DFHIVPOL job, the resources defined limit the number of terminals you can use.

The DFH\$SIPn members of the SYSIN data set also contain some system initialization parameters to exclude CICS resources that are not required by the IVP jobs, or to include some not specified by the default system initialization parameters.

One such parameter is TCT=5\$, specifying the CICS sample LU control table, in the *hlq.SDFHLOAD* library. This LU control table defines the pair of sequential input and output devices, CARDIN and PRINTER. (These are the only devices that are defined in DFHTCT5\$.)

You can edit the DFH\$SIPn members of the SYSIN data set to make these changes:

- The default SVC number is 216. To use a different SVC number, specify CICSSVC=nnn as a parameter to the DFHSTART proc in the appropriate IVP job. For more information about defining CICS SVCs, see [“Installing the CICS SVCs” on page 84](#).

The IVP jobs do not require the Type 6 SVC.

- The applid used is CICSIVP1. To use a different applid, change the APPLID=CICSIVP1 system initialization parameter in the appropriate DFH\$SIPn member.
- The IVP jobs have external security switched off. To run with security (SEC=YES), define a suitable default user ID (for example, IVPUSER) with the required authority to run the IVP transactions. Add DFLTUSER=IVPUSER in the appropriate DFH\$SIPn member. For more information about defining the IVP user ID, see [“Reviewing and defining IVP security” on page 198](#).

You can define transactions as prefixed resources by using the IVP user ID, IVPUSER, or any other prefix, as the prefix (for example, IVPUSER.CEMT or prefix.CEMT). To do so, add SECPRFX=YES or SECPRFX=prefix in the appropriate DFH\$SIPn member for the IVP job.

The use of a prefix enables transactions to be run as part of the IVP jobs without affecting other CICS regions. For example, when the DFH\$BTCH batch stream is run, CICS sends authorization requests to RACF for the transactions and identifies them as IVPUSER.xxxx, where xxxx is the transaction ID (for example, CWTO or CEMT).

- CICS provides Language Environment support, for all the high-level language sample programs. For the Language Environment modules, CICS requires either predefined CSD definitions to be installed or program autoinstall to be active.

The IVP jobs include as comments the required DD statements for the Language Environment libraries.

- The IVP jobs run with auxiliary trace switched on (AUXTR=ON), and the auxiliary trace data set switching facility is set to switch once only (AUXTRSW=NEXT).
- TCT=NO is specified as a system initialization override, which causes CICS to use the dummy LU control table, DFHTCTDY. This dummy LU control table contains only the CICS and SNA control blocks for use with SNA LUs but no LU entries.
-

Resources for the CICS messages facility, CMAC

You can use the CICS messages facility, the CICS-supplied transaction CMAC, to provide the messages and codes descriptions online. Before you can use CMAC, you must create and initialize the CICS messages data set DFHCMACD, define the resources required by the facility, and make them available to your CICS region.

For information about creating and initializing the DFHCMACD data set, see [“DFHCMACI job for creating the messages data set” on page 173](#).

The file DFHCMACD, managed by CICS file control, accesses the DFHCMACD data set. You must create a definition for this file in the CICS system definition file (CSD). The CICS-supplied definition for the DFHCMACD file and other resources that are required by the CICS messages facility are in the CSD group DFHMAC. The CICS startup procedure (in the IVP jobs) has a DD statement for the CMAC file, but for dynamic allocation copy the supplied resource definition for the DFHCMACD file and add the DSNNAME option.

Specify the DFHMAC group of resources for the CICS messages facility only in those CICS regions that use the facility; for example, on some terminal-owning regions, but perhaps not on data-owning regions.

The CICS startup procedure, DFHSTART

All the IVP jobs include a procedure to start CICS. You can use this procedure as a basis for your own CICS startup procedures.

The DFHSTART procedure comprises the following steps:

1. CICSCNTL: determine whether CICS is to be started.
2. DTCNTL: determine whether dump and trace analysis is to be performed.
3. CICS: run CICS.
4. PRTDMPA: print any contents of the CICS DFHDMPA dump data set.
5. PRTDMPB: print any contents of the CICS DFHDMPB dump data set.
6. PRTAUXT: print any contents of the auxiliary trace DFHAUXT data set.
7. PRTBUXT: print any contents of the auxiliary trace DFHBUXT data set.

When you run the DFHISTAR job, it overrides the default values in the IVP jobs with the values you specified in the DFHISTAR job.

Parameters

The following symbolic parameters are defined in the IVP jobs:

ACTIVE(SDFHLIC)

Is the type of license module used:

- SDFHLIC - Normal Licence
- SDFHVUE - Value Unit Edition
- SDFHDEV - Developer Trial

INDEX1(hlq)

Is the high-level index of the CICS runtime data sets, as specified on the DSINFO parameter of the DFHISTAR job.

INDEX2(hlq)

Is the high-level index of the CICS load libraries, as specified on the INDEX parameter of the DFHISTAR job.

INDEX3(hlq)

Is the high-level index of the CICSplex SM load libraries, as specified on the INDEX parameter of the DFHISTAR job.

INDEX4(hlq)

Is the high-level index of the activation module load library, as specified on the INDEX parameter of the DFHISTAR job.

REGNAM(TR)

Is the REGION name for a single or MRO region.

REG(80M)

defines the MVS region size for the step to run CICS.

MEMLIM(10G)

The z/OS MEMLIMIT parameter limits the amount of 64-bit (above-the-bar) storage that the CICS address space can use.

START(AUTO)

Is the type of CICS startup to be performed.

DUMPTR(YES)

Specifies whether dump and trace analysis is required. The steps PRTDMPA, PRTDMPB, DFHAUXT, and DFHBUXT are run only if you specify DUMPTR=YES.

RUNCICS(YES)

Specifies whether CICS is to be started. The step to run CICS runs only if you code RUNCICS=YES (the default). To perform dump and trace analysis without starting CICS, code RUNCICS=NO.

OUTC(*)

Is the output print class.

SIP(T)

Is the suffix of the DFH\$SIP member (in the SYSIN data set) to be used during CICS startup.

CICSSVC(216)

Specifies the CICSSVC number that you have assigned to the CICS type 3 SVC.

USSHOME(/usr/lpp/cicsts/cicsts56)

Specifies the name and path of the root directory for CICS Transaction Server files on z/OS UNIX.

DD statements for CICS data sets

The startup job step, DFHSTART, contains DD statements for the CICS data sets.

The IVP jobs include as comments the required DD statements for the Language Environment libraries.

Table 27. DD statements for CICS data sets in the DFHSTART procedure	
DD name	Description
SYSIN	SYSIN data set, containing the DFH\$SIPn members that specify system initialization parameter overrides.
DFHCMACD	Messages data set, required for the CICS messages transaction, CMAC.
FILEA	Sample VSAM data set, required by the FILEA sample applications.
DFHTEMP	Auxiliary temporary storage data set, required by the FILEA sample applications.
DFHINTRA	Transient data intrapartition data set, required by the FILEA sample applications.
DFHAUXT DFHBUXT	First auxiliary trace (A) data set. Second auxiliary trace (B) data set. The auxiliary trace data sets, DFHAUXT and DFHBUXT, are required because the IVP jobs run with auxiliary trace switched on, and the auxiliary trace data set switching facility is set to switch once only.
DFHLCD	(Mandatory) CICS local catalog data set (VSAM), used by the CICS domains to save some of their information between CICS runs, and to preserve this information across a cold start.
DFHGCD	(Mandatory) CICS global catalog data set (VSAM) has a variety of uses, including: during a CICS run, holding resource definitions that are installed; and, during a controlled shutdown, recording part of the warm keypoint information.

Table 27. DD statements for CICS data sets in the DFHSTART procedure (continued)

DD name	Description
DFHCXRF	Transient data extrapartition data set, used by CICS as the target for messages sent to any transient data destination before CICS has completed intrapartition transient data initialization. Use of this DDname is optional, but if it is not used, any messages that are written here are lost.
DFHLRQ	The local request queue data set is used to store pending BTS requests; for example, timer requests or requests to run activities. It is recoverable and used to ensure that, if CICS fails, no pending requests are lost. For more information, see Configuring for BTS .
LOGUSR	Data set for the extrapartition transient data destination, LOGA, used by the CICS sample programs.
MSGUSR	Data set for the extrapartition transient data destination, CSSL, used by a number of CICS services.
COUT	Data set for the extrapartition transient data destinations used by C/370 application programs. This data set is the destination for the C/370 output data streams, stdout (CCSO) and, indirectly, stderr (CCSE).
DFHDMPA DFHDMPB	First transaction dump (A) data set. Second transaction dump (B) data set. The dump data sets are included because CICS always tries to open a transaction dump data set, and issues a warning message if it is unable to do so for any reason.
DFHCSD	(Mandatory) CICS system definition data set (VSAM).

Verify batch job, DFHIVPBT

The CICS-supplied verify batch job, DFHIVPBT, is tailored to your CICS environment and stored in the *hlq.XDFHINST* library when you run the DFHISTAR job.

For more information about installing CICS using DFHISTAR, see [Installing CICS TS using DFHISTAR](#).

DFHIVPBT starts up CICS, specifying a pair of sequential input and output devices (CARDIN and PRINTER) to be used instead of an ordinary terminal. It then runs a number of CICS transactions that are read from CARDIN. The last transaction in the input stream shuts down CICS. If you want to verify support for the COBOL, C, C++, and PL/I languages, remove the comment marks from the SCEERUN and SCEERUN2 libraries, and increase the memory size for the job.

This IVP comprises the following job steps:

1. Job step GENINPT unloads the member DFH\$BTCH from the *hlq.SDFHSAMP* library into the CARDIN data set, using the MVS utility program, IEBGENER.
2. Job step DFHSTART invokes the CICS initialization program, DFHSIP, to start CICS. The DFHSIP program reads startup system initialization parameters from the DFH\$SIP2 member of the SYSIN data set. For information about the system initialization parameters used by the IVP jobs, see [“Specifying system initialization parameters for the IVP jobs”](#) on page 200.

The DFH\$BTCH data set, as described in [Figure 22 on page 204](#), is used as terminal input.

```

CWTO START OF BATCH STREAM DFH$BTCH\
CEMT S TIME(120)\
CEMT S DUMPDS SWITCH\
CEOT\
CSFE\
PRINT\
THIS MESSAGE HAS BEEN RECEIVED FROM THE
END\
CSXX\
CWTO END OF BATCH STREAM DFH$BTCH - SHUTTING DOWN CICS\
CEMT P SHUT\
EXIT TIME INTVL TO 120 MILLISEC
SWITCH FROM DUMP A TO B
INQUIRE TERMINAL STATUS
F. E. TERMINAL TEST REQUEST
TO SEND ALL CHARACTERS
TO END THE TEST
INVALID TRANSACTION IDENT.
NORMAL TERMINATION OF CICS

```

where \ is the End Of Data Input character X'E0'.

Figure 22. DFH\$BTCH data set, input to the DFHIVPBT job

Check the job log for the DFHIVPBT job to verify that CICS startup and shutdown completed successfully. Verify that the following messages are issued:

```

DFHSI1517 CICSIVP1 Control is being given to CICS.
DFHKE1799 CICSIVP1 TERMINATION OF CICS IS COMPLETE.

```

The output from the DFHIVPBT job includes CICS messages written to one of the extrapartition destinations, responses to the transactions in the DFH\$BTCH data set, and an auxiliary trace.

When CICS is initialized by DFHIVPBT, the log streams do not exist. CICS issues a request to create a log stream dynamically using MVS define log stream services. If system log initialization fails, CICS abends.

Verify interactive job, DFHIVPOL

The verify interactive job, DFHIVPOL, is tailored to your CICS environment and stored in the *hlq.XDFHINST* library when you run the DFHISTAR job.

You can use the DFHIVPOL job to start a CICS region to try out some functions, for example:

- Use the main LU transaction, CEMT. You can also use CEMT from the MVS system console. For information about CEMT, see [CEMT - main terminal](#).
- Use the resource definition online transaction, CEDA. For information about using CEDA, see [Resource management transaction CEDA commands](#).
- Use the sample application transaction AMNU, to access the sample VSAM file, FILEA.

You require either an IBM 3270 Information Display System LU or a console device, or both.

If you use an IBM 3270 Information Display System LU with this IVP, you can try CEDA, CEMT, and the sample applications.

From a console device, the CEDA transaction can be used only to INSTALL resource definitions. The sample programs cannot be run from a console device.

Defining an SNA LU for the online IVP

You can define an SNA LU by using one of these methods:

- Use the autoinstall facility, to avoid defining LUs to CICS explicitly before they can be used.
See [“Autoinstalling for an SNA LU” on page 204](#).
- Define a LU explicitly in the CICS system definition file (CSD), using the DEFINE command of the CSD update batch utility program DFHCSDUP.

See [“Defining an SNA LU in the CSD” on page 205](#).

Autoinstalling for an SNA LU

With autoinstall, the resource definitions you create using RDO can act as models or templates for many resources of the same type. You then leave CICS to match real resources with one of the models. CICS installs table entries for these real resources dynamically, as and when they are required.

When using autoinstall, be aware that when CICS processes an autoinstall request, it uses data from the SNA logmode table. Be aware of this important consideration. An autoinstall request succeeds only when the logmode data, which is passed to CICS in the BIND image, matches one of the model terminal definitions recorded in the autoinstall model table (AMT) from the CSD. Before attempting to start CICS and autoinstall a LU for this IVP, check your SNA definitions with those given in [Coding entries in the z/OS Communications Server LOGON mode table](#). If CICS fails to match model and logmode data, you receive message DFHZC6987I.

The CSD is defined and initialized for all the IVP jobs when you run the DFHCOMDS job and includes some IBM-supplied definitions for use with autoinstall. These definitions are defined in the following groups:

DFHTERM

Model LU definitions for use with autoinstall. For example, two of the TERMINAL definitions are 3270 and LU2.

DFHTYPE

Partial LU definitions (TYPETERMs) defining common LU properties, or attributes. For example, two of the TYPETERM definitions are DFH3270 (to define a non-SNA 3270 LU) and DFHLU2E2 (to define an SNA 3270 model 2 LU). The DFHLU2E2 resource definition matches the SNA-supplied logmode SNX32702.

The DFHTERM and DFHTYPE groups are included in the CICS-defined group list called DFHLIST, which is defined in the GRPLIST operand. If the CICS-supplied definitions are not suitable for your installation, you can create additional TYPETERM and model TERMINAL definitions in the CSD, but without a LU you must do this offline, using the DFHCSDUP utility program. For information about autoinstall definitions, see [Autoinstall](#).

Autoinstall also requires a user program to assign LU identifiers, and, if necessary, to control access to the system. When you run the online IVP, you are unlikely to have any special requirements for LU identifiers, or to control access, in which case you can use the IBM-supplied autoinstall user program, DFHZATDX. If you are using autoinstall for APPC connections and LUs, the sample autoinstall user program is called DFHZATDY.

Defining an SNA LU in the CSD

If you want to use an explicitly defined LU, rather than allow CICS to autoinstall a LU, define it offline using the DFHCSDUP utility program.

The normal way to create resource definitions in the CSD is to use the CEDA DEFINE command from a CICS main LU, but without a LU you can only do this using the DFHCSDUP utility program. For an example of a DFHCSDUP job to define an SNA LU in the CSD, see [Figure 23 on page 205](#).

```
//DEFTERM JOB (accounting information),MSGCLASS=A,
//          MSGLEVEL=(1,1),CLASS=A,NOTIFY=userid
//VTAMDEF EXEC PGM=DFHCSDUP
//STEPLIB DD DSN=CICSTS56.CICS.SDFHLOAD,DISP=SHR
//DFHCSD DD DSN=CICSTS56.CICS.DFHCSD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//SYSIN DD *
*
DEFINE TERMINAL(trmidnt) NETNAME(vtamname) GROUP(grpname)
        TYPETERM(name) INSERVICE(NO) AUTINSTMODEL(NO)
*
APPEND LIST(DFHLIST) TO(yourlist)

*
ADD GROUP(grpname) LIST(yourlist)
*
LIST LIST(yourlist) OBJECTS
/*
//
```

Figure 23. Defining a LU by using the DFHCSDUP utility program

GROUP(name)

Code a unique name for the group to which the LU resource definition is to belong.

NETNAME(name)

Code the 8-character SNA name that identifies this LU to your SNA system.

TERMINAL(name)

Code a unique 4-character LU identifier as the name by which CICS is to know the LU.

TO(yourlist) and LIST(yourlist)

Code a unique name for *yourlist*. If your new group list does not include all the CICS-supplied resources as well as your own, you must specify DFHLIST and *yourlist* on the GRPLIST system initialization parameter of your CICS startup job.

TYPETERM(name)

Specify a unique name to identify the resource definition that matches the properties of the type of LU you are using. For example, to define an SNA 3270 model 2 LU, specify the CICS-supplied TYPETERM definition DFHLU2E2.

To include the CICS-supplied list of resources in a new group list, create a new list by copying the CICS-supplied list, DFHLIST, using the APPEND command. The CICS-supplied group list, DFHLIST, is a protected group that you cannot modify. You can then add your resource definition groups to the new list. Before you run the IVP, define your new group list to CICS, by adding a system initialization override to the SYSIN data set in the DFHIVPOL job stream.

Defining the CICS APPLID to SNA

Ensure that either SNA knows the CICS application identifier (APPLID) or you change the CICS APPLID to one that is already known to your SNA system.

If you use the default APPLID (DBDCCICS), define this to SNA as described in [“Defining specific APPL definitions and APPL parameters to SNA”](#) on page 90, before starting the DFHIVPOL job.

Communicating with CICS from an MVS console or a TSO session

If you want to communicate with CICS from an MVS console, you must define a console in the CSD before starting the IVP. You cannot define a console in the TCT. For more information about defining consoles, see [Defining console devices to CICS](#).

If you want to communicate with CICS from a TSO session, you must define the TSO user as a console device in the CSD before starting the IVP. For more information, see [Defining console devices to CICS](#).

Running the DFHIVPOL job

The DFHIVPOL job includes a procedure, DFHSTART, that calls the CICS initialization program, DFHSIP, to start CICS.

When you have successfully logged on to CICS, you can carry out any of the interactive operations described in [“Verifying SNA LU operations”](#) on page 209.

While logged on to CICS, perform a CEMT SET DUMPDS SWITCH to ensure that both dump data sets are initialized, before the dump utility program is run when you shut down CICS.

Finally, you can shut down CICS.

Sample job log for the DFHIVPOL job

When you run the DFHIVPOL job, your job log looks similar to the sample log shown.

```

1 JES2 JOB LOG -- SYSTEM MV26 -- NODE WINMVS2C
0
17.17.29 JOB35727 ---- TUESDAY, 24 APRIL 2007 ----
17.17.29 JOB35727 ICH700011 CICINST LAST ACCESS AT 16:24:15 ON TUESDAY, APRIL 24, 2007
17.17.29 JOB35727 $HASP373 DFHIVPOL STARTED - INIT 69 - CLASS A - SYS MV26
17.17.29 JOB35727 IEF403I DFHIVPOL - STARTED - TIME=17.17.29
17.17.29 JOB35727 -
17.17.29 JOB35727 -TIMINGS (MINS.)--
17.17.29 JOB35727 -JOBNAME STEPNAM PROCSTEP RC EXCP CPU SRB CLOCK SERV PG PAGE SWAP VIO SWAPS
STEPNO
17.17.29 JOB35727 -DFHIVPOL CICS CICSCNTL 01 23 .00 .00 .00 167 0 0 0 0
0
17.17.29 JOB35727 -DFHIVPOL CICS DTCNTL 01 19 .00 .00 .00 147 0 0 0 0
0
17.17.30 JOB35727 DFHPA1101 CICSIVP1 DFHSIT IS BEING LOADED.
17.17.30 JOB35727 DFHPA1108 CICSIVP1 DFHSIT HAS BEEN LOADED. (GENERATED AT: MM/DD= 01/12 HH:MM= 13:57).
17.17.30 JOB35727 DFHPA1100 CICSIVP1 OVERRIDE PARAMETERS FROM JCL EXEC STATEMENT: START=AUTO,SYSLN
17.17.30 JOB35727 DFHPA1102 CICSIVP1 OVERRIDE PARAMETERS FROM SYSIN: 1
17.17.30 JOB35727 DFHPA1927 CICSIVP1 XRF=NO,
17.17.30 JOB35727 DFHPA1927 CICSIVP1 AUXTR=ON,
17.17.30 JOB35727 DFHPA1927 CICSIVP1 AUXTRSW=NEXT,
17.17.30 JOB35727 DFHPA1927 CICSIVP1 APPLID=CICSIVP1, 2
17.17.30 JOB35727 DFHPA1927 CICSIVP1 FCT=NO,
17.17.30 JOB35727 DFHPA1927 CICSIVP1 TCT=NO,
17.17.30 JOB35727 DFHPA1927 CICSIVP1 SRT=NO,
17.17.30 JOB35727 DFHPA1927 CICSIVP1 SEC=NO,
17.17.30 JOB35727 DFHPA1927 CICSIVP1 TRTABSZ=64,
17.17.30 JOB35727 DFHPA1927 CICSIVP1 PGRET=P/,
17.17.30 JOB35727 DFHPA1927 CICSIVP1 PGPURGE=T/,
17.17.30 JOB35727 DFHPA1927 CICSIVP1 PGCOPY=C/,
17.17.30 JOB35727 DFHPA1927 CICSIVP1 PGCHAIN=X/,
17.17.30 JOB35727 DFHPA1927 CICSIVP1 CICS SVC=233,
17.17.30 JOB35727 DFHPA1927 CICSIVP1 .END
17.17.30 JOB35727 DFHPA1103 CICSIVP1 END OF FILE ON SYSIN.
17.17.31 JOB35727 +DFHTR0103 TRACE TABLE SIZE IS 64KB
17.17.31 JOB35727 +DFHSM0122I CICSIVP1 Limit of DSA storage below 16MB is 5 120KB. 3
17.17.31 JOB35727 +DFHSM0123I CICSIVP1 Limit of DSA storage above 16MB is 48MB.
17.17.31 JOB35727 +DFHSM0113I CICSIVP1 Storage protection is not active.
17.17.31 JOB35727 +DFHSM0126I CICSIVP1 Transaction isolation is not active.
17.17.32 JOB35727 +DFHDM0101I CICSIVP1 CICS is initializing.
17.17.32 JOB35727 +DFHWM0109I CICSIVP1 Web domain initialization has started.
17.17.32 JOB35727 +DFHSM0100I CICSIVP1 Sockets domain initialization has started.
17.17.32 JOB35727 +DFHRX0100I CICSIVP1 RX domain initialization has started.
17.17.32 JOB35727 +DFHRX0101I CICSIVP1 RX domain initialization has ended.
17.17.33 JOB35727 +DFHLG0101I CICSIVP1 Log manager domain initialization has started.
17.17.33 JOB35727 +DFHEJ0101 CICSIVP1 296
296 Enterprise Java domain initialization has started. Java is a
296 trademark of Oracle.
17.17.33 JOB35727 +DFHDM0100I CICSIVP1 Document domain initialization has started.
17.17.33 JOB35727 +DFHXS1100I CICSIVP1 Security initialization has started.
17.17.33 JOB35727 +DFHSI1500 CICSIVP1 CICS startup is in progress for CICS Transaction Server
17.17.33 JOB35727 +DFHDM0304I CICSIVP1 Transaction Dump Data set DFHDMPA opened.
17.17.33 JOB35727 +DFHXS1102I CICSIVP1 Security is inactive.
17.17.33 JOB35727 +DFHSI1501I CICSIVP1 Loading CICS nucleus.
17.17.34 JOB35727 +DFHTR0113 CICSIVP1 Auxiliary trace is being started on data set DFHAUT.
17.17.34 JOB35727 +DFHCQ0100I CICSIVP1 Console queue initialization has started.
17.17.34 JOB35727 +DFHCQ0101I CICSIVP1 Console queue initialization has ended.
17.17.34 JOB35727 +DFHCQ0103I CICSIVP1 MVS console queue is open.
17.17.34 JOB35727 +DFHCQ0200I CICSIVP1 CEKL transaction enabled.
17.17.34 JOB35727 +DFHXS1101I CICSIVP1 Security initialization has ended.
17.17.34 JOB35727 +DFHRM0141 CICSIVP1 Recovery manager autostart override record is not present.
Normal processing
continues.
17.17.34 JOB35727 +DFHDM0101I CICSIVP1 Document domain initialization has ended.
17.17.34 JOB35727 +DFHMM0105I CICSIVP1 Using default Monitoring Control Table.
17.17.34 JOB35727 +DFHMM0110I CICSIVP1 CICS Monitoring is inactive.
17.17.34 JOB35727 +DFHSM0101I CICSIVP1 Sockets domain initialization has ended.

```

```

17.17.35 JOB35727 IEC031I D37-04,IFG0554P,DFHIVPOL,CICS,DFHAUXT,D306,P2P0C6,INST.CICSTS56.CICS.DFHAUXT
17.17.35 JOB35727 +DFHTR0110 - AUXILIARY TRACE DATA SET DFHAUXT FULL - SWITCHING TO DFHBUXT
17.17.35 JOB35727 +DFHWP0110I CICSIVP1 Web domain initialization has ended.
17.17.35 JOB35727 IEC031I D37-04,IFG0554P,DFHIVPOL,CICS,DFHBUXT,D50B,P2P14B,INST.CICSTS56.CICS.DFHBUXT
17.17.35 JOB35727 +DFHTR0109 - AUXILIARY TRACE DATA SET DFHBUXT FULL - AUXILIARY TRACE HAS BEEN STOPPED
17.17.35 JOB35727 +DFHSI1502I CICSIVP1 CICS startup is Warm.
17.17.35 JOB35727 +DFHTS0100I CICSIVP1 Temporary Storage initialization has started.
17.17.35 JOB35727 +DFHLG0103I CICSIVP1 System log (DFHLOG) initialization has started.
17.17.35 JOB35727 +DFHLG0104I CICSIVP1 322
322 System log (DFHLOG) initialization has ended. Log stream
322 CICINST.CICSIVP1.DFHLOG is connected to structure LOG_GENERAL_008.
17.17.35 JOB35727 +DFHLG0103I CICSIVP1 System log (DFHSHUNT) initialization has started.
17.17.35 JOB35727 +DFHSI1503I CICSIVP1 Terminal data sets are being opened.
17.17.35 JOB35727 +DFHLG0104I CICSIVP1 327
327 System log (DFHSHUNT) initialization has ended. Log stream
327 CICINST.CICSIVP1.DFHSHUNT is connected to structure LOG_GENERAL_008.
17.17.36 JOB35727 +DFHLG0102I CICSIVP1 Log manager domain initialization has ended.
17.17.36 JOB35727 +DFHKE0406I CICSIVP1 329
329 CICS is about to wait for predecessors defined in the MVS automatic
329 restart management policy for this region.
17.17.36 JOB35727 +DFHKE0412I CICSIVP1 CICS WAITPRED call to automatic restart manager has completed.
17.17.36 JOB35727 +DFHCP0101I CICSIVP1 CPI initialization has started.
17.17.36 JOB35727 +DFHPR0104I CICSIVP1 Partner resource manager initialization has started.
17.17.36 JOB35727 +DFHAI0101I CICSIVP1 AITM initialization has started.
17.17.36 JOB35727 +DFHFC0100I CICSIVP1 File Control initialization has started.
17.17.36 JOB35727 +DFHTD0100I CICSIVP1 Transient Data initialization has started.
17.17.39 JOB35727 +DFHTS0101I CICSIVP1 Temporary Storage initialization has ended.
17.17.39 JOB35727 +DFHER5730 CICSIVP1 User recovery beginning
17.17.39 JOB35727 +DFHLG0745I CICSIVP1 System log full scan has started.
17.17.39 JOB35727 +DFHLG0748I CICSIVP1 System log selective scan has started.
17.17.39 JOB35727 +DFHLG0749I CICSIVP1 System log scan has completed.
17.17.40 JOB35727 +DFHER5731 CICSIVP1 No active user records on the system log
17.17.40 JOB35727 +DFHER5732 CICSIVP1 User recovery completed
17.17.40 JOB35727 +DFHTD0101I CICSIVP1 Transient Data initialization has ended.
17.17.40 JOB35727 +DFHFC0101I CICSIVP1 File Control initialization has ended.
17.17.40 JOB35727 +DFHTC1575 CICSIVP1 No TCT entry for SAMA
17.17.40 JOB35727 +DFHCP0102I CICSIVP1 CPI initialization has ended.
17.17.40 JOB35727 +DFHPR0105I CICSIVP1 Partner resource manager initialization has ended.
17.17.40 JOB35727 +DFHAI0102I CICSIVP1 AITM initialization has ended.
17.17.40 JOB35727 +DFHAP1203I CICSIVP1 Language Environment is being initialized.
17.17.40 JOB35727 +DFHAP1200 CICSIVP1 A CICS request to the Language Environment has failed. Reason code '0011020'.
17.17.40 JOB35727 +DFHAP1208 CICSIVP1 Language Environment cannot support the COBOL language. 5
17.17.40 JOB35727 +DFHAP1209 CICSIVP1 Language Environment cannot support the C/C++ languages. 5
17.17.40 JOB35727 +DFHAP1210 CICSIVP1 Language Environment cannot support the PL/I language. 5
17.17.40 JOB35727 +DFHAP1211I CICSIVP1 Language Environment initialization completed.
17.17.40 JOB35727 +DFHWP1007 CICSIVP1 Initializing CICS Web environment.
17.17.40 JOB35727 +DFHWP1008 CICSIVP1 CICS Web environment initialization is complete.
17.17.40 JOB35727 +DFHSI1517 CICSIVP1 Control is being given to CICS.
17.17.40 JOB35727 +DFHEJ0102 CICSIVP1 Enterprise Java domain initialization has ended.
17.23.42 JOB35727 +DFHFC0208I CICSIVP1 069
069 LSR pool 1 is being built dynamically by CICS because all of the
069 necessary parameters have not been supplied. Either there is no
069 LSRPOOL definition or it is incomplete. The following are not
069 defined: 'CI SIZE' 'STRINGS' 'MAXKEYLENGTH'. A delay is possible.
17.24.17 JOB35727 +DFHFC0208I CICSIVP1 137
137 LSR pool 1 is being built dynamically by CICS because all of the
137 necessary parameters have not been supplied. Either there is no
137 LSRPOOL definition or it is incomplete. The following are not
137 defined: 'CI SIZE' 'STRINGS' 'MAXKEYLENGTH'. A delay is possible.
17.24.28 JOB35727 +DFHTM1715 CICSIVP1 CICS is being quiesced by userid CICSUSER in transaction CEMT
at netname
IYCQTC70. 4
17.24.28 JOB35727 +DFHDM0102I CICSIVP1 CICS is quiescing.
17.24.28 JOB35727 +DFHTM1782I CICSIVP1 All non-system tasks have been successfully terminated.
17.24.28 JOB35727 +DFHXC2305I CICSIVP1 Termination of VTAM sessions beginning
17.24.28 JOB35727 +DFHCESD CICSIVP1 SHUTDOWN ASSIST TRANSACTION CESD STARTING. SHUTDOWN IS NORMAL.
17.24.29 JOB35727 +DFHXC2316 CICSIVP1 VTAM ACB is closed
17.24.29 JOB35727 +DFHCQ0104I CICSIVP1 MVS console queue is closed.
17.24.31 JOB35727 +DFHRM0204 CICSIVP1 There are no indoubt, commit-failed or backout-failed UOWs.
17.24.32 JOB35727 +DFHRM0130 CICSIVP1 Recovery manager has successfully quiesced.
17.24.32 JOB35727 +DFHUU0303I CICSIVP1 Transaction Dump Data set DFHDMPA closed.
17.24.32 JOB35727 +DFHKE1799 CICSIVP1 TERMINATION OF CICS IS COMPLETE.

```



```

17.24.33 JOB35727 -DFHIVPOL CICS CICS 00 5757 .02 .00 7.05 21599 0 0 0 0
0 3
17.24.33 JOB35727 -DFHIVPOL CICS PRTDMPA 00 136 .00 .00 .00 286 0 0 0 0
0 4
17.24.34 JOB35727 -DFHIVPOL CICS PRTDMPB 00 135 .00 .00 .00 285 0 0 0 0
0 5
17.24.37 JOB35727 -DFHIVPOL CICS PRTAUXT 00 1559 .01 .00 .05 13828 0 0 0 0
0 6
17.24.37 JOB35727 $HASP375 DFHIVPOL ESTIMATE LINES EXCEEDED
17.24.38 JOB35727 $HASP375 DFHIVPOL ESTIMATE EXCEEDED BY 5,000 LINES
17.24.38 JOB35727 $HASP375 DFHIVPOL ESTIMATE EXCEEDED BY 10,000 LINES
17.24.38 JOB35727 $HASP375 DFHIVPOL ESTIMATE EXCEEDED BY 15,000 LINES
17.24.38 JOB35727 $HASP375 DFHIVPOL ESTIMATE EXCEEDED BY 20,000 LINES
17.24.39 JOB35727 $HASP375 DFHIVPOL ESTIMATE EXCEEDED BY 25,000 LINES
17.24.39 JOB35727 $HASP375 DFHIVPOL ESTIMATE EXCEEDED BY 30,000 LINES
17.24.39 JOB35727 $HASP375 DFHIVPOL ESTIMATE EXCEEDED BY 35,000 LINES
17.24.40 JOB35727 -DFHIVPOL CICS PRTBUXT 00 1572 .01 .00 .05 13923 0 0 0 0
0 7
17.24.40 JOB35727 IEF404I DFHIVPOL - ENDED - TIME=17.24.40
17.24.40 JOB35727 -DFHIVPOL ENDED. NAME-CICINST TOTAL CPU TIME= .06 TOTAL ELAPSED TIME= 7.18
17.24.40 JOB35727 $HASP395 DFHIVPOL ENDED

```

Figure 24. Sample job log for the DFHIVPOL job

Note: VTAM is now z/OS Communications Server.

1. For information about the system initialization parameters that the IVP jobs use, see [“Specifying system initialization parameters for the IVP jobs”](#) on page 200. See also [2](#) and [3](#).
2. For more information about defining an APPLID for the CICS IVP jobs, see [“Defining CICS regions as applications to SNA”](#) on page 89. An APPLID of CICSIVP1 is used in [Figure 24](#) on page 209.
3. The DFHSM0122 and DFHM0123 messages inform you of the limits available for the dynamic storage areas below 16 MB, and above 16 MB but below 2 GB. For information about these storage areas, see [CICS dynamic storage areas](#). Storage for the extended read-only DSA, ERDSA, is obtained from read-only key 0 protected storage, because the default system initialization specifies RENTPGM=PROTECT.
4. The DFHTM1715 message is issued because the CICS region was shut down by the terminal user, with netname IYCWTC30, issuing a CEMT PERFORM SHUTDOWN command.
5. If you want COBOL, C, C++, and PL/I languages, remove the comment marks from the SCEERUN and SCEERUN2 libraries, and increase the memory size for the job.
6. The DFHME0107 message Module CJEMCT1E cannot be found in the library, is displayed only when a CICS region is started without any CICSplex System Manager libraries or infrastructure in place. This is informational and does not restrict any CICS functionality.

Verifying SNA LU operations

You can perform a number of activities to verify that CICS is working properly, for example, logging on, using transactions, running sample programs, and shutting CICS down.

Logging on to a SNA LU

When the DFHIVPOL job displays the console message CONTROL IS BEING GIVEN TO CICS, you can log on to CICS using an IBM 3270 Information Display system terminal.

Use the CICS application identifier that you specified when you started CICS to log on through your SNA LU. For example, unless you changed the APPLID as specified as at system initialization (CICSIVP1), enter LOGON APPLID(CICSIVP1).

If you are using autoinstall, your logon request is passed to CICS and, if all the autoinstall requirements described in Autoinstall have been met, CICS installs your LU. It does so by creating a TCT terminal entry (TCTTE) using the model definitions defined in the group list, DFHLIST, and the LU identifier returned by the autoinstall user program (DFHZATDX in this case).

If you are using a LU defined in the CSD explicitly, and included in the group list specified in the startup job stream, CICS identifies the installed resource definitions by the SNA net name and creates the required TCTTE.


When you log on to CICS, your LU can display a good morning message as specified on the GMTRAN system initialization parameter. The default transaction, CSGM, displays a welcome message as defined by the GMTEXT system initialization parameter.

Using CICS provided transactions through a terminal

After you have started CICS with the DFHIVPOL job, you can use the CICS provided transactions to try out various functions of CICS to help you verify that CICS is working properly. You can use the transactions at a CICS terminal, for example, CEMT and, if you defined one, the system console.

For information about the CICS transactions that you can try with the DFHIVPOL job, and about the message-switching responses to those transactions, see [CICS supplied transactions descriptions](#).

Table 28. Typical terminal interactions	
Operator Input	System Response
CEMT	Status: ENTER ONE OF THE FOLLOWING Discard Inquire Perform Set
I	Status: ENTER ONE OF THE FOLLOWING OR PRESS ENTER FOR DEFAULT (Followed by a list of options)
PROG Press ENTER key	STATUS: RESULTS - OVERTYPE TO MODIFY Prog(CEECBLDY) Len(00000000) Ass Pro Ena Pri Res(000) Use(000000000000) Any Cex Ful
Press F3 key Press CLEAR key	
CEMT PERFORM STATISTICS	
Press F3 key Press CLEAR key	SESSION ENDED
CETR	
Press F3 key Press CLEAR key	Clear or F3 pressed Normal termination of CETR
CEMT I TA	Displays list of tasks in the system
Press F3 key Press CLEAR key	SESSION ENDED

Table 28. Typical terminal interactions (continued)	
Operator Input	System Response
CEMT I PROG(DFHFEP)	Prog(DFHFEP)Len(005848) Ass Pro Ena Pri Res(000) Use(0000000) Any Cex Ful Qua
Press F3 key Press CLEAR key	SESSION ENDED
CEOT (Inquire about this terminal)	Ter (tmid) Tra (CEOT) Pri (nnn) Pag Ins Ati Tti (Remember 'tmid' for use in next transaction, CMSG)
Press F3 key Press CLEAR key	SESSION ENDED
	
CMSG 'HELLO',R=tmid,S	(Send the message 'HELLO' to your terminal) MRS OK MESSAGE HAS BEEN ROUTED (briefly at lower right of screen) HELLO (at upper left of screen)

You can enter your CEMT input in either uppercase or lowercase, because the main terminal transaction translates all input to uppercase. Use the CLEAR key and the F3 key as indicated.

If you enter the CETR transaction, CICS displays the status of the various trace options.

You can alter the status of any of the trace options by over typing the current value, indicated by ==> on the CETR display.

Using the CEDA transaction

When DFHIVPOL starts up CICS, system initialization specifies GRPLIST=DFHLIST, which causes all the CICS resource definitions that are required for normal running to be installed.

You can see which resources are included in DFHLIST by using the CEDA transaction; for example, by using the CEDA EXPAND LIST(DFHLIST) command.

Press F8 to see the continuation of the list. If you started the DFHIVPOL job with your own group list specified instead of the DFHLIST group list, specify the name of your list in the CEDA EXPAND command. The CICS-defined groups all begin with "DFH". For information about CEDA and the interactions for a typical sequence of CEDA commands, see [CEDA - resource definition online](#).

The DFHLIST group list does not include any of the sample applications groups, the group names of which all begin with "DFH\$". To use the sample programs, therefore, you must first install the resource definitions for the required samples. For example, to use the FILEA sample application:

1. Install the sample programs that are required for the FILEA applications. Use this command:

```
CEDA INSTALL GROUP(DFH$AFLA)
```

2. Make the FILEA data set available to CICS, by performing one of the following tasks:

- Install a FILE resource definition for the FILEA data set:

```
CEDA INSTALL GROUP(DFH$FILEA)
```

- Provide a DD statement for the FILEA data set in your CICS startup JCL. For example,

```
//FILEA DD DISP=SHR,DSN=CICSTS56.CICS.CICSHTH1.FILEA
```

To end the CEDA session, press F3.

Invoking and running sample programs

To try the assembler language version of the FILEA sample application, install group DFH\$AFLA and then enter the AMNU transaction.

Using transactions from a console device

You can invoke CICS transactions other than CECI from a console device, and other CICS operators can communicate with the console operator. In particular, you can use the console device for CICS main terminal functions to control CICS terminals or to control several CICS regions with multiregion operation.

Normal operating-system use of the console device is not inhibited, and CICS supports multiple console devices where present.

- The CEDA transaction can be used from a console device only to INSTALL resource definitions.
- The CECI transaction and the sample programs cannot be used from a console device.

If you issue the MVS command `d consoles`, a list of console devices is displayed. This list identifies the console devices by name.

You can use a console device to submit MODIFY commands from your job stream if you define a console device in your CSD as `CONSNAME(INTERNAL)`.

For further information about defining consoles, see [Defining console devices to CICS](#).

To enter a command, use:

```
{MODIFY|F} ♦ jobname,[']command[']
```

where:

jobname

Is the region identifier for the CICS region. This identifier is either the name of the job being used to run CICS, for example, DFHIVPOL, or the name of a procedure if CICS was initiated as a started task.

command

Is a string of data, starting with a CICS transaction identifier. If the transaction requires further input, the operator is prompted in the same way as any normal terminal operator. The message from CICS contains a reply number that must be quoted in the reply.

You can use the commands shown in [Figure 25 on page 213](#) to verify the CEMT and CEOT transactions from the MVS console.

Entering commands from TSO

A TSO user can enter CICS commands after invoking the TSO command `CONSOLE`, in either of two formats.

```
CONSOLE {MODIFY|F} cicsid,[']command[']
```

```
CONSOLE  
{MODIFY|F} ♦ cicsid,[']command[']  
END
```

When the TSO CONSOLE command is used, TSO checks the user for authority to issue console commands. Further, if console operator command security is active, the TSO user must be specifically authorized to issue `MODIFY cicsid`.

The TSO user can interact with an alternate CICS by using the command `CONSOLE MODIFY altcics,CEBT`.

You can also use TSO CLIST processing to issue sequences of CICS commands.

Operator input	System response
f dfhivpol,'cemt i terminal'	Displays a list of terminals attached to CICS
f dfhivpol,'cemt i dump'	Displays status of transaction dump data sets
f dfhivpol,'cemt p statistics'	CICS writes statistics to SMF data sets
f dfhivpol,'cemt i ta'	Displays number and types of tasks currently running
f dfhivpol,'cemt p dump'	CICS invokes SDUMP macro for system dump to be taken
f dfhivpol,'cemt i prog(dfhpep)'	Displays details of DFHPEP module
f dfhivpol,'ceot'	Displays details of operator console
f dfhivpol,'cemt i journalname'	Displays status of CICS logs

Figure 25. Using an MVS console for main terminal operations

Stopping CICS

To stop CICS, enter `CEMT P SHUT` from the SNA LU or MVS console. `CEMT P SHUT` is a short form of `CEMT PERFORM SHUTDOWN`.

The system responds with message DFH1713 and following messages as shown in the sample job log shown in [Figure 24 on page 209](#).

Verifying shared data tables support

You can verify that shared data tables can be used in a number of ways.

1. Start a CICS region on which you have installed support for shared data tables. To use shared data tables, you must install the DFHDTSVC, DFHDTCV, and DFHMVRMS modules in either an authorized system library in the MVS linklist (LNKLST concatenation of the MVS system) or in the LPA. When you install CICS, these modules are installed into the *hlq.SDFHLINK* library, which you usually include in the MVS linklist.
2. Define and install a user-maintained data table.
3. Try a generic read command on your data table, using the CECI transaction. Generic reads of user-maintained data tables are allowed only with shared data tables. If shared data tables is operational, you see a normal response. If shared data tables is not operational, you see an INVREQ response. This verification process uses user-maintained data tables throughout, because the behavior of CICS-maintained data tables is not apparent to their users. For example, a normal response is returned for a generic read of a CICS-maintained data table, regardless of whether or not shared data tables is operational.

To verify that the cross-memory services of shared data tables are working:

1. Start a second CICS region (the *requester*) that has an interregion communication connection to the first CICS region (the *server*, which contains the user-maintained data table and source data set).
2. On the requester CICS region, do the following tasks:

- a. Define and install a remote file referring to (associated with) the user-maintained data table on the server CICS region.
- b. Close the interregion communication connection between the two CICS regions so that function shipping is impossible; that is, only the cross-memory services of shared data tables can be used to access the shared data table from the requester CICS region. To close the connection, you can enter the command:

```
CEMT SET IRC CLOSED
```

To verify that function shipping cannot work, try a remote READ of a file (not a data table) on the server CICS region; you get a SYSIDERR response.

- c. Try a generic read command on your data table, using the CECI transaction. If the cross-memory services of shared data tables can be used, you see a normal response.
3. To restore interregion communication between the two CICS regions, open the connection again. To do this, you can enter the command:

```
CEMT SET IRC OPEN
```

Example verification of shared data tables

This example shows the steps to perform to verify the shared data tables for the CICS shared data tables environment.

1. A CICS region, CICSIDC, is started. CICSIDC is the server CICS region in this example.
2. On CICSIDC, the following steps are completed:
 - a. The user-maintained data table, MYSDT, is defined and installed. The MYSDT data table is based on the sample data set, *hlq.CICSIDC.FILEA*, installed on that region.
 - b. The following generic READ command is entered at a terminal:

```
CECI READ FILE(MYSDT) RIDFLD(00092) KEYLENGTH(5) GE GTEQ
```

[Figure 27 on page 215](#) shows the initial response (LOADING), and [Figure 28 on page 216](#) shows the subsequent response when the command is repeated after the data table has completed loading.

The following steps were completed to verify the cross-memory services of shared data tables:

1. A second CICS region, CICSIDA, is started with support for shared data tables. CICSIDA is the requester CICS region in this example.
2. The following IRC connections and sessions are defined and installed on the associated CICS regions:

Region	CONNECTION	SESSION
CICSIDA	CICA	ATOC
CICSIDC	CICC	CTOA

See [Figure 31 on page 217](#) and [Figure 32 on page 217](#) for the CICA and ATOC resource definitions attributes. The attributes for the CICC and CTOA resource definitions are similar.

3. On CICSIDA, the following steps are completed:
 - a. The file, REMSDT, is defined and installed as remote, referring to the MYSDT data table on CICSIDC. See [Figure 33 on page 218](#) for the parameters used for the REMSDT resource definition.
 - b. The file, REMFIL, is defined and installed as remote, referring to the FILEA sample file on CICSIDC.
 - c. The CEMT SET IRC CLOSED command is used to close the IRC connection to CICSIDC.
 - d. The following generic READ command is entered at a terminal:

```
CECI READ FILE(REMFIL) RIDFLD(00092) KEYLENGTH(5)
LENGTH(80) GE GTEQ
```

Figure 29 on page 216 shows the SYSIDERR response, because the remote file cannot be accessed by function-shipping. You also see this response for the remote data table, REMSDT, if the IRC connection is closed.

e. The following generic READ command is entered at a terminal:

```
CECI READ FILE(REMSDT) RIDFLD(00092) KEYLENGTH(5)
LENGTH(80) GE GTEQ
```

Figure 30 on page 217 shows the NORMAL response. You receive a NORMAL response only if MYSDT is already open on CICSIDC, as achieved in step “2.b” on page 214.

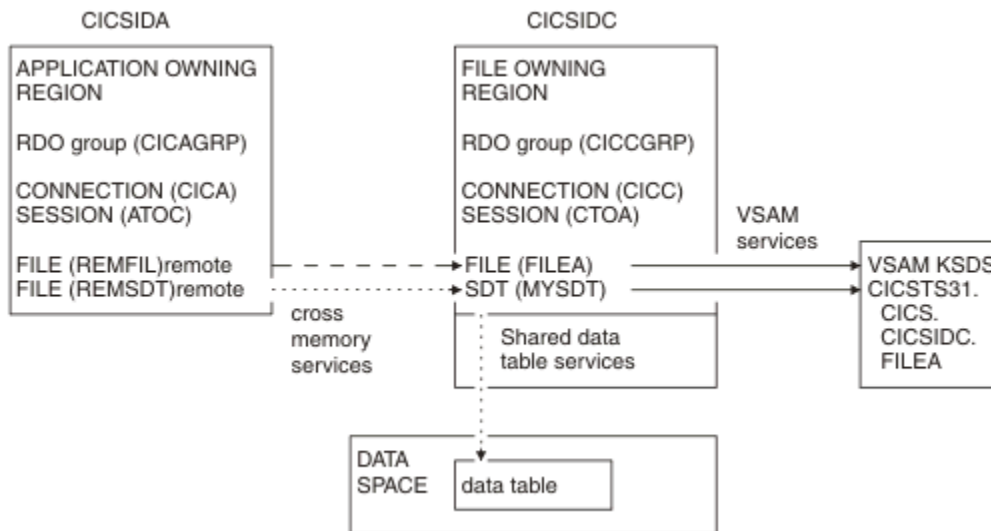


Figure 26. CICS environment for example verification of shared data tables.

```
read file(MYSDT) ridfld(00092) keylength(5) ge gteq
STATUS: COMMAND EXECUTION COMPLETE                NAME=
EXEC CICS READ
  File( 'MYSDT ' )
  < SYsid() >
  ( SEt() | Into( ' ' ) )
  < Length( +00000 ) >
  RIdfld( '00092' )
  < Keylength( +00005 ) < GEneric > >
  < RBa | RRn | DEBRec | DEBKey >
  < GTeq | Equal >
  < UNcommitted | Consistent | REpeatable | UPdate <token()> >
  < Nosuspend >

RESPONSE: LOADING                                EIBRESP=+00000000094 EIBRESP2=+00000000104
PF 1 HELP 2 HEX 3 END 4 EIB 5 VAR 6 USER 7 SBH 8 SFH 9 MSG 10 SB 11 SF
```

Figure 27. On CICSIDC, response to initial CECI generic READ FILE command with SDT support.

```

read file(MYSDT) ridfld(00092) keylength(5) ge gteq
STATUS: COMMAND EXECUTION COMPLETE NAME=
EXEC CICS READ
  File( 'MYSDT ' )
  < SYsid() >
  ( SEt()
    | Into( ' 000983J. S. TILLING WASHINGTON, DC 34512' ... ) )
  < Length( +00080 ) >
  RIdfld( '00092' )
  < Keylength( +00005 ) < GEneric > >
  < RBa | RRn | DEBRec | DEBKey >
  < GTeq | Equal >
  < UNcommitted | Consistent | REpeatable | UPdate <token()> >
  < Nosuspend >

```

```

RESPONSE: NORMAL EIBRESP=+0000000000 EIBRESP2=+0000000000
PF 1 HELP 2 HEX 3 END 4 EIB 5 VAR 6 USER 7 SBH 8 SFH 9 MSG 10 SB 11 SF

```

Figure 28. On CICSIDC, response to CECI generic READ FILE command with SDT support.

```

read file(FILEA) ridfld(00092) keylength(5) length(80) ge gteq
STATUS: COMMAND EXECUTION COMPLETE NAME=
EXEC CICS READ
  File( 'FILEA ' )
  < SYsid() >
  ( SEt()
    | Into( ' ' ... ) )
  < Length( +00080 ) >
  RIdfld( '00092' )
  < Keylength( +00005 ) < GEneric > >
  < RBa | RRn | DEBRec | DEBKey >
  < GTeq | Equal >
  < UNcommitted | Consistent | REpeatable | UPdate <token()> >
  < Nosuspend >

```

```

RESPONSE: SYSIDERR EIBRESP=+0000000053 EIBRESP2=+00000000130
PF 1 HELP 2 HEX 3 END 4 EIB 5 VAR 6 USER 7 SBH 8 SFH 9 MSG 10 SB 11

```

Figure 29. On CICSIDA, response to remote CECI generic READ FILE command, with IRC closed.


```

read file(MYSDT) ridfld(00092) keylength(5) length(80) ge gteq
STATUS:  COMMAND EXECUTION COMPLETE          NAME=
EXEC CICS READ
  File( 'MYSDT' )
  < Sysid() >
  ( SET()
    | Into( ' 000983J. S. TILLING      WASHINGTON, DC      34512' ... ) )
  < Length( +00080 ) >
  RIDfld( '00092' )
  < Keylength( +00005 ) < GGeneric > >
  < RBa | RRn | DEBRec | DEBKey >
  < GTeq | Equal >
  < UNcommitted | Consistent | REpeatable | UDate <token()> >
  < Nosuspend >

```

```

RESPONSE: NORMAL          EIBRESP=+00000000000 EIBRESP2=+00000000000
PF 1 HELP 2 HEX 3 END 4 EIB 5 VAR 6 USER 7 SBH 8 SFH 9 MSG 10 SB 11 SF

```

Figure 30. On CICSIDA, response to remote CECI generic READ FILE command, with IRC closed.

```

OBJECT CHARACTERISTICS                      CICS RELEASE = 0730

Connection      : CICA
Group           : CICAGRP
DEscription     : MRO CONNECTION CICSIDA TO CICSIDC
CONNECTION IDENTIFIERS
Netname         : CICSIDC
INDsys         :
REMOTE ATTRIBUTES
REMOTESystem    :
REMOTENAME     :
CONNECTION PROPERTIES
ACcessmethod    : IRC                Vtam | IRc | INdirect | Xm
Protocol        :                   Appc | Lu61
SIngleseSS      : No                No | Yes
DATAstream      : User              User | 3270 | SCs | STrfield | Lms
RECORDformat    : U                 U | Vb
OPERATIONAL PROPERTIES
AUtoconnect     : No                No | Yes | All
INService       : Yes               Yes | No

```

Figure 31. Example CONNECTION resource definition, CICA, installed on CICSIDA.

Note: VTAM is now the z/OS Communications Server.

```

OBJECT CHARACTERISTICS                      CICS RELEASE = 0730

Sessions        : ATOC
Group           : CICAGRP
DEscription     : SESSION FOR MRO CICA TO CICC
SESSION IDENTIFIERS
Connection      : CICA
SESSName        :
NETnameq        :
MODename        :
SESSION PROPERTIES
Protocol        : Lu61                Appc | Lu61
MAXimum         : 000 , 000           0-999
RECEIVEPfx      : RB
RECEIVECount    : 005                 1-999
SENDPfx         : SB
SENDCount       : 003                 1-999
SENDSize        : 04096               1-30720
RECEIVESize     : 04096               1-30720
SESSPriority     : 100                 0-255

```

Figure 32. Example SESSION resource definition, ATOC, associated with connection, CICA.

OBJECT CHARACTERISTICS

CICS RELEASE = 0730

```

File           : REMSDT
Group          : CICCGRP
DEscription    :
VSAM PARAMETERS
DSName         :
Password       :
Rlsaccess      : No           PASSWORD NOT SPECIFIED
Lsrpoolid      : 1           No | Yes
READInteg      : Uncommitted 1-8 | None
DSNSharing     : Allreqs      Uncommitted | Consistent | Repeat
STRings        : 001         Allreqs | Modifyreqs
Nsrgroup       :
REMOTE ATTRIBUTES
REMOTESystem    : CICC
REMOTENAME      : MYSDT
RECORDSize      : 1-32767
Keylength       : 1-255
INITIAL STATUS
STATus          : Enabled     Enabled | Disabled | Unenabled

```

Figure 33. Example remote FILE resource definition, REMSDT, installed on CICSIDA.

Verifying the CICS-DBCTL interface

You can use the installation verification procedure, DFHIVPDB, to verify that the CICS-DBCTL interface can be used successfully.

To run the DFHIVPDB job successfully:

1. Tailor the DFHIVPDB job to your CICS and IMS environment.

You can tailor DFHIVPDB as part of the process of tailoring all CICS sample postinstallation jobs, as described in [“Defining DL/I support”](#) on page 192. When you run the DFHISTAR job as part of the CICS installation process, the DFHIVPDB job is installed in the *hlq.XDFHINST* library.

Change the prefix of the IMS.SDFSRESL library, previously called IMS.RESLIB, in the DFHIVPDB job to the prefix that you use for your IMS libraries.

2. Create the data sets required by the CICS region used by the DFHIVPDB job.

To create the data sets, you can tailor and run copies of the following CICS sample jobs:

DFHCOMDS

This job creates the CICS data sets common to all CICS regions.

DFHDEFDS

This job creates the data sets required for each CICS region.

When you run the DFHISTAR job as part of the CICS installation process, these jobs are installed in the *hlq.XDFHINST* library.

3. Run the IMS installation verification procedures, as outlined in [“The IMS installation requirements for the DFHIVPDB job”](#) on page 218.

The IMS installation requirements for the DFHIVPDB job

The DFHIVPDB job depends on running the IMS installation verification procedures, as part of the INSTALL/IVP process.

The following assumptions about the IMS INSTALL/IVP process are made:

1. The IMS sample database, DI21PART, has been successfully defined. This database comprises two data sets:
 - DI21PART
 - DI21PARO
2. The DI21PART database has been loaded with the IMS-supplied sample data.

3. The following IMS-supplied procedures have been installed in an executable procedure library:

- ACBGEN
- PSBGEN

4. The sample DRA startup table, DFSPZPIV, has been built and installed in the IMS.SDFSRESL library, previously called IMS.RESLIB.

5. The sample DBCTL system, IVP3, is available.

For information about installing IMS, the INSTALL/IVP process, and running the IMS IVPs, see [Installation in IMS product documentation](#).

The DFHIVPDB job steps

The DFHIVPDB job consists of the GEN and CICS job steps.

1. **GEN.** This step unloads the member DFH\$DBAN from the *hlq*.SDFHSAMP library into a temporary sequential data set called CARDIN. This member contains the transactions to invoke the assembler versions of the DL/I sample applications that CICS reads from CARDIN as soon as initialization is complete. The sequential data set CARDIN is defined in the sample terminal control table, DFHTCT5\$, as a simulated terminal.

The COBOL version, DFH\$DBCB, and the PL/I version, DFH\$DBPL, of the sample DL/I transactions are also in the *hlq*.SDFHSAMP library. If you want to run the COBOL or PL/I versions, modify this job step to load CARDIN with the appropriate member.

Output generated by the transactions is sent to a similar device; a sequential data set defined as PRINTER.

2. **CICS.** This job step issues the DFHSTART procedure to start CICS, with the CICS-supplied resource group list DFH\$IVPL. CICS attempts to connect to the DBCTL system IVP3, run the sample DLI transactions, and then shut down the CICS region. If the DBCTL system, IVP3, is not running, the sample DLI transactions stop with an abend message.

If you want to examine the sample members used by this IVP, here is a list of them, indicating where you can find each one:

DFHIVPDB

This IVP contains some explanatory comments, and was installed in the *hlq*.XDFHINST library when you ran the DFHISTAR job. For details of the DFHISTAR job, see [“Tailoring the CICS-supplied skeleton jobs” on page 167](#).

DFH\$SIP5

DFH\$SIP5 is the member of the *hlq*.SYSIN data set that contains the system initialization parameter overrides specific to the DFHIVPDB job. You can specify other system initialization parameters (for example, APPLID, CICSSVC, and DFLTUSER) for the DFHIVPDB job; the DFH\$SIP5 member of the *hlq*.SYSIN data set is a convenient place to do so.

DFHTCT5\$

DFHTCT5\$ is the sample TCT that specifies the sequential devices that CICS uses in this IVP as a simulated terminal, with a terminal name of SAMA. The source statements are in the DFH\$TCTS member of the *hlq*.SDFHSAMP library.

Running the DFHIVPDB job

Before you submit the DFHIVPDB job, run the DFHRMUTL program to reset the global catalog control record to perform an INITIAL start on the next CICS startup.

Here is an example of the DFHRMUTL program:

```
//DFHRMUTI JOB 24116475, 'DFHRMUTL',  
//          CLASS=A,MSGCLASS=H,NOTIFY=userid  
//*  
//*-----*/  
//* RESET GLOBAL CATALOG CONTROL RECORD TO INITIAL START */  
//*-----*/
```

```
//DFHRMUTL EXEC PGM=DFHRMUTL,REGION=1M
//STEPLIB DD DSN=CICSTS56.CICS.SDFHLOAD,DISP=SHR
//SYSPRINT DD SYSOUT=*
//DFHGCD DD DSN=CICSTS56.CICS.DBDCICX.DFHGCD,DISP=OLD
//SYSIN DD *
SET_AUTO_START=AUTOINIT
/*
```

When you are satisfied that you have made all the necessary preparations, and that all the prerequisite jobs have been run, submit the DFHIVPDB job. The job loads the DL/I transactions into CARDIN. CICS reads the transactions, and sends the output to the PRINTER sequential data set. The process is as follows:

- The first transaction copied from the DFH\$DBAN member of the *hlq*.SDFHSAMP library to CARDIN is CDBC CONNECT SUFFIX(IV). This transaction connects CICS to DBCTL, using the sample DRA startup table, DFSPZPIV.
- The final transaction copied from the DFH\$DBAN member of the *hlq*.SDFHSAMP library to CARDIN is CEMT PERFORM SHUT.
- If you want to use some commands online before CICS shuts down, delete the CEMT command before you run the job. You can then issue CEMT, CEDA, and other CICS-supplied transactions, and initiate a shutdown either from a CICS terminal or through an MVS console.
- If you want to communicate with CICS through an MVS console, you must define a console to CICS before you start DFHIVPDB, as described in [Defining console devices to CICS](#).
- If you want to enter MODIFY commands from terminals connected to TSO, you must define the TSO users as console devices, as described in [Defining console devices to CICS](#).

A sample job log from a run of the DFHIVPDB job is given in [Figure 34 on page 222](#). The results you get from the transaction processing are similar to those shown in [Figure 34 on page 222](#), [Figure 35 on page 223](#), and [Figure 36 on page 223](#).

```

1      J E S 2   J O B   L O G   --   S Y S T E M   M V 2 6   --   N O D E   W I N M V S 2 C
0
09.36.19 JOB36923 ---- TUESDAY, 24 APR 2007 ----
09.36.19 JOB36923 ICH70001I CICINST  LAST ACCESS AT 08:27:32 ON TUESDAY, APRIL 24, 2007
09.36.19 JOB36923 $HASP373 DFHIVPDB STARTED - INIT 4      - CLASS A - SYS MV26
09.36.19 JOB36923 IEF403I DFHIVPDB - STARTED - TIME=09.36.19
09.36.19 JOB36923 -
09.36.19 JOB36923 -TIMINGS (MINS.)--          ---PAGING COUNTS---
STEPNO  -JOBNAME  STEPNAME  PROCSTEP    RC    EXCP    CPU    SRB    CLOCK    SERV  PG    PAGE    SWAP    VIO  SWAPS
09.36.19 JOB36923 -DFHIVPDB          GEN          00     53    .00    .00    .00    184    0      0      0      0      0
09.36.19 JOB36923 -DFHIVPDB CICS      CICSCNTL    01     16    .00    .00    .00    148    0      0      0      0      0
09.36.19 JOB36923 -DFHIVPDB CICS      DTCNTL     01     15    .00    .00    .00    161    0      0      0      0      0
09.36.20 JOB36923 DFHPA1101  CICSI1V1 DFHSIT  IS BEING LOADED.
09.36.20 JOB36923 DFHPA1108  CICSI1V1 DFHSIT  HAS BEEN LOADED. (GENERATED AT: MM/DD= 01/12 HH:MM= 13:57).
09.36.20 JOB36923 DFHPA1100  CICSI1V1 OVERRIDE PARAMETERS FROM JCL EXEC STATEMENT: START=AUTO,SYSLIN
09.36.20 JOB36923 DFHPA1102  CICSI1V1 OVERRIDE PARAMETERS FROM SYSIN: 1
09.36.20 JOB36923 DFHPA1927  CICSI1V1 GRPLIST=DFH$IVPL,  INCLUDE DLI SAMPLE PROGRAMS & TRANSACTIONS      20000000
09.36.20 JOB36923 DFHPA1927  CICSI1V1 FCT=NO,          TCT INCLUDES SEQ DEVICES      30000000
09.36.20 JOB36923 DFHPA1927  CICSI1V1 TCT=5$,          TCT INCLUDES SEQ DEVICES      40000000
09.36.20 JOB36923 DFHPA1927  CICSI1V1 XRF=NO,          TCT INCLUDES SEQ DEVICES      50000000
09.36.20 JOB36923 DFHPA1927  CICSI1V1 STNTR=OFF,          TCT INCLUDES SEQ DEVICES      53000000
09.36.20 JOB36923 DFHPA1927  CICSI1V1 STNTRFC=1,          TRACE FILE CONTROL AND DLI EVENTS 56000000
09.36.20 JOB36923 DFHPA1927  CICSI1V1 AUXTR=ON,          TRACE FILE CONTROL AND DLI EVENTS 60000000
09.36.20 JOB36923 DFHPA1927  CICSI1V1 AUXTRSW=NEXT,          TRACE FILE CONTROL AND DLI EVENTS 70000000
09.36.20 JOB36923 DFHPA1927  CICSI1V1 SRT=NO,          TRACE FILE CONTROL AND DLI EVENTS 70000100
09.36.20 JOB36923 DFHPA1927  CICSI1V1 SEC=NO,          TRACE FILE CONTROL AND DLI EVENTS 70000200
09.36.20 JOB36923 DFHPA1927  CICSI1V1 TRTABSZ=64,          TRACE FILE CONTROL AND DLI EVENTS 70000300
09.36.20 JOB36923 DFHPA1927  CICSI1V1 APPLID=CICSI1V1,          TRACE FILE CONTROL AND DLI EVENTS 70000400
09.36.20 JOB36923 DFHPA1927  CICSI1V1 CICSSVC=233,          TRACE FILE CONTROL AND DLI EVENTS 70000500
09.36.20 JOB36923 DFHPA1927  CICSI1V1 .END          TRACE FILE CONTROL AND DLI EVENTS 80000000
09.36.20 JOB36923 DFHPA1103  CICSI1V1 END OF FILE ON SYSIN. 09.36.21 JOB36923 +DFHTR0103 TRACE TABLE SIZE IS 64KB
09.36.21 JOB36923 +DFHSM0122I CICSI1V1 Limit of DSA storage below 16MB is 5 120KB.
09.36.21 JOB36923 +DFHSM0123I CICSI1V1 Limit of DSA storage above 16MB is 48MB.
09.36.21 JOB36923 +DFHSM0113I CICSI1V1 Storage protection is not active.
09.36.21 JOB36923 +DFHSM0126I CICSI1V1 Transaction isolation is not active.
09.36.21 JOB36923 +DFHDM0101I CICSI1V1 CICS is initializing.
09.36.23 JOB36923 +DFHWM0109I CICSI1V1 Web domain initialization has started.
09.36.23 JOB36923 +DFHSO0100I CICSI1V1 Sockets domain initialization has started.
09.36.23 JOB36923 +DFHRX0100I CICSI1V1 RX domain initialization has started.
09.36.23 JOB36923 +DFHRX0101I CICSI1V1 RX domain initialization has ended.
09.36.23 JOB36923 +DFHLG0101I CICSI1V1 Log manager domain initialization has started.
09.36.23 JOB36923 +DFHEJ0101 CICSI1V1 790
790      Enterprise Java domain initialization has started. Java is a
790      trademark of Oracle.
09.36.23 JOB36923 +DFHDM0100I CICSI1V1 Document domain initialization has started.
09.36.23 JOB36923 +DFHXS1100I CICSI1V1 Security initialization has started.
09.36.23 JOB36923 +DFHSI1500 CICSI1V1 CICS startup is in progress for CICS Transaction Server
09.36.23 JOB36923 +DFHXS1102I CICSI1V1 Security is inactive.
09.36.23 JOB36923 +DFHDM0104I CICSI1V1 Transaction Dump Data set DFHDMPB opened.
09.36.23 JOB36923 +DFHSI1501I CICSI1V1 Loading CICS nucleus.
09.36.26 JOB36923 +DFHTR0113 CICSI1V1 Auxiliary trace is being started on data set DFHAUTX.
09.36.26 JOB36923 +DFHCQ0100I CICSI1V1 Console queue initialization has started.
09.36.26 JOB36923 +DFHCQ0101I CICSI1V1 Console queue initialization has ended.
09.36.26 JOB36923 +DFHCQ0103I CICSI1V1 MVS console queue is open.
09.36.26 JOB36923 +DFHCQ0200I CICSI1V1 CEKL transaction enabled.
09.36.26 JOB36923 +DFHXS1101I CICSI1V1 Security initialization has ended.
09.36.26 JOB36923 +DFHRM0140 CICSI1V1 Recovery manager autostart override found with value: 'AUTOINIT'.
09.36.26 JOB36923 +DFHRM0149I CICSI1V1 Recovery manager autostart override record will be deleted.
09.36.26 JOB36923 +DFHDM0101I CICSI1V1 Document domain initialization has ended.
09.36.26 JOB36923 +DFHMM0105I CICSI1V1 Using default Monitoring Control Table.
09.36.26 JOB36923 +DFHSO0101I CICSI1V1 Sockets domain initialization has ended.
09.36.26 JOB36923 +DFHWM0110I CICSI1V1 Web domain initialization has ended.
09.36.26 JOB36923 +DFHMM0110I CICSI1V1 CICS Monitoring is inactive.
09.36.26 JOB36923 +DFHSI1502I CICSI1V1 CICS startup is Initial.
09.36.26 JOB36923 +DFHTS0100I CICSI1V1 Temporary Storage initialization has started.
09.36.26 JOB36923 +DFHSI1503I CICSI1V1 Terminal data sets are being opened.
09.36.27 JOB36923 +DFHLG0102I CICSI1V1 Log manager domain initialization has ended.
09.36.27 JOB36923 IEC161I 080-053,DFHIVPDB,CICS CICS,DFHTEMP,,,
09.36.27 JOB36923 IEC161I INST.CICSTS56.CNTL.CICS.DFHTEMP,
09.36.27 JOB36923 IEC161I INST.CICSTS56.CNTL.CICS.DFHTEMP.DATA,
09.36.27 JOB36923 IEC161I ICFCAT.SYSPLEX2.CATALOGB
09.36.27 JOB36923 +DFHTS0102I CICSI1V1 About to format the temporary storage data set (359 control intervals).
09.36.27 JOB36923 +DFHKE0406I CICSI1V1 825
825      CICS is about to wait for predecessors defined in the MVS automatic
825      restart management policy for this region.
09.36.27 JOB36923 +DFHKE0412I CICSI1V1 CICS WAITPRED call to automatic restart manager has completed.
09.36.27 JOB36923 +DFHCP0101I CICSI1V1 CPI initialization has started.
09.36.27 JOB36923 +DFHPR0104I CICSI1V1 Partner resource manager initialization has started.
09.36.27 JOB36923 +DFHAI0101I CICSI1V1 AITM initialization has started.

```

```

09.36.27 JOB36923 +DFHFC0100I CICSIVP1 File Control initialization has started.
09.36.27 JOB36923 +DFHTD0100I CICSIVP1 Transient Data initialization has started.
09.36.27 JOB36923 +DFHFC0101I CICSIVP1 File Control initialization has ended.
09.36.27 JOB36923 +DFHTD0101I CICSIVP1 Transient Data initialization has ended.
09.36.27 JOB36923 +DFHTS0101I CICSIVP1 Temporary Storage initialization has ended.
09.36.27 JOB36923 +DFHCP0102I CICSIVP1 CPI initialization has ended.
09.36.27 JOB36923 +DFHPR0105I CICSIVP1 Partner resource manager initialization has ended.
09.36.27 JOB36923 +DFHAI0102I CICSIVP1 AITM initialization has ended.
09.36.28 JOB36923 +DFHSI1511I CICSIVP1 Installing group list DFH$IVPL.
09.36.29 JOB36923 IEC031I D37-04,IFG0554P,DFHIVPDB,CICS,DFHAUXT,D306,P2P0C6,INST.CICSTS56.CICS.DFHAUXT
09.36.29 JOB36923 +DFHTR0110 - AUXILIARY TRACE DATA SET DFHAUXT FULL - SWITCHING TO DFHBUXT
09.36.29 JOB36923 IEC031I D37-04,IFG0554P,DFHIVPDB,CICS,DFHBUXT,D50B,P2P14B,INST.CICSTS56.CICS.DFHBUXT
09.36.29 JOB36923 +DFHTR0109 - AUXILIARY TRACE DATA SET DFHBUXT FULL - AUXILIARY TRACE HAS BEEN STOPPED
09.36.30 JOB36923 +DFHLG0103I CICSIVP1 System log (DFHLOG) initialization has started.
09.36.31 JOB36923 +DFHLG0104I CICSIVP1 844
844 System log (DFHLOG) initialization has ended. Log stream
844 CICINST.CICSIVP1.DFHLOG is connected to structure LOG_GENERAL_008.
09.36.31 JOB36923 +DFHLG0103I CICSIVP1 System log (DFHSHUNT) initialization has started.
09.36.31 JOB36923 +DFHLG0104I CICSIVP1 846
846 System log (DFHSHUNT) initialization has ended. Log stream
846 CICINST.CICSIVP1.DFHSHUNT is connected to structure LOG_GENERAL_008.
09.36.31 JOB36923 +DFHAP1203I CICSIVP1 Language Environment is being initialized.
09.36.31 JOB36923 +DFHAP1200 CICSIVP1 A CICS request to the Language Environment has failed. Reason code '0011020'.
09.36.31 JOB36923 +DFHAP1208 CICSIVP1 Language Environment cannot support the COBOL language. 2
09.36.31 JOB36923 +DFHAP1209 CICSIVP1 Language Environment cannot support the C/C++ languages. 2
09.36.31 JOB36923 +DFHAP1210 CICSIVP1 Language Environment cannot support the PL/I language. 2
09.36.31 JOB36923 +DFHAP1211I CICSIVP1 Language Environment initialization completed.
09.36.31 JOB36923 +DFHWP1007 CICSIVP1 Initializing CICS Web environment.
09.36.32 JOB36923 +DFHWP1008 CICSIVP1 CICS Web environment initialization is complete.
09.36.32 JOB36923 +DFHGI1517 CICSIVP1 Control is being given to CICS.
09.36.32 JOB36923 +DFHEJ0102 CICSIVP1 Enterprise Java domain initialization has ended.
09.37.54 JOB36923 +DFHTM1715 CICSIVP1 CICS is being quiesced by userid CICSUSER in transaction CEMT at terminal SAMA.
09.37.54 JOB36923 +DFHDM0102I CICSIVP1 CICS is quiescing.
09.37.54 JOB36923 +DFHDB8122I CICSIVP1 CICS is about to disconnect from DBCTL for CICS shutdown.
09.37.54 JOB36923 +DFHGESD CICSIVP1 SHUTDOWN ASSIST TRANSACTION CESD STARTING. SHUTDOWN IS NORMAL.
09.37.54 JOB36923 +DFHDB8123I CICSIVP1 CICS disconnection from DBCTL for CICS shutdown has completed successfully.
09.37.54 JOB36923 +DFHTM1782I CICSIVP1 All non-system tasks have been successfully terminated.
09.37.55 JOB36923 +DFHZZC2305I CICSIVP1 Termination of VTAM sessions beginning
09.37.55 JOB36923 +DFHZZC2316 CICSIVP1 VTAM ACB is closed
09.37.55 JOB36923 +DFHCQ0104I CICSIVP1 MVS console queue is closed.
09.37.58 JOB36923 +DFHRM0204 CICSIVP1 There are no indoubt, commit-failed or backout-failed UOWs.
09.37.59 JOB36923 +DFHRM0130 CICSIVP1 Recovery manager has successfully quiesced.
09.37.59 JOB36923 +DFHUU0303I CICSIVP1 Transaction Dump Data set DFHDMPB closed.
09.37.59 JOB36923 +DFHKE1799 CICSIVP1 TERMINATION OF CICS IS COMPLETE.
09.37.59 JOB36923 -DFHIVPDB CICS CICS 00 4070 .03 .00 1.66 23769 0 0 0 0
0 4
09.38.00 JOB36923 -DFHIVPDB CICS PRTDMPA 00 137 .00 .00 .00 289 0 0 0 0
0 5
09.38.00 JOB36923 -DFHIVPDB CICS PRTDMPB 00 138 .00 .00 .00 291 0 0 0 0
0 6
09.38.03 JOB36923 -DFHIVPDB CICS PRTAUXT 00 1935 .01 .00 .04 13326 0 0 0 0
0 7
09.38.04 JOB36923 $HASP375 DFHIVPDB ESTIMATE LINES EXCEEDED
09.38.04 JOB36923 $HASP375 DFHIVPDB ESTIMATE EXCEEDED BY 5,000 LINES
09.38.05 JOB36923 $HASP375 DFHIVPDB ESTIMATE EXCEEDED BY 10,000 LINES
09.38.05 JOB36923 $HASP375 DFHIVPDB ESTIMATE EXCEEDED BY 15,000 LINES
09.38.05 JOB36923 $HASP375 DFHIVPDB ESTIMATE EXCEEDED BY 20,000 LINES
09.38.05 JOB36923 $HASP375 DFHIVPDB ESTIMATE EXCEEDED BY 25,000 LINES
09.38.06 JOB36923 $HASP375 DFHIVPDB ESTIMATE EXCEEDED BY 30,000 LINES
09.38.06 JOB36923 -DFHIVPDB CICS PRTBUXT 00 1909 .01 .00 .05 13560 0 0 0 0
0 8
09.38.06 JOB36923 IEF404I DFHIVPDB - ENDED - TIME=09.38.06
09.38.06 JOB36923 -DFHIVPDB ENDED. NAME=CICINST TOTAL CPU TIME= .06 TOTAL ELAPSED TIME= 1.78
09.38.06 JOB36923 $HASP395 DFHIVPDB ENDED

```

Figure 34. Sample job log output from the DFHIVPDB job

- The DFHIVPDB job uses some system initialization parameters included in the DFH\$SIP5 member of the SYSIN data set, to override default system initialization parameters. Further, the DFH\$SIP5 member was edited to specify other system initialization parameters to create the DFHIVPDB job log shown. For information about these extra system initialization parameters used by the IVP jobs, see [“Specifying system initialization parameters for the IVP jobs”](#) on page 200.
- If you want COBOL, C, C++, and PL/I languages, remove the comment marks from the SCEERUN and SCEERUN2 libraries, and increase the memory size for the job.

You see messages similar to those in [Figure 35](#) on page 223 at the end of the MSGUSER section of the job output.

```

DFHLG0302 04/24/2007 09:36:31 CICSIVP1 Journal name DFHLOG has been installed. Journal type: MVS
CICINST.CICSIVP1.DFHLOG.
DFHLG0302 04/24/2007 09:36:31 CICSIVP1 Journal name DFHSHUNT has been installed. Journal type: MVS
CICINST.CICSIVP1.DFHSHUNT.
DFHLG0744 04/24/2007 09:36:31 CICSIVP1 All records in log stream CICINST.CICSIVP1.DFHLOG have been deleted.
DFHLG0744 04/24/2007 09:36:31 CICSIVP1 All records in log stream CICINST.CICSIVP1.DFHSHUNT have been deleted.
DFHDB8116 I 04/24/2007 09:36:32 CICSIVP1 Connection to DBCTL IM7D is proceeding. Startup Table Suffix used is IV.
DFHDB8101 I 04/24/2007 09:36:32 CICSIVP1 Connection to DBCTL IM7D is now complete. Startup Table Suffix used is IV.
DFHZC3441 I 04/24/2007 09:37:55 CICSIVP1 Orderly termination of VTAM sessions requested. ((1) Module name: DFHZSHU)
DFHRM0205 04/24/2007 09:37:58 CICSIVP1 An activity keypoint has been successfully taken.
DFHLG0743 04/24/2007 09:37:58 CICSIVP1 Tail of log stream CICINST.CICSIVP1.DFHLOG deleted at block id
X'00000000000000FDD'.

```

Note: VTAM is now the z/OS Communications Server.

Figure 35. Sample job log output from the end of the MSGUSER section of the DFHIVPDB job

You see messages similar to those in [Figure 36 on page 223](#) in the Printer section of the job output.

```

1DFHDB8210D Connection to DBCTL is proceeding. Check CDBC TD queue.
DFHDB8225I CICSIVP1 The DBCTL ID is IM7D. The DRA Startup Table suffix is IV.

INPUT: ASMCDA02MS16995-28
PART=02MS16995-28      DESC=  SCREW
  AREA  INV  PROJ  DIV  UNIT  CURRENT  ON  IN  TOTAL  COUNT  BACK
    DEPT  CD          PRICE  REQMTS  ORDER  STOCK  DISBURSE  TAKEN  ORDR
1.    AA   165   11    0.152    260      0    300    4030    N    0
2.    BA   165   15    0.069     60      0     80    5000    N    0
3.    FF   554   6D    0.069    440      0    430    5000    N    0
4.    2    59   109   26    6.980    950      0   1000    5000    N    0

INPUT: ASMCDA02JAN1N976B
PART=02JAN1N976B      DESC=  DIODE  CODE-A
  AREA  INV  PROJ  DIV  UNIT  CURRENT  ON  IN  TOTAL  COUNT  BACK
    DEPT  CD          PRICE  REQMTS  ORDER  STOCK  DISBURSE  TAKEN  ORDR
1.    2    55   091   26    0.000    170   2000    170    4710    N    0

```

Figure 36. Sample job log output from the Printer section of the output of the DFHIVPDB job

Testing the CICS Db2 environment

Use Phase 5 of the Db2 installation verification procedure to test the CICS Db2 environment.

To use the Db2 installation verification procedure, and Phase 5 in particular, see the Db2 administration documentation, which gives information about the procedure and describes the steps involved.

Running Db2 jobs DSNTJ5C and DSNTJ5P

To prepare the sample applications to be used in a CICS Db2 environment, run the DSNTJ5C and DSNTJ5P jobs supplied with Db2.

DSNTJ5C installs the sample application transactions in COBOL and prepares the organization application. DSNTJ5P installs the transactions in PL/I and prepares the organization, project, and phone applications.

Both these jobs perform the following functions:

- Compile and link-edit the CICS online applications.
- Bind the CICS online applications.
- Create the BMS maps for the online applications.

Starting a Db2 organization or project application

After logging on to CICS, you can start an organization or project application by entering one of the CICS transaction codes: D8PP, D8PS, or D8CS.

- D8PP starts the PL/I project version

- D8PS starts the PL/I organization version
- D8CS starts the COBOL organization version

If you enter one of these transaction codes, the panels shown in [Figure 37 on page 224](#) or [Figure 38 on page 224](#) are displayed.

```

              ACTION SELECTION
MAJOR SYSTEM ....: 0          ORGANIZATION
ACTION .....:
OBJECT .....:
SEARCH CRITERIA.:
DATA .....:
SELECT AN ACTION FROM FOLLOWING LIST

A      ADD (INSERT)
D      DISPLAY (SHOW)
E      ERASE (REMOVE)
U      UPDATE (CHANGE)

```

Figure 37. Initial panel for the Db2 project application in CICS

```

              ACTION SELECTION
MAJOR SYSTEM ....: P          PROJECTS
ACTION .....:
OBJECT .....:
SEARCH CRITERIA.:
DATA .....:
SELECT AN ACTION FROM FOLLOWING LIST

A      ADD (INSERT)
D      DISPLAY (SHOW)
E      ERASE (REMOVE)
U      UPDATE (CHANGE)

```

Figure 38. Initial panel for the Db2 project application in CICS

For detailed information about running the organization and project applications, see [Overview of the CICS Db2 interface](#).

Starting the Db2 phone application

To start the phone application, clear the screen and type in the transaction code D8PT. You can change the transaction codes when you install Db2. Check with your system administrator to find out if they have been changed from those shown.

Chapter 11. Default CICS resource definitions for CICSplex SM

View the default CICS resource definitions supplied for CICSplex SM to start a CMAS, MAS, and WUI. These are supplied in samples EYU\$CDEF, EYU\$MDEF, and EYU\$WDEF in the SEYUSAMP library.

Default CICS resource definitions for a CMAS

On an INITIAL start of a CMAS, CICSplex SM uses the default CICS resource definitions supplied in sample EYU\$CDEF. [Table 29 on page 225](#) through [Table 33 on page 229](#) list these definitions.

Important: Do not change the recovery options of the EYUDREPN FILE definition. This definition is used when CICSplex SM determines that Data Repository file operations do not require logging. It is usual to receive LSR pool messages for EYUDREPN during CMAS initialization and ignore them. Make sure that the CICS JCL does not have a DD statement for EYUDREPN, and do not associate EYUDREPN with a data set name.

Resource type

Default resource definitions

File

Table 29. Sample File resource definitions supplied in EYU\$CDEF	
Name	Description
EYUDREP	CICSplex SM data repository
EYUDREPN	CICSplex SM data repository, alternate definition

Profile

Table 30. Sample Profile resource definitions supplied in EYU\$CDEF	
Name	Description
EYUCICSD	CICSplex SM debugger

Program

Table 31. Sample Program resource definitions supplied in EYU\$CDEF	
Name	Description
EYUTMMDT	CICSplex SM monitor definition table
EYUTXDDA	CICSplex SM repository record control table
EYUTXLNT	CICSplex SM notification table
EYUTXLPD	CICSplex SM parameter table
EYU9BA00	CICSplex SM BAS object
EYU9CM00	CICSplex SM CMAS communications
EYU9DBG0	CICSplex SM debugger
EYU9DBG1	CICSplex SM debugger
EYU9DBG2	CICSplex SM debugger
EYU9DEBUG	CICSplex SM debugger

<i>Table 31. Sample Program resource definitions supplied in EYU\$CDEF (continued)</i>	
Name	Description
EYU9MN00	CICSplex SM MAS monitor
EYU9NA00	CICSplex SM CMAS MAS component
EYU9NXOP	CICSplex SM CMAS MAS program
EYU9PS00	CICSplex SM CMAS real-time analysis (RTA)
EYU9TS00	CICSplex SM CMAS topology
EYU9WM00	CICSplex SM CMAS workload management
EYU9XC00	CICSplex SM CMAS cache
EYU9XDBU	CICSplex SM CMAS batch repository input/output utility
EYU9XD00	CICSplex SM CMAS data repository
EYU9XLCI	CICSplex SM debugger ATTACH program
EYU9XLCS	CICSplex SM PLTPI startup program
EYU9XLEV	CICSplex SM main initialization
EYU9XLGR	CICSplex SM global user abend exit
EYU9XLME	CICSplex SM message exit
EYU9XLOP	CICSplex SM object environment CREATE
EYU9XLSR	CICSplex SM system user abend exit
EYU9XL00	CICSplex SM CMAS kernel linkage
EYU9XM00	CICSplex SM CMAS message format
EYU9XQ00	CICSplex SM queue manager
EYU9XS00	CICSplex SM common services
EYU9XSTR	CICSplex SM CMAS-MAS task recovery task related user exit
EYU9XZ00	CICSplex SM CMAS trace

Transaction

<i>Table 32. Sample Transaction resource definitions supplied in EYU\$CDEF</i>	
Name	Description
BMLT	CICSplex SM BAS long-running task
CODB	CICSplex SM debugger
COD0	CICSplex SM debugger
COD1	CICSplex SM debugger
COD2	CICSplex SM debugger
COLU	CICSplex SM CICS online utility
COSD	CICSplex SM CMAS stop
LCMU	CICSplex SM Transmit must-complete remote message argument lists
LCPP	CICSplex SM check joining CMAS for plex export

<i>Table 32. Sample Transaction resource definitions supplied in EYU\$CDEF (continued)</i>	
Name	Description
LECI	CICSplex SM CMAS-MAS CPI-C initial contact - CMAS
LECR	CICSplex SM CMAS-MAS CPI-C RECEIVE manager - CMAS
LECS	CICSplex SM CMAS-MAS CPI-C SEND manager - CMAS
LEEI	CICSplex SM ESSS initial contact transient
LEER	CICSplex SM ESSS RECEIVE link manager
LEMI	CICSplex SM CMAS-MAS MRO initial contact - CMAS
LEMS	CICSplex SM CMAS-MAS MRO SEND manager - CMAS
LENS	CICSplex SM CMAS-NetView SEND link manager
LMIR	CICSplex SM MRO RECEIVE link - CMAS
LNCI	CICSplex SM CMAS-CMAS CPI-C initial contact
LNCS	CICSplex SM CMAS-CMAS CPI-C SEND link manager
LNMI	CICSplex SM CMAS-CMAS initial contact task
LNMS	CICSplex SM CMAS-CMAS MRO initial contact task
LPDG	CICSplex SM net direct address generation
LPLK	CICSplex SM network lock
LPLT	CICSplex SM connection long-running task
LPRT	CICSplex SM connection services remove link
LPSC	CICSplex SM repository synchronization check
LPSM	CICSplex SM CMAS - shutdown MAS
LRLT	CICSplex SM security long-running task
LSGT	CICSplex SM garbage collection long-running task
LSRT	CICSplex SM communications long-running task
LWTM	CICSplex SM timing transaction
MCCM	CICSplex SM data collector manager
MCTK	CICSplex SM monitor end-of-task data collector
MMEI	CICSplex SM end-of-interval processor
MMIS	CICSplex SM monitor MAS attach
MMST	CICSplex SM monitor initialization
PEAD	CICSplex SM RTA evaluation asynchronous data processor
PELT	CICSplex SM RTA evaluation long-running task
PMLT	CICSplex SM RTA MRM long-running task
PNLT	CICSplex SM RTA ACT long-running task
PPLT	CICSplex SM RTA analysis point long-running task
PRLT	CICSplex SM RTA ANL long-running task

Table 32. Sample Transaction resource definitions supplied in EYU\$CDEF (continued)

Name	Description
PRPR	CICSplex SM RTA analysis asynchronous data processor
PSLT	CICSplex SM POL long-running transaction
TICT	CICSplex SM topology long-running task map change
TIRT	CICSplex SM topology long-running task RODM
TIST	CICSplex SM topology long-running task services
TSMH	CICSplex SM topology resident map list task
TSPD	CICSplex SM topology CICSplex delete
TSSC	CICSplex SM topology system start event
TSSJ	CICSplex SM topology join event
WMCC	CICSplex SM workload manager (WLM) create work
WMGR	CICSplex SM WLM long running task
WMLA	CICSplex SM WLM update AOR abend lists
WMQB	CICSplex SM query workloads for a CICSplex
WMQM	CICSplex SM manage query workload process
WMQS	CICSplex SM quiescing AOR task
WMSC	CICSplex SM update scope service class tables
WMWC	CICSplex SM WLM create workload
WMWD	CICSplex SM WLM AOR descriptor generation
WMWT	CICSplex SM WLM end workload
WSCL	CICSplex SM global lock service
WSLW	CICSplex SM locking task
XDBM	CICSplex SM batch repository mainline
XDNC	CICSplex SM API notification long-running task
XDND	CICSplex SM API disposition long-running task
XDNE	CICSplex SM API connection long-running task
XDNR	CICSplex SM API command-processing task
XDNS	CICSplex SM API service long-running task
XDSR	CICSplex SM build MPSYNCCR record queue
XLEV	CICSplex SM initialization transaction
XLNX	CICSplex SM notify long-running task
XLST	CICSplex SM selector transaction
XMLT	CICSplex SM consolidated message log long-running task
XQST	CICSplex SM asynchronous spooling of CICSplex SM queue
XZLT	CICSplex SM CMAS-MAS trace processor

Transient data queue

Table 33. Sample Transient data queue resource definitions supplied in EYU\$CDEF	
Name	Description
COLG	CICSplex SM log output
COPR	CICSplex SM parameters

Default CICS resource definitions for a MAS

On an INITIAL start of a MAS, or if a MAS is started by the COLM transaction, CICSplex SM uses the default CICS resource definitions supplied in sample EYU\$MDEF. [Table 34 on page 229](#) through [Table 37 on page 231](#) list these definitions.

Resource type

Default resource definitions

Profile

Table 34. Sample Profile resource definitions supplied in EYU\$MDEF	
Name	Description
EYUCICSD	CICSplex SM debugger transaction

Program

Table 35. Sample Program resource definitions supplied in EYU\$MDEF	
Name	Description
EYUTXLPD	CICSplex SM parameter table
EYU9AEDF	CICSplex SM API EDF formatter
EYU9BA01	CICSplex SM MAS BAS object
EYU9CM01	CICSplex SM CMAS communications
EYU9DBG0	CICSplex SM debugger
EYU9DBG1	CICSplex SM debugger
EYU9DBG2	CICSplex SM debugger
EYU9DBG3	CICSplex SM debugger
EYU9DBG4	CICSplex SM debugger
EYU9DBG5	CICSplex SM debugger
EYU9DBG6	CICSplex SM debugger
EYU9DBG7	CICSplex SM debugger
EYU9DBG8	CICSplex SM debugger
EYU9DBG9	CICSplex SM debugger
EYU9MN01	CICSplex SM MAS monitor
EYU9NA01	CICSplex SM MAS agents
EYU9NLDC	CICSplex SM MAS XDUREQC exit
EYU9NLDR	CICSplex SM MAS XDUREQ exit
EYU9NLID	CICSplex SM MAS XRSINDI exit
EYU9NLME	CICSplex SM MAS XMEOUT exit
EYU9NLSO	CICSplex SM MAS XSNOFF exit
EYU9NMST	CICSplex SM MAS XSTOUT exit
EYU9NMTE	CICSplex SM MAS XMNOUT exit
EYU9NPS2	CICSplex SM MAS shutdown task

<i>Table 35. Sample Program resource definitions supplied in EYU\$MDEF (continued)</i>	
Name	Description
EYU9NXLM	CICSplex SM MAS PLTPI
EYU9NXSD	CICSplex SM MAS global shutdown exit
EYU9NXSH	CICSplex SM MAS global shutdown program
EYU9PS01	CICSplex SM MAS real-time analysis
EYU9TS01	CICSplex SM MAS topology
EYU9WM01	CICSplex SM MAS workload management
EYU9WRAM	CICSplex SM WLM route
EYU9XC01	CICSplex SM MAS cache
EYU9XC02	CICSplex SM MAS cache
EYU9XD01	CICSplex SM MAS data repository
EYU9XLAP	CICSplex SM API object environment CREATE
EYU9XLEV	CICSplex SM main initialization
EYU9XLGR	CICSplex SM global user abend XPCTA exit
EYU9XLOP	CICSplex SM object environment CREATE
EYU9XLSR	CICSplex SM system user abend XSRAB exit
EYU9XL01	CICSplex SM MAS kernel linkage
EYU9XM01	CICSplex SM MAS message format
EYU9XQ01	CICSplex SM MAS queue manager
EYU9XSTR	CICSplex SM CMAS-MAS task recovery task related user exit
EYU9XS01	CICSplex SM MAS common services
EYU9XZ01	CICSplex SM MAS trace

Transaction

<i>Table 36. Sample Transaction resource definitions supplied in EYU\$MDEF</i>	
Name	Description
CODB	CICSplex SM debugger
COD0	CICSplex SM debugger
COD1	CICSplex SM debugger
COD2	CICSplex SM debugger
COIE	CICSplex SM MAS status transaction
COHT	CICSplex SM MAS historical data collection transaction
COIR	CICSplex SM RTA evaluation definition task - MAS
COI0	CICSplex SM MAS communications receive transaction
COLM	CICSplex SM MAS startup
COLU	CICSplex SM CICS online utility

<i>Table 36. Sample Transaction resource definitions supplied in EYU\$MDEF (continued)</i>	
Name	Description
CONA	CICSplex SM MAS alternate long-running task transaction
COND	CICSplex SM MAS shutdown transaction
CONH	CICSplex SM MAS history task
CONL	CICSplex SM MAS initialization transaction
CONM	CICSplex SM MAS monitor task
CORT	CICSplex SM RTA task
COSH	CICSplex SM MAS shutdown
COWC	CICSplex SM MAS workload management garbage collection

Transient data queue

<i>Table 37. Sample Transient data queue resource definitions supplied in EYU\$MDEF</i>	
Name	Description
COPR	CICSplex SM buffers

Default CICS resource definitions for a WUI

On an INITIAL start of a WUI, or if a WUI is started by the COVC transaction, CICSplex SM uses the default CICS resource definitions supplied in sample EYU\$WDEF. [Table 38 on page 231](#) and [Table 43 on page 233](#) list these definitions. As a WUI server is a MAS, CICSplex SM also uses the default CICS resource definitions supplied in sample EYU\$MDEF, and listed in [“Default CICS resource definitions for a MAS” on page 229](#).

Resource type

Default resource definitions

Enqmodel

<i>Table 38. Sample Enqmodel resource definitions supplied in EYU\$WDEF</i>	
Name	Description
EYUWREP	CICSplex SM WUI repository update serialization

File

<i>Table 39. Sample File resource definitions supplied in EYU\$WDEF</i>	
Name	Description
EYUWREP	CICSplex SM WUI data repository

Mapset

<i>Table 40. Sample Mapset resource definitions supplied in EYU\$WDEF</i>	
Name	Description
EYU9VCE	CICSplex SM WUI control mapset

Program

<i>Table 41. Sample Program resource definitions supplied in EYU\$WDEF</i>	
Name	Description
EYU9VKEC	CICSplex SM WUI server
EYU9VKIT	CICSplex SM WUI initialization
EYU9VWAN	CICSplex SM WUI analyzer
EYU9VWCV	CICSplex SM WUI converter
EYUTVOSE	CICSplex SM WUI resource table text strings (US English)
EYUTVTGE	CICSplex SM WUI GIF files (US English)
EYUTVTHE	CICSplex SM WUI HTML pages (US English)
EYUTVTJE	CICSplex SM Java classes (US English)
EYUTVTME	CICSplex SM WUI messages (US English)
EYUTVTSE	CICSplex SM WUI strings (US English)
EYUTVOSK	CICSplex SM WUI resource table text strings (Japanese)
EYUTVTGK	CICSplex SM WUI GIF files (Japanese)
EYUTVTHK	CICSplex SM WUI HTML pages (Japanese)
EYUTVTJK	CICSplex SM Java classes (Japanese)
EYUTVTMK	CICSplex SM WUI messages (Japanese)
EYUTVTSK	CICSplex SM WUI strings (Japanese)
EYUTVOSS	CICSplex SM WUI resource table text strings (Simplified Chinese)
EYUTVTGS	CICSplex SM WUI GIF files (Simplified Chinese)
EYUTVTHS	CICSplex SM WUI HTML pages (Simplified Chinese)
EYUTVTJS	CICSplex SM Java classes (Simplified Chinese)
EYUTVTMS	CICSplex SM WUI messages (Simplified Chinese)
EYUTVTSS	CICSplex SM WUI strings (Simplified Chinese)

Transaction

<i>Table 42. Sample Transaction resource definitions supplied in EYU\$WDEF</i>	
Name	Description
COVA	CICSplex SM WUI user API task
COVC	CICSplex SM WUI server controller
COVE	CICSplex SM WUI HTTP error handler
COVG	CICSplex SM WUI global task
COVP	CICSplex SM WUI proxy
COVU	CICSplex SM WUI resource server

Transient data queue

<i>Table 43. Sample Transient data queue resource definitions supplied in EYU\$WDEF</i>	
Name	Description
COLG	CICSplex SM log output
COVE	CICSplex SM WUI export data set
COVI	CICSplex SM WUI import data set
COVP	CICSplex SM WUI parameters

Notices

This information was developed for products and services offered in the United States of America. This material might be available from IBM in other languages. However, you may be required to own a copy of the product or product version in that language in order to access it.

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 rights 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 grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

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 jurisdictions do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

This information 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.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who want 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 Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119 Armonk,
NY 10504-1785
United States of America*

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 Client Relationship Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

The performance data discussed herein is presented as derived under specific operating conditions. Actual results may vary.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to actual people or business enterprises is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

Programming interface information

IBM CICS supplies some documentation that can be considered to be Programming Interfaces, and some documentation that cannot be considered to be a Programming Interface.

Programming Interfaces that allow the customer to write programs to obtain the services of CICS Transaction Server for z/OS, Version 5 Release 6 (CICS TS 5.6) are included in the following sections of the online product documentation:

- [Developing applications](#)
- [Developing system programs](#)
- [CICS TS security](#)
- [Developing for external interfaces](#)
- [Application development reference](#)
- [Reference: system programming](#)
- [Reference: connectivity](#)

Information that is NOT intended to be used as a Programming Interface of CICS TS 5.6, but that might be misconstrued as Programming Interfaces, is included in the following sections of the online product documentation:

- [Troubleshooting and support](#)
- [CICS TS diagnostics reference](#)

If you access the CICS documentation in manuals in PDF format, Programming Interfaces that allow the customer to write programs to obtain the services of CICS TS 5.6 are included in the following manuals:

- Application Programming Guide and Application Programming Reference
- Business Transaction Services

- Customization Guide
- C++ OO Class Libraries
- Debugging Tools Interfaces Reference
- Distributed Transaction Programming Guide
- External Interfaces Guide
- Front End Programming Interface Guide
- IMS Database Control Guide
- Installation Guide
- Security Guide
- CICS Transactions
- CICSplex System Manager (CICSplex SM) Managing Workloads
- CICSplex SM Managing Resource Usage
- CICSplex SM Application Programming Guide and Application Programming Reference
- Java Applications in CICS

If you access the CICS documentation in manuals in PDF format, information that is NOT intended to be used as a Programming Interface of CICS TS 5.6, but that might be misconstrued as Programming Interfaces, is included in the following manuals:

- Data Areas
- Diagnosis Reference
- Problem Determination Guide
- CICSplex SM Problem Determination Guide

Trademarks

IBM, the IBM logo, and ibm.com[®] are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at [Copyright and trademark information at www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Apache, Apache Axis2, Apache Maven, Apache Ivy, the Apache Software Foundation (ASF) logo, and the ASF feather logo are trademarks of Apache Software Foundation.

Gradle and the Gradlephant logo are registered trademark of Gradle, Inc. and its subsidiaries in the United States and/or other countries.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

The registered trademark Linux[®] is used pursuant to a sublicense from the Linux Foundation, the exclusive licensee of Linus Torvalds, owner of the mark on a worldwide basis.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Red Hat[®], and Hibernate[®] are trademarks or registered trademarks of Red Hat, Inc. or its subsidiaries in the United States and other countries.

Spring Boot is a trademark of Pivotal Software, Inc. in the United States and other countries.

UNIX is a registered trademark of The Open Group in the United States and other countries.
Zowe™, the Zowe logo and the Open Mainframe Project™ are trademarks of The Linux Foundation.
The Stack Exchange name and logos are trademarks of Stack Exchange Inc.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM online privacy statement

IBM Software products, including software as a service solutions, (*Software Offerings*) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information (PII) is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect PII. If this Software Offering uses cookies to collect PII, specific information about this offering's use of cookies is set forth below:

For the CICSplex SM Web User Interface (main interface):

Depending upon the configurations deployed, this Software Offering may use session and persistent cookies that collect each user's user name and other PII for purposes of session management, authentication, enhanced user usability, or other usage tracking or functional purposes. These cookies cannot be disabled.

For the CICSplex SM Web User Interface (data interface):

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user's user name and other PII for purposes of session management, authentication, or other usage tracking or functional purposes. These cookies cannot be disabled.

For the CICSplex SM Web User Interface ("hello world" page):

Depending upon the configurations deployed, this Software Offering may use session cookies that do not collect PII. These cookies cannot be disabled.

For CICS Explorer:

Depending upon the configurations deployed, this Software Offering may use session and persistent preferences that collect each user's user name and password, for purposes of session management, authentication, and single sign-on configuration. These preferences cannot be disabled, although storing a user's password on disk in encrypted form can only be enabled by the user's explicit action to check a check box during sign-on.

If the configurations deployed for this Software Offering provide you, as customer, the ability to collect PII from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see [IBM Privacy Policy](#) and [IBM Online Privacy Statement](#), the section entitled *Cookies, Web Beacons and Other Technologies* and the [IBM Software Products and Software-as-a-Service Privacy Statement](#).

Index

Special Characters

-supplied procedures, copying into a [52](#)

A

ACB (z/OS Communications Server), authorizing the region user ID to open [63](#)

ACBNAME, parameter of SNA APPL statement [90](#)

ACBNAME, VTAM APPL statement parameter [96](#)

ACCEPT, function of SMP/E [50](#), [157](#)

ACCTFILE, primer sample application accounts data set [173](#)

ACIXFILE, primer sample application index data set [173](#)

ACTIVATE, parameter of DFHISTAR [29](#)

activating real-time analysis

 SNA definitions

 for CMAS [98](#)

activation module [17](#), [29](#)

address space (CMAS)

 define SNA requirements

 activate the major nodes [98](#)

 define z/OS Communications Server for SNA requirements

 define APPL statement [96](#)

 define cross-domain resources [96](#)

 define z/OS Communications Server requirements

 define APPL statement [95](#)

 define cross-domain resources [95](#)

 journalmodel [178](#)

 preparing for API [72](#)

 start ESSS [71](#)

 update IEASYSxx [75](#)

ADDTVOL, parameter of DFHISTAR [39](#)

ADFHAPD1 distribution library [32](#)

ADFHAPD2 distribution library [32](#)

ADFHC370 elements [32](#)

ADFHCLIB distribution library [32](#)

ADFHCOB elements [32](#)

ADFHINST distribution library [32](#)

ADFHMAC distribution library [32](#)

ADFHMLIB distribution library [32](#)

ADFHMOD distribution library [32](#)

ADFHMSGs distribution library [32](#)

ADFHMSRC distribution library [32](#)

ADFHPARM distribution library [32](#)

ADFHPL1 elements [32](#)

ADFHPROC distribution library [32](#)

ADFHSAMP distribution library [32](#)

AINDEX, parameter of DFHISTAR [38](#)

ALINDEX, parameter of DFHISTAR [39](#)

allocating

 distribution and target libraries [48](#)

 RELFILE data sets [48](#)

 SMP/E data sets [49](#)

APAR fixes [156](#), [159](#)

APF, MVS initialization parameter

APF, MVS initialization parameter (*continued*)

 IEASYSxx value [74](#)

APPL statement

 CMAS [96](#)

application programming interface (API)

 installing REXX function package [72](#)

 preparing for [72](#)

APPLID

 accessing from several MVS images [91](#)

 and IMS LOGMODE entry [187](#)

 notifying SNA of the CICS APPLID [206](#)

 using for SNA logon [209](#)

APPLY, function of SMP/E [50](#), [157](#)

applying service

 supplied SMP/E procedure [164](#)

applying service to CICS [156](#)

ASMHL exec, to create an MVS nucleus module list for CICS [88](#)

ASMPPTS, parameter of DFHISTAR [39](#)

ASMPSCDS, parameter of DFHISTAR [39](#)

ASMPSTS, parameter of DFHISTAR [39](#)

ASREXIT [124](#)

ATCCONxx, SNA list member [97](#)

ATCSTRxx, SNA list member [97](#)

attributes for SMP/E data sets [34](#), [35](#)

AUTH, parameter of SNA APPL statement [90](#)

AUTH, SNA APPL statement parameter

 CMAS [96](#)

automated-operation programs

 message formatting coexistence [74](#)

auxiliary trace (A) data set, DFHAUXT

 creating for a CICS region [172](#)

 DD statement in IVP startup procedure [202](#)

auxiliary trace (B) data set, DFHBUXT

 creating for a CICS region [172](#)

 DD statement in IVP startup procedure [202](#)

AZONE, parameter of DFHISTAR [39](#)

AZONECSI, parameter of DFHISTAR [39](#)

AZONELOG, parameter of DFHISTAR [39](#)

B

BLKFB80 parameter of the DFHISTAR job [30](#)

BLKU parameter of the DFHISTAR job [30](#)

block sizes for data sets

 block sizes for [30](#)

C

catalogs

 distribution and target libraries [48](#)

category 1 transactions, authorizing the region user ID [64](#)

CBPDO method of installing

 description of [1](#)

CDRSC

 create or access member

CDRSC (continued)

- create or access member (continued)
 - in SYS1.VTAMLST library [97](#)
- defining [96](#)
- CDRSCI, parameter of SNA START command [92](#)
- CDS (control data set), saved [35](#)
- CEDA transaction
 - installing the CICS sample programs [209](#)
- CEECCICS, Language Environment interface module [178](#)
- CEEMSG, transient data destination, Language Environment [180](#)
- CEEOUT, transient data destination, Language Environment [180](#)
- CESE, transient data destination, Language Environment [180](#)
- CESO, transient data destination, Language Environment [180](#)
- checklists for setup
 - CMAS [4](#)
 - MAS [4](#)
 - overview [3](#)
 - WUI [4](#)
- CICS Explorer
 - Installation planning [3](#)
 - Installing [11](#)
- CICS libraries
 - SMP/E function ACCEPT (DFHINST6) [50](#)
- CICS modules required in the LPA
 - DFH99SVC [99](#)
 - DFHCSVC [99](#)
 - DFHDUMPX [99](#)
 - DFHIRP [99](#)
 - DFHSPEX [99](#)
 - DFHSEN [99](#)
 - DFHSSGC [99](#)
 - DFHSSWT [99](#)
- CICS provided transactions in interactive IVP
 - from a terminal [209](#)
- CICS Sockets [182](#)
- CICS system definition data set, DFHCSD
 - creating for all CICS regions [171](#)
 - DD statement in IVP startup procedure [203](#)
 - requirements for autoinstall [205](#)
- CICS system definition file (CSD)
 - updating [176](#)
- CICS TS
 - installing [17](#), [19](#)
 - storage requirements [2](#)
- CICSplex SM installation attributes [43](#)
- class of service (COS)
 - ACF/Communications Server LOGMODE entry [187](#)
- CMAC, CICS messages facility
 - DD statement in IVP startup procedure [202](#)
 - defining resources required [173](#)
 - defining the messages data set, DFHCMACD [173](#)
 - messages data set, DFHCMACD [202](#)
 - resources required to use with the IVP jobs [200](#)
- CMACVOL parameter of the DFHISTAR job [33](#)
- CMAS
 - setup worksheet [6](#)
- CMAS (address space)
 - define SNA requirements
 - activate the major nodes [98](#)
 - define z/OS Communications Server for SNA requirements

CMAS (address space) (continued)

- define z/OS Communications Server for SNA requirements (continued)
 - define APPL statement [96](#)
- define z/OS Communications Server requirements
 - define APPL statement [95](#)
 - define cross-domain resources [95](#)
- preparing for API [72](#)
- start ESSS [71](#)
- update IEASYSxx [75](#)
- CMAS (address space)
 - define SNA requirements [95](#)
 - journalmodel [178](#)
- CMAS setup worksheet [6](#)
- CMASNAME parameter of the DFHISTAR job [43](#)
- CMCIPT parameter of the DFHISTAR job [44](#)
- CMD, MVS initialization parameter
 - IEASYSxx value [74](#)
- CMSSYSID parameter of the DFHISTAR job [44](#)
- COBOL under Language Environment [180](#)
- coexistence
 - automated-operation programs [74](#)
 - releases of CICS before this release [74](#)
- common CICS data sets, creating (DFHCOMDS job) [171](#)
- communications
 - CMAS-to-CMAS [96](#)
- Configuring TS Explorer [11](#)
- console device for the IVP jobs
 - using transactions [209](#)
- CONSOLE macro [129](#)
- console message formatting
 - activating [82](#)
 - default initialization parameters [82](#)
 - defining CICS as an MVS subsystem [73](#)
 - modules required [83](#)
- console messages [129](#)
- console support
 - TSO CLIST processing [209](#)
- consolidated software inventory (CSI) [36](#)
- coupling facility
 - definition [136](#)
 - structures [136](#)
- coupling facility cache structures, VSAM RLS [126](#)
- coupling facility lock structure, VSAM RLS [125](#)
- COUT, transient data extrapartition data set
 - DD statement in IVP startup procedure [203](#)
- cross system locking, VSAM RLS [125](#)
- cross-domain resources (CDRSC)
 - create or access member
 - in SYS1.VTAMLST library [96](#), [97](#)
 - defining [96](#)
- cross-system coupling facility (XCF)
 - for cross-system MRO [183](#)
 - used for interregion communication [182](#)
- cross-system MRO (XCF/MRO)
 - generating support for [184](#)
- CSD files
 - updating [176](#)
- CSI (consolidated software inventory) [36](#)
- CSSLIB parameter of the DFHISTAR job [40](#)
- CSYSNAME parameter of the DFHISTAR job [44](#)
- CSYSPLEX parameter of the DFHISTAR job [44](#)
- CSYSYSID parameter of the DFHISTAR job [44](#)
- customizing CICS

customizing CICS (*continued*)
adding ISC support [185](#)
installing CICS modules in MVS link pack area [98](#)
using HPO support [88](#)

D

data sets
RELFILE, allocating [48](#)
data sets (non-SMP/E), assigning a high-level index [38](#)
data sets for CICS
common to all regions, creating (DFHCOMDS job) [171](#)
unique to each region, creating (DFHDEFDS job) [172](#)
DBCTL
installation verification procedure (IVP) [218](#)
debugging profiles alternate index data set, DFHDPFMP
creating for each CICS region [172](#)
debugging profiles base data set, DFHDPFMB
creating for each CICS region [172](#)
debugging profiles path data set, DFHDPFMP
creating for each CICS region [172](#)
Default CICS user ID, defining to RACF [69](#)
defining a LU for online IVP [204](#)
defining IPIC connections [190](#)
Defining the hlq.SDFHLPA library to your MVS [112](#)
DEVVOL parameter of the DFHISTAR job [31](#)
Developer Trial [17](#)
DFH\$AFLA CSD group, FILEA sample programs [211](#)
DFH\$BTCH, input to the verify batch job [203](#)
DFH\$FILA CSD group, FILEA sample data set [212](#)
DFH\$SIPn members of SYSIN data set, for IVP jobs [200](#)
DFH99SVC, SVC services [99](#)
DFHASMVS, -supplied procedure [53](#)
DFHAUPL, -supplied procedure [53](#)
DFHAUXT data set, auxiliary trace
creating for a CICS region [172](#)
DD statement in IVP startup procedure [202](#)
DFHBMSU, -supplied procedure [53](#)
DFHBXT data set, auxiliary trace
creating for a CICS region [172](#)
DD statement in IVP startup procedure [202](#)
DFHCMACD, messages data set [173](#)
DFHCMACI, job to create and initialize the messages data set [173](#)
DFHCOMDS, job to create common CICS data sets [171](#)
DFHCSD data set, CICS system definition
creating for all CICS regions [171](#)
DD statement in IVP startup procedure [203](#)
requirements for autoinstall [205](#)
DFHCSDUP requirements
DFHCSD [178](#)
SYSIN [178](#)
DFHCSVC, CICS Type 3 SVC [99](#)
DFHCSVC, the CICS type 3 SVC
correct-level check [84](#)
DFHCSVCU [86](#)
installing [84](#)
using for MRO between different CICS releases [87](#)
using more than one version [86](#)
DFHCSVCU utility program [84](#)
DFHCTAIX, sample application alternate index [173](#)
DFHCTCUS, sample application details data set [173](#)
DFHCTHLP, sample application help data set [173](#)
DFHCXRF, transient data extrapartition data set

DFHCXRF, transient data extrapartition data set (*continued*)
DD statement in IVP startup procedure [203](#)
local request queue data set [203](#)
DFHDEFDS, job to create data sets for each region [172](#)
DFHDLPSB TYPE=ENTRY macro (remote DL/I) [193](#)
DFHDMPA data set, dump
creating for a CICS region [172](#)
DD statement in IVP startup procedure [203](#)
DFHDMPB data set, dump
creating for a CICS region [172](#)
DD statement in IVP startup procedure [203](#)
DFHDPFMB debugging profiles alternate index data set
creating for each CICS region [172](#)
DFHDPFMB debugging profiles base data set
creating for each CICS region [172](#)
DFHDPFMB debugging profiles path data set
creating for each CICS region [172](#)
DFHDSPEX, CICS post exit stub [99](#)
DFHDUMPX, CICS dump query exit
LPA (link pack area) [99](#)
DFHEBTAL, -supplied procedure [53](#)
DFHEITAL, -supplied procedure [53](#)
DFHEXTAL, -supplied procedure [53](#)
DFHGCD data set, global catalog
creating for a CICS region [172](#)
DD statement in IVP startup procedure [202](#)
DFHHPSVC, SNA authorized path SVC
loading into the MVS nucleus [88](#)
DFHHTML data set for HTML templates
creating for a CICS region [172](#)
DFHIJVMJ [51](#)
DFHINST1, installation job [48](#)
DFHINST2, installation job [48](#)
DFHINST3, installation job [49](#)
DFHINST4, installation job [49](#)
DFHINST5, installation job [50](#)
DFHINST6, installation job [50](#)
DFHINTRA data set, intapartition transient data
creating for a CICS region [172](#)
DD statement in IVP startup procedure [202](#)
DFHIRP, interregion communication program
MRO requirement [183](#)
DFHISMKD, installation job [48](#)
DFHISTAR, creating postinstallation jobs [168](#)
DFHISTAR, job to define installation-related jobs
BLKFB80 parameter [30](#)
BLKU parameter [30](#)
CMACVOL parameter [33](#)
CMASNAME parameter [43](#)
CMCIPTOR parameter [44](#)
CMSSYSID parameter [44](#)
CSSLIB parameter [40](#)
CSYSNAME parameter [44](#)
CSYSPLX parameter [44](#)
CSYSYSID parameter [44](#)
DEVVOL parameter [31](#)
DINDEX parameter [28](#)
DISTVOL parameter [32](#)
DSINFO parameter [38](#)
DZONE parameter [37](#)
DZONECSI parameter [36](#)
DZONELOG parameter [36](#)
GINDEX parameter [28](#)
GZONE parameter [37](#)

DFHISTAR, job to define installation-related jobs (*continued*)

GZONECSI parameter [36](#)
GZONELOG parameter [35](#)
JAVADIR parameter [30](#)
JES parameter [27](#)
JOB parameter [27](#)
LIB parameter [26](#)
LINDEX parameter [28](#)
LOGGER-INFO parameter [43](#)
NEWDREP-PREFIX parameter [44](#)
OLDDREP parameter [44](#)
OPTVOL parameter [32](#)
parameter default values [22](#)
PREFIX parameter [28](#)
SCEEBND2 dsname [40](#)
SCEECICS dsname [41](#)
SCEECPP parameter [41](#)
SCEELIB dsname [40](#)
SCEELKED parameter [40](#)
SCEELKEX parameter [40](#)
SCEE OBJ parameter [41](#)
SCEERUN dsname [41](#)
SCEERUN2 dsname [41](#)
SCEESAM parameter [42](#)
SCLBSID parameter [41](#)
SCOPE parameter [27](#), [46](#)
SCSFMOD0 dsname [43](#)
SCSQAU parameter [42](#)
SCSQCI parameter [42](#)
SCSQL parameter [42](#)
SCSQLE parameter [42](#)
SEZACMTX dsname [41](#)
SEZARPCL dsname [41](#)
SISPLOAD parameter [40](#)
SIXMEXP dsname [43](#)
SMP/E zone and zone log dispositions [37](#)
SMPLTS parameter [35](#)
SMPMTS parameter [35](#)
SMPPTS parameter [35](#)
SMPSCDS parameter [35](#)
SMPSTS parameter [35](#)
SMPVOL parameter [32](#)
SMPWORK parameter [34](#)
SMS parameter [31](#)
TAPEUNIT parameter [38](#)
TARGVOL parameter [32](#)
TCPIPHST parameter [45](#)
TCPIPPRT parameter [45](#)
TEMPLIB parameter [26](#)
TIMEZONE parameter [45](#)
TINDEX parameter [28](#)
TZONE parameter [37](#)
TZONECSI parameter [36](#)
TZONELOG parameter [35](#)
UTILITIES parameter [27](#)
WORKUNIT parameter [31](#)
WUI parameter [45](#)
WUINAME parameter [45](#)
WUIPLEX parameter [45](#)
WUISYSID parameter [45](#)
DFHIVPBT, batch IVP
DFH\$BTCH input [203](#)
DFHIVPOL, interactive IVP
CICS-supplied transactions [209](#)

DFHIVPOL, interactive IVP (*continued*)

sample programs [209](#)
terminating CICS [213](#)
transactions [209](#)
DFHLCD data set, CICS local catalog
creating for each CICS region [172](#)
DD statement in IVP startup procedure [202](#)
DFHLNKVS, -supplied procedure [53](#)
DFHMAPS, -supplied procedure [53](#)
DFHMAPT, -supplied procedure [53](#)
DFHSMPE, -supplied procedure [53](#)
DFHSMPE, cataloged procedure [158](#)
DFHSSSEN, subsystem interface end-of-memory / end-of-task clean up routine [99](#)
DFHSSGC, connect module [99](#)
DFHSSIyy, message-formatting initialization member [80](#)
DFHSSWT, WTO router [99](#)
DFHSTART, -supplied procedure [53](#)
DFHSTART, CICS startup procedure [201](#)
DFHTEMP data set, temporary storage
creating for each CICS region [172](#)
DD statement in IVP startup procedure [202](#)
DFHYBTPL, -supplied procedure [53](#)
DFHYBTVL, -supplied procedure [54](#)
DFHYITDL, -supplied procedure [54](#)
DFHYITEL, -supplied procedure [54](#)
DFHYITGL, -supplied procedure [54](#)
DFHYITPL, -supplied procedure [54](#)
DFHYITVL, -supplied procedure [54](#)
DFHYXTDL, -supplied procedure [54](#)
DFHYXTEL, -supplied procedure [54](#)
DFHYXTPL, -supplied procedure [54](#)
DFHYXTVL, -supplied procedure [54](#)
DFHZITCL, -supplied procedure [54](#)
DFHZITDL, -supplied procedure [54](#)
DFHZITEL, -supplied procedure [54](#)
DFHZITFL, -supplied procedure [54](#)
DFHZITGL, -supplied procedure [54](#)
DFHZITPL, -supplied procedure [54](#)
DFHZXTCL, -supplied procedure [54](#)
DFHZXTDL, -supplied procedure [54](#)
DFHZXTEL, -supplied procedure [54](#)
DFHZXTPL, -supplied procedure [54](#)
DINDEX parameter of the DFHISTAR job [28](#)
dispositions, zone and zone log [37](#)
distribution libraries
allocating and cataloging [48](#)
SMP/E function ACCEPT (DFHINST6) [50](#)
distribution zone (DZONE)
CSI [36](#)
log [36](#)
DISTVOL parameter of the DFHISTAR job [32](#)
DL/I
adding remote DL/I support [193](#)
database control (DBCTL) [192](#)
defining a PSB directory for remote DL/I support [193](#)
DFHDLPSB TYPE=ENTRY (remote DL/I) [193](#)
function shipping [192](#)
MXSSASZ parameter (remote DL/I) [193](#)
requirements for remote database access [193](#)
system initialization parameters (remote DL/I) [193](#)
DNS [182](#)
DSINFO parameter of the DFHISTAR job [38](#)
dump (A) data set, DFHDMPA

- dump (A) data set, DFHDMPA (*continued*)
 - creating for a CICS region [172](#)
 - DD statement in IVP startup procedure [203](#)
- dump (B) data set, DFHDMPB
 - creating for a CICS region [172](#)
 - DD statement in IVP startup procedure [203](#)
- dynamic LU alias
 - LUAPFX parameter [90](#)
- dynamic parse validation routines, installing [71](#)
- DZONE parameter of the DFHISTAR job [37](#)
- DZONECSI parameter of the DFHISTAR job [36](#)
- DZONELOG parameter of the DFHISTAR job [36](#)

E

- EAS, parameter of SNA APPL statement [90](#)
- ECI [181](#)
- ELPA (extended link pack area)
 - eligible modules [101](#)
- Environment Services System Services (ESSS)
 - adding to MVS linklist [71](#)
 - description of [71](#)
- ENVIRONMENT, EYUIPCSP EXEC parameter
 - IMBED statements required in BLSCECT [121](#)
- ESSS (Environment Services System Services)
 - adding to MVS linklist [71](#)
 - description of [71](#)
- EXCI Pipe Allocation [83](#)
- extended recovery facility (XRF)
 - sessions, HAVAIL, parameter of SNA APPL [90](#)
- EYU9XEES module [71](#)
- EYUCMASJ [169](#)
- EYUCMASP [169](#)
- EYUCMASP, -supplied procedure [55](#)
- EYUCMSOP [169](#)
- EYUCMSDS [169](#)
- EYUCMSSP [169](#)
- EYUCSYDS [169](#)
- EYUCSYSJ [169](#)
- EYUCSYSP [169](#)
- EYUCSYSP, -supplied procedure [55](#)
- EYUJHIST [169](#)
- EYUJWREP [169](#)
- EYULMSOP [169](#)
- EYULMSSP [169](#)
- EYULPMOD [169](#)
- EYUWUIOP [169](#)
- EYUWUIDS [169](#)
- EYUWUIIN [169](#)
- EYUWUIJ [169](#)
- EYUWUIP [169](#)
- EYUWUIP, -supplied procedure [55](#)
- EYUWUISP [169](#)

F

- FILEA data set, sample program data
 - creating for each CICS region [172](#)
 - DD statement in IVP startup procedure [202](#)
- FILEA sample applications
 - transactions used [199](#)
- FORMATMSG, parameter of DFHSSIyy [81](#)
- function shipping [192](#)

G

- generic routecodes [81](#)
- gethostbyaddr function [182](#)
- GID [66](#)
- GIMMPDFT modules, attributes of SMP/E data sets [34](#)
- GIMSP messages and control statements [51](#)
- GIMZPOOL, member of SYS1.MACLIB [49](#)
- GINDEX parameter of the DFHISTAR job [28](#)
- global catalog data set, DFHGCD
 - creating for a CICS region [172](#)
 - DD statement in IVP startup procedure [202](#)
- global zone (GZONE)
 - CSI [36](#)
 - log [35](#)
- group identifier (GID) [66](#)
- GZONE parameter of the DFHISTAR job [37](#)
- GZONECSI parameter of the DFHISTAR job [36](#)
- GZONELOG parameter of the DFHISTAR job [35](#)

H

- hardware requirements
 - installing CICS TS [2](#)
- HAVAIL, parameter of SNA APPL statement [90](#)
- HFS0DSN parameter of the DFHISTAR job [30](#)
- HFS1DSN parameter of the DFHISTAR job [30](#)
- HFSADSN parameter of the DFHISTAR job
 - HFSADSN parameter [30](#)
- HIDEPASSWORD, parameter of DFHSSIyy [81](#)
- high-performance option (HPO)
 - defining the CICS SVCs to MVS [84](#)
 - DFHHPSVC, SNA authorized path SVC [88](#)
 - SNA authorized path, DFHHPSVC [88](#)
 - system initialization parameter [88](#)
- HTML template data set, DFHHTML
 - creating for a CICS region [172](#)
- HTTP [182](#)

I

- ICCFCC, -supplied procedure [55](#)
- ICCFCL, -supplied procedure [55](#)
- ICCFCL, -supplied procedure [55](#)
- ICCFCL, -supplied procedure [55](#)
- IEASYSxx, MVS initialization member
 - establishing requirements [75](#)
 - noting values in [74](#)
 - updating
 - for a CMAS [75](#)
- IEFSSNaa, MVS subsystem initialization member [76](#)
- IEW0461 message from the binder [50](#)
- IEW2646 message from the binder [50](#)
- IEW2651 message from the binder [50](#)
- IMS, database control (DBCTL) [192](#)
- installation
 - ACF/Communications Server definition for CICS
 - LOGMODE entries [187](#)
 - ACF/Communications Server definition for IMS [187](#)
 - ACF/SNA definition for CICS [186](#)

- installation (*continued*)
 - ACF/SNA definition for IMS
 - LOGMODE entries [187](#)
 - of CICS sample programs [209](#)
 - type 3 SVC routine [183](#)
- installation jobs
 - messages [50](#)
 - restarting [47](#)
 - running [46](#)
- installation verification procedure (IVP)
 - applids, defining and activating [198](#)
 - auxiliary temporary storage data set, DFHTEMP [202](#)
 - auxiliary trace data set (A), DFHAUXT [202](#)
 - auxiliary trace data set (B), DFHBUXT [202](#)
 - batch IVP, DFHIVPBT [203](#)
 - CICS region definition data set [203](#)
 - CICS startup procedure, DFHSTART [201](#)
 - CMAC resources [200](#)
 - DFH\$SIP2 member of SYSIN data set [200](#)
 - DFHIVPBT, batch IVP
 - system initialization parameters [200](#)
 - FILEA data set, sample application data [202](#)
 - global catalog data set, DFHGCD [202](#)
 - interactive IVP, DFHIVPOL
 - defining a LU for [204](#)
 - local catalog data set, DFHLCD [202](#)
 - local request queue data set, DFHLRQ [203](#)
 - messages data set, DFHCMACD [202](#)
 - RACF user ID authority [198](#)
 - security [198](#)
 - SYSIN data set [202](#)
 - system definition data set [203](#)
 - system initialization parameters [200](#)
 - transaction dump (A) data set, DFHDMPA [203](#)
 - transaction dump (B) data set, DFHDMPB [203](#)
 - transient data extrapartition data set, COUT [203](#)
 - transient data extrapartition data set, DFHCXRF [203](#)
 - transient data extrapartition data set, LOGUSR [203](#)
 - transient data extrapartition data set, MSGUSR [203](#)
 - transient data intrapartition data set, DFHINTRA [202](#)
 - using the CEDA transaction [209](#)
 - using transactions from console device [209](#)
- installing CICSplex modules in MVS link pack area
 - eligible modules [118](#)
- installing Language Environment support [179](#)
- installing modules in the MVS link pack area [112](#)
- installing support for programming languages [178](#)
- installing system
 - software requirements [2](#)
- interface modules
 - CEECICS [178](#)
 - Language Environment [178](#)
- interregion communication (IRC) [182](#)
- intersystem communication over TCP/IP (IPIC)
 - defining connections [190](#)
- intrapartition transient data data set, DFHINTRA
 - creating for a CICS region [172](#)
 - DD statement in IVP startup procedure [202](#)
- IPCS tools
 - preparing to use
 - updating library allocations [121](#)
- IPIC
 - TCP/IP initialization parameter [190](#)

- ISC [182](#)
- ISMF storage class application [126](#)

J

- Java
 - samples [192](#)
- JAVADIR [51](#)
- JAVADIR parameter of the DFHISTAR job [30](#)
- JES parameter of the DFHISTAR job [27](#)
- JOB parameter of the DFHISTAR job [27](#)
- journalmodels, CMAS [178](#)

L

- Language Environment
 - support for C language [180](#)
 - support for COBOL [180](#)
 - support for PL/I [181](#)
- LIB parameter of the DFHISTAR job [26](#)
- libraries
 - creating multiple sets [55](#)
 - load libraries
 - support for secondary extents [158](#)
 - RACF-protecting [62](#)
 - SMP/E function ACCEPT (DFHINST6) [50](#)
 - target libraries [32](#)
- libraries, CICSplex SM
 - DFHISTAR related [168](#)
 - skeleton jobs [168](#)
- LINDEX parameter of the DFHISTAR job [28](#)
- link editing modules
 - IEW0461 message [50](#)
 - IEW2646 message [50](#)
 - IEW2651 message [50](#)
 - when running installation jobs [50](#)
- link pack area (LPA)
 - CICS modules required in the LPA [99](#)
 - eligible CICSplex modules [118](#)
 - eligible modules [101](#)
 - installing CICS modules in [98](#), [112](#)
 - installing CICSplex modules in [117](#)
 - space for CICS requirements [111](#)
- link-edit temporary store (LTS) [35](#)
- link-editing
 - messages during [50](#)
- linklist, installing CICS-required modules [71](#)
- listener [181](#)
- LNK, MVS initialization parameter
 - IEASYSxx value [74](#)
- LNKAUTH, MVS initialization parameter
 - IEASYSxx value [75](#)
- load libraries
 - secondary extents [158](#)
- local catalog (DFHLCD)
 - creating for each CICS region [172](#)
 - DD statement in IVP startup procedure [202](#)
- log manager
 - coupling facility [136](#)
 - defining logger environment [129](#)
 - requirements planning [130](#)
 - using storage-only logging [150](#)
- log streams, authorizing the region user ID to use [64](#)

- logger environment
 - coupling facility space required for RLS AORs [149](#)
 - RLS implications [149](#)
 - staging data sets [149](#)
- LOGGER-INFO parameter of the DFHISTAR job [43](#)
- LOGMODE entry
 - CICS [187](#)
 - IMS [187](#)
- LOGMODE tables
 - required entries for autoinstall [204](#)
- LOGMODE, parameter of SNA APPL statement [90](#)
- LOGUSR, transient data extrapartition data set
 - DD statement in IVP startup procedure [203](#)
- LPA (link pack area)
 - eligible CICSplex modules [118](#)
 - eligible modules [101](#)
 - installing CICS modules in [98](#), [112](#)
 - installing CICSplex modules in [117](#)
 - space for CICS requirements [111](#)
- LPA, MVS initialization parameter
 - IEASYSxx value [75](#)
- LPA, system initialization parameter [115](#)
- LTS (link-edit temporary store) [35](#)
- LUAPFX, parameter of SNA APPL statement
 - choosing a value [94](#)
- LUs
 - defining for the online IVP [204](#)
- LUTYPE 6 sessions
 - ACQ [90](#)
 - PARSESS [91](#)
 - PERSIST [91](#)

M

- macro temporary store (MTS) [35](#)
- macros
 - DFHDLPSB TYPE=ENTRY (remote DL/I) [193](#)
- MAS
 - setup worksheet [7](#)
- MAS setup worksheet [7](#)
- MAXCAD, MVS initialization parameter
 - IEASYSxx value [75](#)
- MAXPROCUSER [66](#)
- messages
 - GIMSP [51](#)
 - IEW0461, from the binder [50](#)
 - IEW2646, from the binder [50](#)
 - IEW2651, from the binder [50](#)
 - SMP/E [50](#)
 - when running installation jobs [50](#)
- modegroup
 - z/OS Communications Server LOGMODE entries [187](#)
- modeset
 - LU services manager [187](#)
- MODIFY command [209](#)
- modules eligible for the LPA/ELPA [101](#)
- modules, installing in LPA [98](#)
- MRO between different CICS release levels [87](#)
- MSGUSR, transient data extrapartition data set
 - DD statement in IVP startup procedure [203](#)
- MTS (Macro temporary store) [35](#)
- multiple libraries
 - creating [55](#)

- multiregion operation (MRO)
 - between different CICS release levels [87](#)
 - cross-system MRO (XCF/MRO) [183](#)
 - defining CICS as an z/OS subsystem [182](#)
 - installation [182](#)
- MVS ASREXIT [124](#)
- MVS definitions
 - for CICS performance [122](#)
 - for console messages [129](#)
- MVS linklist
 - installing CICS-required modules [71](#)
 - installing IGWABWO [73](#)
 - installing IGWARLS [73](#)
 - other MVS product modules [73](#)
- MVS logger [129](#)
- MVS subsystem, defining CICS as [73](#)
- MVS workload management [122](#)

N

- NEWDREP-PREFIX parameter of the DFHISTAR job [44](#)
- NQNMODE, parameter of SNA START command [92](#)
- NSINTERADDR [182](#)
- NSYSLX, MVS initialization parameter
 - IEASYSxx value [75](#)

O

- OLDDREP parameter of the DFHISTAR job [44](#)
- OPTVOL parameter of the DFHISTAR job [32](#)

P

- PARSESS, parameter of SNA APPL statement [91](#)
- PASS option of SNA APPL parameter AUTH [96](#)
- PDIR (PSB directory) [192](#)
- PDSE [2](#)
- performance definitions for MVS [122](#)
- performance parameters (CICS), matching to service policies [122](#)
- PERSIST, parameter of SNA APPL statement [91](#)
- postinstallation member [168](#)
- postinstallation members
 - EYUCMASJ [169](#)
 - EYUCMASP [169](#)
 - EYUCMSOP [169](#)
 - EYUCMSDS [169](#)
 - EYUCMSSP [169](#)
 - EYUCSYDS [169](#)
 - EYUCSYSJ [169](#)
 - EYUCSYSP [169](#)
 - EYUJHIST [169](#)
 - EYUJWREP [169](#)
 - EYULMSOP [169](#)
 - EYULMSSP [169](#)
 - EYULPMOD [169](#)
 - EYUWUIOP [169](#)
 - EYUWUIDS [169](#)
 - EYUWUIIN [169](#)
 - EYUWUIJ [169](#)
 - EYUWUIP [169](#)
 - EYUWUISP [169](#)
- predefined LU alias [93](#)

- PREFIX parameter of the DFHISTAR job [28](#)
- procedures, copying into a procedure library [52](#)
- PROG, MVS initialization parameter
 - IEASYSxx value [75](#)
- program temporary fix (PTF) [159](#)
- protecting CICS load libraries [62](#)
- PRTAUXT step of IVP jobs [201](#)
- PRTBUXT step of IVP jobs [201](#)
- PRTDMPA step of IVP jobs [201](#)
- PRTDMPB step of IVP jobs [201](#)
- PRVMOD, system initialization parameter [115](#)
- PTF (program temporary fix) [156](#), [159](#)
- PTF temporary store (PTS) [35](#)
- PTS (PTF temporary store) [35](#)

R

- RACF (resource access control facility)
 - activating resource classes [69](#)
 - authorizing the region user ID for category 1 transactions [64](#)
 - authorizing the region user ID to open a region's z/OS Communications Server ACB [63](#)
 - authorizing the region user ID to use MVS log streams [64](#)
 - defining the default CICS user ID [69](#)
 - installing the interface routines [71](#)
 - IVP user ID authority [198](#)
 - LOGSTRM general resource class [64](#)
 - protecting CICS load libraries [62](#)
- RECEIVE, function of SMP/E [157](#)
- references, unresolved [50](#), [51](#)
- region user ID, authorizing for category 1 transactions [64](#)
- region user ID, authorizing to OPEN a region's ACB [63](#)
- RELFILE data sets
 - allocating [48](#)
- requirements for CICS TS
 - hardware [2](#)
 - software [2](#)
- resource classes, activating [69](#)
- resource definitions
 - dynamic creation during CICSplex SM initialization [176](#)
- RESTORE, function of SMP/E [157](#)
- REXX function package, installing [72](#)
- routecodes, generic [81](#)
- ROUTECD, parameter of DFHSSIyy [81](#)
- RSVNONR, MVS initialization parameter
 - IEASYSxx value [75](#)
- RSVSTRT, MVS initialization parameter
 - IEASYSxx value [75](#)

S

- sample applications data sets, creating
 - ACCTFILE, primer sample application accounts [173](#)
 - ACIXFILE, primer sample application index [173](#)
 - DFHCTAIX, sample application alternate index [173](#)
 - DFHCTCUS, sample application details [173](#)
 - DFHCTHLP, sample application help [173](#)
- sample JCL
 - to modify CICSplex SM supplied CICS resource definitions [177](#)
- sample program data, FILEA

- sample program data, FILEA (*continued*)
 - creating for each CICS region [172](#)
 - DD statement in IVP startup procedure [202](#)
- sample programs in interactive IVP [209](#)
- sample programs, installing [209](#)
- saved control data set (SCDS) [35](#)
- SCDS (saved control data set) [35](#)
- SCEEBND2 parameter of the DFHISTAR job [40](#)
- SCEECICS parameter of the DFHISTAR job [41](#)
- SCEECPP parameter of the DFHISTAR job [41](#)
- SCEELIB parameter of the DFHISTAR job [40](#)
- SCEELKED parameter of the DFHISTAR job [40](#)
- SCEELKEX parameter of the DFHISTAR job [40](#)
- SCEEOBJ parameter of the DFHISTAR job [41](#)
- SCEERUN parameter of the DFHISTAR job [41](#)
- SCEERUN2 parameter of the DFHISTAR job [41](#)
- SCEESAMP parameter of the DFHISTAR job [42](#)
- SCLBSID parameter of the DFHISTAR job [41](#)
- SCOPE parameter of the DFHISTAR job [27](#), [46](#)
- SCSFMOD0 parameter of the DFHISTAR job [43](#)
- SCSQANLE parameter of the DFHISTAR job [42](#)
- SCSQAUTH parameter of the DFHISTAR job [42](#)
- SCSQCICS parameter of the DFHISTAR job [42](#)
- SCSQLOAD parameter of the DFHISTAR job [42](#)
- SDFHENV [51](#)
- SDFHPL1 target library [32](#)
- SDSNLOAD parameter
 - SDSNLOAD parameter [42](#)
- SDSNLOAD parameter of the DFHISTAR job [42](#)
- SDUMP options [121](#)
- secondary extents, load libraries [158](#)
- security with the CICS IVPs [198](#)
- service, applying
 - supplied SMP/E procedure [164](#)
- servicing CICS [156](#)
- session outage notification (SON)
 - SONSCIP [91](#)
- setting up
 - CMAS [176](#)
- SEZACMTX parameter of the DFHISTAR job [41](#)
- SEZARPCL parameter of the DFHISTAR job [41](#)
- sharing control data sets, VSAM RLS [127](#)
- SISPLoad parameter of the DFHISTAR job [40](#)
- SIXMEXP parameter of the DFHISTAR job [43](#)
- SMF, MVS initialization parameter
 - IEASYSxx value [75](#)
- SMP/E data sets, attributes [34](#), [35](#)
- SMP/E options, specifying [37](#)
- SMP/E zone and zone log dispositions [37](#)
- SMP/E, using to apply service [164](#)
- SMP0001, SMP/E sample USERMOD for SMP/E data sets [34](#)
- SMPCNTL, DD statement of DFHSMPE [158](#)
- SMPLTS parameter of the DFHISTAR job [35](#)
- SMPMTS parameter of the DFHISTAR job [35](#)
- SMPPTS parameter of the DFHISTAR job [35](#)
- SMPSCDS parameter of the DFHISTAR job [35](#)
- SMPSTS parameter of the DFHISTAR job [35](#)
- SMPVOL parameter of the DFHISTAR job [32](#)
- SMPWORK parameter of the DFHISTAR job [34](#)
- SMS parameter of the DFHISTAR job [31](#)
- SNA
 - APPL statement
 - ACBNAME parameter [90](#)
 - AUTH parameter [90](#)

SNA (*continued*)

APPL statement (*continued*)

- cross-domain services [91](#)
- EAS parameter [90](#)
- HAVAIL parameter [90](#)
- LOGMODE parameter [90](#)
- LUAPFX parameter [90](#)
- PARSESS parameter [91](#)
- PERSIST parameter [91](#)
- SONSCIP parameter [91](#)
- VPACING parameter [91](#)

authorized path [88](#)

CICS APPLID [90](#)

- cross-domain services [91](#)
- definitions required for CICS [90](#)
- LOGMODE table entries for autoinstall [204](#)

SNA requirements

activating definitions [98](#)

software requirements

installing system [2](#)

SONSCIP, parameter of SNA APPL statement [91](#)

source temporary store (STS) [35](#)

SSL [182](#)

staging data sets [149](#)

storage key definition [122](#)

storage requirements

for CICS TS [2](#)

storage-only logging [150](#)

STS (source temporary store) [35](#)

subsystem definition to z/OS [182](#)

subsystem, MVS, defining CICS as [73](#)

SVC

correct-level check for DFHCSVC [84](#)

DFHHPSVC, loading into the MVS nucleus [88](#)

installing the CICS type 3 SVC [84](#)

using DFHCSVC for MRO between different CICS releases [87](#)

using more than one version of DFHCSVC [86](#)

SVCs, installing DFHCSVC in the LPA [99](#)

SYMREC Authorization Exit [124](#)

SYMREC macro [124](#)

SYS1.CPSM320.SEYULPA

defining to your MVS [118](#), [119](#)

SYS1.PARMLIB library

IEASYSxx member

establishing requirements [75](#)

when creating CMAS started tasks [75](#)

MAXCAD [75](#)

SYS1.VTAMLST library

application definitions [96](#)

SYSIN data set, partitioned

creating for all CICS regions [171](#)

DD statement in IVP startup procedure [202](#)

SYSMOD (system modification) [156](#), [164](#)

sysplex, MVS

requirements for cross-system MRO [183](#)

sysplex, z/OS

cross-system coupling facility (XCF)

for MRO links across z/OS images

[182](#)

SYSTCPD [182](#)

system abend [722](#) [50](#)

system initialization parameters

TCPIP [190](#)

system modification (SYSMOD) [164](#)

System Modification Program Extended (SMP/E)

ACCEPT function [50](#), [157](#)

APPLY function [50](#), [157](#)

DFHSMPE [158](#)

messages [51](#)

messages during link-editing [50](#)

RECEIVE function [157](#)

RESTORE function [157](#)

T

tailoring CICS

installing MRO support [182](#)

TAPEUNIT parameter of the DFHISTAR job [38](#)

target libraries

allocating and cataloging [48](#)

SMP/E function ACCEPT (DFHINST6) [50](#)

target zone (TZONE)

CSI [36](#)

log [35](#)

TARGVOL parameter of the DFHISTAR job [32](#)

TCP/IP

DNS [182](#)

ECI [181](#)

enablement [181](#)

HTTP [182](#)

installation requirements [182](#)

ISC [182](#)

listener [181](#)

SSL [182](#)

TCPIP, system initialization parameter [190](#)

TCPIPST parameter of the DFHISTAR job [45](#)

TCPIPRT parameter of the DFHISTAR job [45](#)

TEMPLIB parameter of the DFHISTAR job [26](#)

temporary storage data set, DFHTEMP

creating for each CICS region [172](#)

DD statement in IVP startup procedure [202](#)

TIMEZONE parameter of the DFHISTAR job [45](#)

TINDEX parameter of the DFHISTAR job [28](#)

transaction dump data sets

dump (A) data set, DFHDMPA [203](#)

dump (B) data set, DFHDMPB [203](#)

transient data extrapartition data sets

destination before transient data initialization, DFHCXRF [203](#)

destination before transient data initialization, DFHLRQ [203](#)

destination used by C/370 application programs [203](#)

destination used by CICS services [203](#)

destination, LOGA, used by CICS sample programs [203](#)

type 3 SVC routine

in LPA [183](#)

used for interregion communication [182](#)

TZONE parameter of the DFHISTAR job [37](#)

TZONECSI parameter of the DFHISTAR job [36](#)

TZONELOG parameter of the DFHISTAR job [35](#)

U

UID [66](#)

UNIX System Services access

MAXPROCUSER [66](#)

unresolved references [50](#), [51](#)
updating CICSplex definition
 BLSCECT [121](#)
 CSD files using DFHCSDUP [176](#)
user identifier (UID) [66](#)
USSDIR [51](#)
USSDIRA, parameter of DFHISTAR [29](#)
UTILITIES parameter of the DFHISTAR job [27](#)

V

Value Unit Edition [17](#)
VARY NET command [98](#)
verification tests, running IVP jobs [197](#)
verify batch job, DFHIVPBT
 DFH\$BTCH input [203](#)
verify interactive job, DFHIVPOL
 invoking and running sample programs [209](#)
 terminating CICS [213](#)
 using transactions [209](#)
verify interactive job, DFHIVPOLSNA LU Verify
 using CICS-supplied terminal transactions [209](#)
verify interactive job, DFHIVPOLVTAM
 DFHIVPOL [209](#)
VPACE option of SNA APPL parameter AUTH [90](#)
VPACING, parameter of SNA APPL statement [91](#)
VSAM RLS
 activating the coupling facility structures [129](#)
 adding new parameters to SYS1.PARMLIB [128](#)
 amending parameters in SYS1.PARMLIB(BPXPRMxx) [77](#)
 authorizing CICS user IDs [128](#)
 CFRM policy [126](#)
 defining coupling facility cache structures [126](#)
 defining sharing control data sets [127](#)
 defining SMS storage classes for RLS access [126](#)
 defining the coupling facility lock structure [125](#)
 definitions, required [125](#)
 establishing new procedures [128](#)
 ISMF storage class application [126](#)
 IXCMIAPU utility [126](#)
VTAM LU alias [92](#)

W

Web User Interface Server
 setup worksheet [8](#)
workload management, MVS [122](#)
worksheet for IVP
 CMAS [9](#)
 MAS [9](#)
 WUI [9](#)
worksheets for setup
 CMAS [4](#)
 MAS [4](#)
 overview [3](#)
 WUI [4](#)
WORKUNIT parameter of the DFHISTAR job [31](#)
WTO (write-to-operator) macro [129](#)
WUI parameter of the DFHISTAR job [45](#)
WUI setup worksheet [8](#)
WUINAME parameter of the DFHISTAR job [45](#)
WUIPLEX parameter of the DFHISTAR job [45](#)
WUISYSID parameter of the DFHISTAR job [45](#)

X

XCF (cross-system coupling facility)
 for cross-system MRO [183](#)
XCF/MRO (cross-system MRO)
 generating support for [184](#)
XRF (extended recovery facility)
 sessions, HAVAIL, parameter of SNA APPL [90](#)
XTRAQUAL parameter
 XTRAQUAL parameter [29](#)
XTRAQUAL parameter of the DFHISTAR job [29](#)
XXRSTAT exit, effect on takeover of an SNA ACB [63](#)

Z

z/OS Communication Server
 version, release, and modification level indicator [91](#)
z/OS Communications
 Server
 LOGMODE entries [187](#)
 modegroups [187](#)
z/OS Communications Server ACB, authorizing the region
 user ID to open [63](#)
z/OS Communications Server for SNA requirements [96](#)
z/OS image
 XCF/MRO is required for links between CICS regions in
 different [182](#)
z/OS UNIX [2](#)
z/OS UNIX access [66](#)
z/OS UNIX System Services access [66](#)
zone and zone log dispositions [37](#)

