CICS Transaction Server for z/OS



System Initialization Parameter Reference

Version 5 Release 4

CICS Transaction Server for z/OS



System Initialization Parameter Reference

Version 5 Release 4

Note
Before using this information and the product it supports, read the information in "Notices" on page 137.

This edition applies to the IBM CICS Transaction Server for z/OS Version 5 Release 4 (product number 5655-Y04) and to all subsequent releases and modifications until otherwise indicated in new editions.

© Copyright IBM Corporation 1989, 2017.
US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

About this PDF vii

The system initialization parameter descriptions and summary 1

ADI 12
AIBRIDGE 12
AICONS 12
AIEXIT 13
AILDELAY 13
AIQMAX 14
AIRDELAY 14
AKPFREQ 15
APPLID 15
AUTCONN 17
AUTODST 17
AUTORESETTIME 17
AUXTR 18
AUXTRSW 18
BMS 18
BRMAXKEEPTIME 20
CDSASZE 20
CHKSTRM 21
CHKSTSK 21
CICSSVC 21
CILOCK 22
CLINTCP 22
CLSDSTP 22
CLT 23
CMDPROT 23
CMDSEC 24
CONFDATA 24
CONFTXT 26
CPSMCONN 26
CRLPROFILE 28
CSDACC 28
CSDBKUP 29
CSDBUFND 29
CSDBUFNI 29
CSDDISP 30
CSDDSN 30
CSDFRLOG 30
CSDINTEG 31
CSDJID 32
CSDLSRNO 32
CSDRECOV 32
CSDRLS 33
CSDSTRNO 34
CWAKEY 34
DAE 35
DATFORM 35
DB2CONN 35
DBCTLCON 36
DEBUGTOOL 36
DFLTUSER 36
DIP 37
DISMACP 37
DOCCODEPAGE 37
DSALIM 38
DSHIPIDL 38
DSHIPINT 39
DSRTPGM 39
DTRPGM 40
DTRTRAN 40
DUMP 41
DUMPDS 41
DUMPSW 41
DURETRY 42
ECDSASZE 42
EDSALIM 43
ENCRYPTION (DEPRECATED) 44
EODI 44
ERDSASZE 45
ESDSASZE 45
ESMEXITS 46
EUDSASZE 46
FCT 46
FCQRONLY 47
FEPI 47
FLDSEP 48
FLDSTRT 48
FORCEQR 49
FSSTAFF 49
FTIMEOUT 50
GMTEXT 51
GMTRAN 51
GNTRAN 52
GRNAME 53
GRPLIST 54
GTFTR 55
HPO 55
HTTPSERVERHDR 56
HTTPUSRAGENTHDR 56
ICP 56
ICV 56
ICVR 57
ICVTSD 58
INFOCENTER 58
INITPARM 58
INTTR 59
IRCSTRT 59
ISC 59
JESDI 59
JVMPROFILEDIR 60
KEYRING 60
LGDFINT 60
LGNMSG 61
LLACOPY 61
LOCALCCSID 62
LPA 62
MAXOPENTCBS 63
MAXSOCKETS 63

© Copyright IBM Corp. 1989, 2017

MAXSSLTCBS 64
MAXXPTCBS 64
MCT 64
MINTLSLEVEL 65
MN 65
MNCONV 66
MNEXC 66
MNFREQ 67
MNIDN 67
MNPER 67
MNRES 67
MNSYNC 68
MNTIME 68
MQCONN 68
MROBTCH 69
MROFSE 69
MROLRM 70
MSGCASE 70
MSGLVL 70
MXT 70
NATLANG 71
NCPLDFT 72
NEWSIT 72
NISTSP800131A 73
NONRLSRECOV 74
NQRNL 75
OFFSITE 75
OPERTIM 76
OPNDLIM 76
PARMERR 76
PDI 77
PDIR 77
PGAICTLG 77
PGAIEXIT 78
PGAIPGM 78
PGCHAIN 78
PGCOPY 78
PGPURGE 79
PGRET 79
PLTPI 79
PLTPISEC 80
PLTPIUSR 80
PLTSD 80
PRGDLAY 81
PRINT 81
PRTYAGE 82
PRVMOD 83
PSBCHK 83
PSDINT 83
PSTYPE 84
PVDELAY 85
QUIESTIM 85
RACFSYNC 85
RAMAX 86
RAPOOL 86
RDSASZE 87
RENTPGM 87
RESP 88
RESSEC 88
RLS 88
RLSTOLSR 89
RMTRAN 90
RRMS 90
RST 90
RSTSIGNOFF 91
RSTSIGNTIME 91
RUWAPOOL 92
SDSASZE 93
SDTRAN 93
SEC 94
SECPRFX 94
SIT 95
SKRxxxx 95
SNPRESET 96
SNSCOPE 96
SOTUNING 97
SPCTR 98
SPCTRxx 98
SPOOL 101
SRBSVC 101
SRT 101
SRVERCP 101
SSLCACHE 102
SSLDELAY 102
START 102
STARTER 104
STATEOD 104
STATINT 105
STATRCD 105
STGPROT 106
STGRCVY 106
STNTR 107
STNTRxx 107
SUBTSKS 110
SUFFIX 111
SYDUMAX 111
SYSIDNT 111
SYSTR 111
TAKEOVR 112
TBEXITS 113
TCP 113
TCPIP 114
TCSACTN 114
TCSWAIT 115
TCT 115
TCTUAKEY 116
TCTUALOC 116
TD 116
TDINTRA 117
TRANISO 117
TRAP 118
TRDUMAX 118
TRTABSZ 119
TRTRANSZ 119
TRTRANTY 120
TS 120
TSMAINLIMIT 120
TST 121
UDSASZE 121
UOWNETQL 122
USERTR 122
USRDELAY 123

USSCONFIG 124
USSHOME 124
VTAM (z/OS Communications Server) 124
VTPREFIX 124
WEBDELAY 125
| WLMHEALTH 126
WRKAREA 126
XAPPC 127
XCFGROUP 127
XCMD 128
XDB2 128
XDCT 129
XFCT 129
XHFS 130
XJCT 130
XLT 131
XPCT 131
XPPT 132
XPSB 133
| XPTKT 133
XRES 134
XRF 135
XTRAN 135
XTST 135
XUSER 136

Notices 137

Index 143

About this PDF

This PDF is a reference of the CICS system initialization parameters that are used to modify CICS system attributes when you start a CICS region. This PDF is a companion to the information in *Resource Reference*. Before CICS TS 5.4, the information in this PDF was in the *System Definition Guide*.

The primary method of providing system initialization parameters is in a system initialization table (SIT). As a result, "SIT parameter" is widely used to refer to the system initialization parameters.

For details of the terms and notation used in this book, see Conventions and terminology used in the CICS documentation in IBM Knowledge Center.

Date of this PDF

This PDF was created on June 6th 2017.

© Copyright IBM Corp. 1989, 2017

The system initialization parameter descriptions and summary

You can define system initialization parameters to CICS® in a number of ways. All the system initialization parameters are listed, together with how they can be specified to CICS and their default values.

You can specify system initialization parameters in any of the following ways:

- In a DFHSIT macro.
- In a **PARM** parameter on the EXEC PGM=DFHSIP statement.
- In the SYSIN data set of the CICS startup job stream.
- Through the system console. For parameters that require mixed case values, type / on an SDSF command line to open the System Command Extension facility. Use quotation marks around the parameter and value that you want to specify to preserve the mixed case. The command is still echoed in uppercase in the job log, but SDSF processes the value in mixed case.

The following table lists the parameters and their default values, and whether the parameter can be specified in a **PARM** parameter, in the SYSIN data set, through the system console, or in a DFHSIT macro.

For a summary of changes by release to SIT parameters, see Changes to SIT parameters.

Table 1. System initialization parameters with override options and default settings

Parameter	PARM	SYSIN	System console	DFHSIT	Default value	Description
ADI	YES	YES	YES	YES	30	XRF(B) - Alternate delay interval
AIBRIDGE	YES	YES	YES	YES	AUTO	Bridge Autoinstall URM
AICONS	YES	YES	YES	YES	NO	No autoinstall for MVS™ CONSOLES
AIEXIT	YES	YES	YES	YES	DFHZATDX	Autoinstall user program name
AILDELAY	YES	YES	YES	YES	0	Delete delay period for AI terminals
AIQMAX	YES	YES	YES	YES	100	Maximum number of terminals queued for AI
AIRDELAY	YES	YES	YES	YES	700	Restart delay period for AI terminals
AKPFREQ	YES	YES	YES	YES	4000	Activity keypoint frequency
APPLID	YES	YES	YES	YES	DBDCCICS	z/OS® Communications Server APPL identifier
AUTCONN	YES	YES	YES	YES	0	Autoconnect delay
AUTODST	YES	YES	YES	YES	NO	Language Environment® automatic storage tuning
AUTORESETTIME	YES	YES	YES	YES	IMMEDIATE	Time-of-day synchronization

© Copyright IBM Corp. 1989, 2017

Table 1. System initialization parameters with override options and default settings (continued)

Parameter	PARM	SYSIN	System console	DFHSIT	Default value	Description
AUXTR	YES	YES	YES	YES	OFF	Auxiliary trace option
AUXTRSW	YES	YES	YES	YES	NO	Auxiliary trace autoswitch facility
BMS	YES	YES	YES	YES	FULL, UNALIGN, DDS	Basic Mapping Support options
BRMAXKEEPTIME	YES	YES	YES	YES	86400	Bridge Max Keeptime
CDSASZE	YES	YES	YES	NO	0	The size of CDSA
CHKSTRM	YES	YES	YES	NO	NONE	Activation of terminal storage-violation checking
CHKSTSK	YES	YES	YES	NO	NONE	Activation of task storage-violation checking
CICSSVC	YES	YES	YES	YES	216	The CICS SVC number
CILOCK	YES	YES	YES	YES	NO	Do not keep CI lock after read update
CLINTCP	YES	YES	YES	YES	437	Default client code page
CLSDSTP	YES	YES	YES	YES	NOTIFY	Notification for ISSUE PASS command
CLT	YES	YES	YES	YES	No default	The command list table option or suffix
CMDPROT	YES	YES	YES	YES	YES	Exec storage command checking
CMDSEC	YES	YES	NO	YES	ASIS	API command security checking
CONFDATA	YES	YES	NO	YES	SHOW	Show confidential data in dump and trace
CONFTXT	YES	YES	NO	YES	NO	Do not prevent z/OS Communications Server tracing user data
CPSMCONN	YES	YES	YES	YES	NO	Do not connect to CPSM
CRLPROFILE	YES	YES	YES	YES	No default	Name of profile that allows CICS to access certificate revocation lists
CSDACC	YES	YES	YES	YES	READWRITE	CSD access
CSDBKUP	YES	YES	YES	YES	STATIC	Backup type of CSD (STATIC or DYNAMIC)
CSDBUFND	YES	YES	YES	YES	No default	Number of data buffers for the CSD
CSDBUFNI	YES	YES	YES	YES	No default	Number of index buffers for the CSD
CSDDISP	YES	YES	YES	YES	No default	CSD disposition for dynamic allocation
CSDDSN	YES	YES	YES	YES	No default	CSD data set name for dynamic allocation
CSDFRLOG	YES	YES	YES	YES	NO	Journal ID for CSD forward recovery

Table 1. System initialization parameters with override options and default settings (continued)

Parameter	PARM	SYSIN	System console	DFHSIT	Default value	Description
CSDINTEG	YES	YES	YES	YES	UNCOMMITTED	Read integrity = uncommitted
CSDJID	YES	YES	YES	YES	NO	Journal ID for CSD auto journaling
CSDLSRNO	YES	YES	YES	YES	1	The VSAM LSR pool number for the CSD
CSDRECOV	YES	YES	YES	YES	NONE	CSD recoverable file option
CSDRLS	YES	YES	YES	YES	NO	Use traditional VSAM
CSDSTRNO	YES	YES	YES	YES	6	CSD number of strings
CWAKEY	YES	YES	YES	YES	USER	CWA storage key
DAE	YES	YES	YES	YES	NO	SDUMPS are not suppressed by DAE
DATFORM	YES	YES	YES	YES	MMDDYY	CSA date format
DB2CONN	YES	YES	YES	YES	NO	Do not connect to DB2® at CICS startup
DBCTLCON	YES	YES	YES	YES	NO	Do not connect to DBCTL at CICS start
DEBUGTOOL	YES	YES	YES	YES	NO	No Debugging Tool access
DFLTUSER	YES	YES	NO	YES	CICSUSER	Default user
DIP	YES	YES	YES	YES	NO	Batch data interchange program
DISMACP	YES	YES	YES	YES	YES	Disable macro programs
DOCCODEPAGE	YES	YES	YES	YES	037	Default host code page
DSALIM	YES	YES	YES	YES	5M	Upper limit of DSA below 16 MB line
DSHIPIDL	YES	YES	YES	YES	020000	Delete shipped idle time
DSHIPINT	YES	YES	YES	YES	120000	Delete shipped interval
DSRTPGM	YES	YES	YES	YES	NONE	Distributed routing program
DTRPGM	YES	YES	YES	YES	DFHDYP	Dynamic routing program
DTRTRAN	YES	YES	YES	YES	CRTX	Default dynamic transaction routing transid
DUMP	YES	YES	YES	YES	YES	Dump option
DUMPDS	YES	YES	YES	YES	AUTO	CICS dump data set opening option
DUMPSW	YES	YES	YES	YES	NO	Dump data set autoswitch option
DURETRY	YES	YES	YES	YES	30	SDUMP total retry time (in seconds)
ECDSASZE	YES	YES	YES	NO	0	Size of the ECDSA
EDSALIM	YES	YES	YES	YES	800M	Upper limit of DSA in 31-bit storage
EODI	YES	YES	YES	YES	E0	End-of-data indicator for sequential devices

Table 1. System initialization parameters with override options and default settings (continued)

Parameter	PARM	SYSIN	System console	DFHSIT	Default value	Description
ERDSASZE	YES	YES	YES	NO	0	Size of the ERDSA
ESDSASZE	YES	YES	YES	NO	0	Size of the ESDSA
ESMEXITS	NO	NO	NO	YES	NOINSTLN	External security manager exits
EUDSASZE	YES	YES	YES	NO	0	Size of the EUDSA
FCT	YES	YES	YES	YES	NO	File control table option or suffix
FCQRONLY	YES	YES	YES	YES	YES	Threadsafe FC runs on QR TCB
FEPI	YES	YES	YES	YES	NO	Front-End Programming Interface
FLDSEP	YES	YES	YES	YES	' ' (4 blanks)	End-of-field separator characters
FLDSTRT	YES	YES	YES	YES	' ' (1 blank)	Field start character for built-in function
FORCEQR	YES	YES	YES	YES	NO	Do not force QR for threadsafe programs
FSSTAFF	YES	YES	YES	YES	NO	Function-shipped START affinity option
FTIMEOUT	YES	YES	YES	YES	30	File timeout 30 seconds
GMTEXT	YES	YES	YES	YES	'WELCOME TO CICS'	Good-morning message text
GMTRAN	YES	YES	YES	YES	CSGM	Initial transaction
GNTRAN	YES	YES	YES	YES	NO	Signoff transaction
GRNAME	YES	YES	YES	YES	No default	Generic resource name for CICS TORs
GRPLIST	YES	YES	YES	YES	DFHLIST	List name of CSD groups for startup
GTFTR	YES	YES	YES	YES	OFF	GTF trace option
HPO	NO	NO	NO	YES	NO	z/OS Communications Server High Performance Option (HPO)
HTTPSERVERHDR	YES	YES	YES	YES	YES	Value set in the server header for an HTTP response
HTTPUSRAGENTHDR	YES	YES	YES	YES	YES	Value set in the user-agent header for an HTTP request
ICP	YES	YES	YES	YES	COLD	Interval control program start option
ICV	YES	YES	YES	YES	1000	Region exit interval (milliseconds)
ICVR	YES	YES	YES	YES	2000	Runaway task interval (milliseconds)
ICVTSD	YES	YES	YES	YES	0	Terminal scan delay interval

Table 1. System initialization parameters with override options and default settings (continued)

Parameter	PARM	SYSIN	System console	DFHSIT	Default value	Description
INFOCENTER	YES	YES	YES	YES	No default	The server name of where the CICS Information Center is installed and the port number that it uses to run in server mode
INITPARM	YES	YES	YES	YES	No default	Initialization parameters for programs
INTTR	YES	YES	YES	YES	ON	CICS internal trace option
IRCSTRT	YES	YES	YES	YES	NO	Interregion communication start
ISC	YES	YES	YES	YES	NO	Intersystem communication option
JESDI	YES	YES	YES	YES	30	JES delay interval for XRF alternate
JVMPROFILEDIR	YES	YES	YES	YES	/usr/lpp/cicsts /cicsts54/ JVMProfiles	JVM profile directory
KEYRING	YES	YES	YES	YES	No default	Key ring to be used by SSL support
LGDFINT	YES	YES	YES	YES	5	Log defer interval in Log Manager
LGNMSG	YES	YES	YES	YES	NO	Extract z/OS Communications Server logon data
LLACOPY	YES	YES	YES	YES	YES	Use MVS LLACOPY support
LOCALCCSID	YES	YES	YES	YES	037	The default CCSID for the local region
LPA	YES	YES	YES	YES	NO	Use LPA option for CICS/user modules
MAXSOCKETS	YES	YES	YES	YES	65535	Maximum number of IP sockets
MAXSSLTCBS	YES	YES	YES	YES	32	Limit on number of SSL TCBs
MCT	YES	YES	YES	YES	NO	Monitoring control table option or suffix
MINTLSLEVEL	YES	YES	YES	YES	TLS10	Level of encryption for TLS
MN	YES	YES	YES	YES	OFF	CICS monitoring option
MNCONV	YES	YES	YES	YES	NO	Monitoring converse recording option
MNEXC	YES	YES	YES	YES	OFF	Monitoring exception class option
MNFREQ	YES	YES	YES	YES	0	Monitoring frequency period
MNIDN	YES	YES	YES	YES	OFF	Monitoring identity class option

Table 1. System initialization parameters with override options and default settings (continued)

Parameter	PARM	SYSIN	System console	DFHSIT	Default value	Description
MNPER	YES	YES	YES	YES	OFF	Monitoring performance class option
MNRES	YES	YES	YES	YES	OFF	Monitoring resource class option
MNSYNC	YES	YES	YES	YES	NO	Monitoring syncpoint recording option
MNTIME	YES	YES	YES	YES	GMT	Monitoring timestamp (GMT or LOCAL)
MQCONN	YES	YES	YES	YES	NO	Do not connect to WebSphere® MQ at startup
MROBTCH	YES	YES	YES	YES	1	Number of MRO requests to batch
MROFSE	YES	YES	YES	YES	NO	Extend lifetime of long-running mirror
MROLRM	YES	YES	YES	YES	NO	Long-running mirror task option
MSGCASE	YES	YES	YES	YES	MIXED	CICS messages in mixed case
MSGLVL	YES	YES	YES	YES	1	System console MSG level option
MXT	YES	YES	YES	YES	250	Maximum number of tasks in CICS
NATLANG	YES	YES	YES	YES	E	List of national languages
NCPLDFT	YES	YES	YES	YES	DFHNC001	Named counter default pool name
NEWSIT	YES	YES	YES	NO	NO	Load a specified SIT and enforce use of all system initialization parameters
NISTSP800131A	YES	YES	YES	YES	NOCHECK	Whether the CICS region is to check for compliance to the NIST 800-131A standard
NONRLSRECOV	YES	YES	YES	YES	VSAMCAT	Select location of recovery options for non-RLS files
NQRNL	YES	YES	YES	YES	NO	RNL processing by z/OS global resource serialization
OFFSITE	YES	YES	YES	NO	NO	Restart in off-site recovery mode
OPERTIM	YES	YES	YES	YES	120	Write to operator timeout (seconds)
OPNDLIM	YES	YES	YES	YES	10	OPNDST/CLSDST request limit
PARMERR	YES	YES	YES	YES	INTERACT	System initialization parameter errors option
PDI	YES	YES	YES	YES	30	Primary delay interval - XRF active

Table 1. System initialization parameters with override options and default settings (continued)

Parameter	PARM	SYSIN	System console	DFHSIT	Default value	Description
PDIR	YES	YES	YES	YES	NO	DL/I PSB directory option or suffix
PGAICTLG	YES	YES	YES	YES	MODIFY	PG autoinstall catalog state
PGAIEXIT	YES	YES	YES	YES	DFHPGADX	PG autoinstall exit program
PGAIPGM	YES	YES	YES	YES	INACTIVE	PG autoinstall state
PGCHAIN	YES	YES	YES	YES	No default	BMS CHAIN command
PGCOPY	YES	YES	YES	YES	No default	BMS COPY command
PGPURGE	YES	YES	YES	YES	No default	BMS PURGE command
PGRET	YES	YES	YES	YES	No default	BMS RETURN command
PLTPI	YES	YES	YES	YES	NO	Program list table PI option or suffix
PLTPISEC	YES	YES	NO	YES	NONE	No PLT security checks on PI programs
PLTPIUSR	YES	YES	NO	YES	CICS region userid	PLT PI user ID = CICS region user ID
PLTSD	YES	YES	YES	YES	NO	Program list table SD option or suffix
PRGDLAY	YES	YES	YES	YES	0	BMS purge delay interval
PRINT	YES	YES	YES	YES	NO	Print key option
PRTYAGE	YES	YES	YES	YES	1000	Dispatcher priority aging value
PRVMOD	YES	YES	YES	NO	No default	Names of modules that are not to be used from the LPA
PSBCHK	YES	YES	NO	YES	NO	PSB resource checking required
PSDINT	YES	YES	YES	YES	0	Persistent session delay interval
PSTYPE	YES	YES	YES	YES	SNPS	z/OS Communications Server single node persistent Sessions
PVDELAY	YES	YES	YES	YES	30	Timeout value for LUIT table
QUIESTIM	YES	YES	YES	YES	240	Timeout value for quiesce requests
RACFSYNC	YES	YES	NO	YES	YES	Listen for type 71 ENF events
RAMAX	YES	YES	YES	YES	256	Maximum I/O area for RECEIVE ANY
RAPOOL	YES	YES	YES	YES	50	Maximum RECEIVE ANY request parm. lists
RDSASZE	YES	YES	YES	NO	0	Size of the RDSA
RENTPGM	YES	YES	YES	YES	PROTECT	Reentrant program write protection

Table 1. System initialization parameters with override options and default settings (continued)

Parameter	PARM	SYSIN	System console	DFHSIT	Default value	Description
RESP	YES	YES	YES	YES	FME	Logical unit response type
RESSEC	YES	YES	NO	YES	ASIS	Resource security check
RLS	YES	YES	YES	YES	NO	RLS option
RLSTOLSR	YES	YES	YES	YES	NO	RLS files in LSRPOOL build calculation
RMTRAN	YES	YES	YES	YES	CSGM	XRF alternate recovery transaction
RRMS	YES	YES	YES	YES	NO	Recoverable resource management services
RST	YES	YES	YES	YES	NO	Recovery service table (XRF-DBCTL)
RSTSIGNOFF	YES	YES	YES	YES	NOFORCE	XRF - Re-sign on after takeover
RSTSIGNTIME	YES	YES	YES	YES	500	XRF - sign off timeout value
RUWAPOOL	YES	YES	YES	YES	NO	Allocating storage pool for Language Environment
SDSASZE	YES	YES	YES	NO	0	Size of the SDSA
SDTRAN	YES	YES	YES	YES	CESD	Shutdown transaction
SEC	YES	YES	NO	YES	YES	External security manager option
SECPRFX	YES	YES	NO	YES	NO	Security prefix
SIT	YES	YES	YES	NO	No default	Suffix of the system initialization table that you want loading at the start of system initialization
SKRxxxx	YES	YES	YES	YES	No default	Single keystroke retrieval operation required
SNPRESET	YES	YES	NO	YES	UNIQUE	Specifies whether preset userid terminals share a single ACEE associated with the userid, or a unique ACEE for every terminal.
SNSCOPE	YES	YES	NO	YES	NONE	Multiple CICS sessions per user ID
SOTUNING	YES	YES	YES	YES	YES	Performance tuning for HTTP connections
SPCTR	YES	YES	YES	YES	(1,2)	Levels of special tracing required for CICS as a whole
SPCTRxx	YES	YES	YES	NO	(1,2)	The level of special tracing for a particular CICS component used by a transaction, terminal, or both

Table 1. System initialization parameters with override options and default settings (continued)

Parameter	PARM	SYSIN	System console	DFHSIT	Default value	Description
SPOOL	YES	YES	YES	YES	NO	System spooling interface option
SRBSVC	YES	YES	YES	YES	215	HPO Type 6 SVC number
SRT	YES	YES	YES	YES	1$	System recovery table option or suffix
SRVERCP	YES	YES	YES	YES	037	Default server code page to be used by the DFHCNV data conversion table (if the SRVERCP parameter in the DFHCNV macro is set to SYSDEF)
SSLCACHE	YES	YES	YES	YES	CICS	SSL session ID caching
SSLDELAY	YES	YES	YES	YES	600	SSL timeout value
START	YES	YES	YES	YES	AUTO	CICS system initialization option
STARTER	NO	NO	NO	YES	YES	Starter ($ and #) suffixes option **Note:** The default is NO but the parameter must be set to YES here to enable the SIT to assemble correctly.
STATEOD	YES	YES	YES	YES	0	Statistics end-of-day time
STATINT	YES	YES	YES	YES	010000	Statistics interval time
STATRCD	YES	YES	YES	YES	OFF	Statistics recording status
STGPROT	YES	YES	YES	YES	YES	Storage protection facility
STGRCVY	YES	YES	YES	YES	NO	Storage recovery option
STNTR	YES	YES	YES	YES	1	Level of standard tracing required for CICS as a whole
STNTRxx	YES	YES	YES	NO	1	Level of standard tracing you require for a particular CICS component
SUBTSKS	YES	YES	YES	YES	0	Number of concurrent mode TCBs
SUFFIX	NO	NO	NO	YES	$$	Suffix of this SIT
SYDUMAX	YES	YES	YES	YES	999	Number of SYSDUMPS to be taken
SYSIDNT	YES	YES	YES	YES	CICS	Local system identifier
SYSTR	YES	YES	YES	YES	ON	Master system trace flag
TAKEOVR	YES	YES	YES	YES	MANUAL	XRF alternate takeover option
TBEXITS	YES	YES	YES	YES	No default	Backout exit programs
TCP	YES	YES	YES	YES	YES	Terminal control program option or suffix
TCPIP	YES	YES	YES	YES	YES	TCP/IP support

Table 1. System initialization parameters with override options and default settings (continued)

Parameter	PARM	SYSIN	System console	DFHSIT	Default value	Description
TCSACTN	YES	YES	YES	YES	NONE	TC Shutdown action
TCSWAIT	YES	YES	YES	YES	4	TC Shutdown wait
TCT	YES	YES	YES	YES	NO	Terminal control table option or suffix
TCTUAKEY	YES	YES	YES	YES	USER	TCT user area storage key
TCTUALOC	YES	YES	YES	YES	ANY	TCT user area ANY
TD	YES	YES	YES	YES	(3,3)	Transient data buffers and strings
TDINTRA	YES	YES	YES	YES	NOEMPTY	Initial state of transient data queues
TRANISO	YES	YES	YES	YES	NO	Transaction Isolation
TRAP	YES	YES	YES	YES	OFF	F.E. global trap exit option
TRDUMAX	YES	YES	YES	YES	999	Number of TRANDUMPS to be taken
TRTABSZ	YES	YES	YES	YES	12288	Internal trace table size in 1K bytes
TRTRANSZ	YES	YES	YES	YES	1024	Transaction dump trace table size
TRTRANTY	YES	YES	YES	YES	TRAN	Transaction dump trace option
TS	YES	YES	YES	YES	(3,3)	Temporary storage buffers and strings
TSMAINLIMIT	YES	YES	YES	YES	64M	Upper limit of storage for TS main queues
TST	YES	YES	YES	YES	NO	Temporary storage table option or suffix
UDSASZE	YES	YES	YES	NO	0	Size of the UDSA
UOWNETQL	YES	YES	YES	YES	No default	Qualifier for NETUOWID
USERTR	YES	YES	YES	YES	ON	Master user trace flag
USRDELAY	YES	YES	YES	YES	30	Timeout value for user directory entries
USSCONFIG	YES	YES	YES	YES	/var/cicsts/ dfhconfig	The name and path of the root directory for CICS configuration files on z/OS UNIX
USSHOME	YES	YES	YES	YES	/usr/lpp/cicsts/ cicsts54	The name and path of the root directory for CICS files on z/OS UNIX
VTAM®	YES	YES	YES	YES	YES	z/OS Communications Server access method option
VTPREFIX	YES	YES	YES	YES	\	Client virtual terminal prefix
WEBDELAY	YES	YES	YES	YES	(5,60)	Web timer values

Table 1. System initialization parameters with override options and default settings (continued)

Parameter	PARM	SYSIN	System console	DFHSIT	Default value	Description
WLMHEALTH	YES	YES	YES	YES	(20,25)	Parameters used by CICS on z/OS WLM Health API (IWM4HLTH) calls to inform z/OS WLM about the health state of CICS regions.
WRKAREA	YES	YES	YES	YES	512	Common work area (CWA) size in bytes
XAPPC	YES	YES	NO	YES	NO	RACF® class APPCLU required
XCFGROUP	YES	YES	YES	YES	DFHIR000	XCF group to use for MRO communications
XCMD	YES	YES	NO	YES	YES	SPI use default name for RACF check
XDCT	YES	YES	NO	YES	YES	Security check for transient data queues
XDB2	YES	YES	YES	YES	NO	Security check for DB2ENTRY resources
XFCT	YES	YES	NO	YES	YES	Security check for files
XHFS	YES	YES	NO	YES	YES	Security check for z/OS UNIX files
XJCT	YES	YES	NO	YES	YES	Security check for journals
XLT	YES	YES	YES	YES	NO	Transaction list table option or suffix
XPCT	YES	YES	NO	YES	YES	Security check for started transactions
XPPT	YES	YES	NO	YES	YES	Security check for programs
XPSB	YES	YES	NO	YES	YES	Security check for DL/I PSBs
XPTKT	YES	YES	NO	YES	YES	Security check for PassTicket
XRES	YES	YES	NO	YES	YES	For resources subject to XRES security, checks use the default name for the RACF check. For a list of resources subject to XRES security checks, see Resource and command check cross-reference.
XRF	YES	YES	YES	YES	NO	Extended recovery feature (XRF) option
XTRAN	YES	YES	NO	YES	YES	Security check for transaction-attach
XTST	YES	YES	NO	YES	YES	Security check for temporary storage queues
XUSER	YES	YES	NO	YES	YES	Surrogate user checking to be done

ADI

The **ADI** parameter specifies the alternate delay interval in seconds for an alternate CICS region when you are running CICS with XRF.

ADI={30|number}

The minimum delay that you can specify is 5 seconds. This is the time that must elapse between the (apparent) loss of the surveillance signal in the active CICS region, and any reaction by the alternate CICS region. The corresponding parameter for the active is PDI. ADI and PDI need not have the same value.

Note: You must give careful consideration to the values you specify for the parameters ADI and JESDI so that they do not conflict with your installation's policy on PR/SM™ RESETTIME and the XCF INTERVAL and OPNOTIFY intervals. You should ensure that the sum of the interval you specify for ADI plus JESDI exceeds the interval specified by the XCF INTERVAL and the PR/SM policy interval RESETTIME.

AIBRIDGE

The **AIBRIDGE** parameter specifies whether the autoinstall user replaceable module (URM) is to be called when creating bridge facilities (virtual terminals) used by the 3270 bridge mechanism. Specify this parameter only in the bridge router region.

AIBRIDGE={AUTO|YES}

Valid values are as follows:

AUTO
: This is the default, and specifies that bridge facilities are defined automatically by CICS. The autoinstall URM is not called.

YES
: Specifies that the autoinstall URM is to be called for all new bridge facilities.

For information about writing an autoinstall user replaceable module, see Writing a program to control autoinstall of programs.

AICONS

The **AICONS** parameter specifies whether you want autoinstall support for consoles.

AICONS={NO|YES|AUTO}

You can also set the state of autoinstall support for consoles dynamically using the **SET AUTOINSTALL** command. Valid values for this parameter are as follows:

NO
: This is the default, and specifies that the CICS regions does not support autoinstall for consoles.

YES
: Specifies that console autoinstall is active and CICS is to call the autoinstall control program, as part of the autoinstall process, when an undefined console issues an MVS MODIFY command to CICS.

AUTO
: Specifies that console autoinstall is active but CICS is not to call the autoinstall control program when an undefined console issues an MVS MODIFY command to CICS. CICS is to autoinstall undefined consoles automatically without any input from the autoinstall control program. The 4-character termid required for the console's TCT entry is generated by CICS, beginning with a ¬ (logical not) symbol.

AIEXIT

The **AIEXIT** parameter specifies the name of the autoinstall user-replaceable program that you want CICS to use when autoinstalling local z/OS Communications Server terminals, APPC connections, virtual terminals, and shipped terminals and connections.

AIEXIT={DFHZATDX|DFHZATDY|name}

Autoinstall is the process of installing resource definitions automatically, using z/OS Communications Server logon or BIND data, model definitions, and an autoinstall program.

You can specify only one user-replaceable program on the **AIEXIT** parameter. Which of the CICS-supplied programs (or customized versions thereof) that you choose depends on what combination of resources you need to autoinstall.

For background information about autoinstall, see Autoinstall. Valid values for this parameter are as follows:

DFHZATDX
A CICS-supplied autoinstall user program. This value is the default. It installs definitions for:

- Locally-attached z/OS Communications Server terminals
- Virtual terminals used by the CICS Client products
- Remote shipped terminals
- Remote shipped connections

DFHZATDY
A CICS-supplied autoinstall user program. It installs definitions for:

- Locally-attached z/OS Communications Server terminals
- Local APPC connections
- Virtual terminals used by the CICS Client products
- Remote shipped terminals
- Remote shipped connections

name The name of your own customized autoinstall program, which can be based on one of the supplied sample programs. For programming information about writing your own autoinstall program, see Writing a program to control autoinstall of terminals.

AILDELAY

The **AILDELAY** parameter specifies the delay period that elapses after all sessions between CICS and an autoinstalled terminal, APPC device, or APPC system are ended, before the terminal or connection entry is deleted.

AILDELAY={0|hhmmss}

All sessions are ended when the terminal or system logs off, or when a transaction disconnects it from CICS.

The **AILDELAY** parameter does not apply to the following types of autoinstalled APPC connection, which are not deleted:

- Sync level 2-capable connections (for example, CICS-to-CICS connections)
- Sync level 1-only, limited resource connections installed on a CICS that is a member of a generic resource group

Valid values for this parameter are as follows:

hhmmss

A 1 to 6-digit number. The default is 0. For non-LU6.2 terminals and LU6.2 single-session connections installed by a CINIT, 0 means that the terminal entry is deleted as soon as the session is ended. For LU6.2 connections installed by a BIND, 0 means that the connection is deleted as soon as all sessions are ended, but is reusable if a new BIND occurs before the deletion starts.

If you leave out the leading zeros, they are supplied (for example, 123 becomes 000123—that is, 1 minute 23 seconds).

AIQMAX

The **AIQMAX** parameter specifies the maximum number of z/OS Communications Server terminals and APPC connections that can be queued concurrently for autoinstall, the limit is the sum of installs and deletes.

AIQMAX={100|number}

The value for this parameter must be a number in the range 0 through 999. The default is 100. A zero value disables the autoinstall function.

Specify a number that is large enough to allow for installs and deletes of both APPC connections and terminals.

Note: This value does not limit the total number of terminals that can be autoinstalled. If you have a large number of terminals autoinstalled, shutdown can fail due to the **MXT** system initialization parameter being reached or CICS becoming short on storage. For information about preventing this possible cause of shutdown failure, see MVS and DASD: improving performance in Improving performance.

AIRDELAY

The **AIRDELAY** parameter specifies the delay period that elapses after an emergency restart before autoinstalled terminal and APPC connection entries that are not in session are deleted.

AIRDELAY={700|hhmmss}

The **AIRDELAY** parameter also applies when you issue a z/OS Communications Server**CEMT SET VTAM OPEN** command after a z/OS Communications Server abend and PSTYPE=MNPS is coded. This causes autoinstalled resources to be deleted, if the session was not restored and has not been used since the ACB was opened.

The **AIRDELAY** parameter does not apply to the following types of autoinstalled APPC connection, which are always written to the CICS global catalog and recovered during a warm or emergency start:

- Sync level 2-capable connections (for example, CICS-to-CICS connections)
- Sync level 1-capable, limited resource connections installed on a CICS that is a member of a generic resource group

hhmmss

A 1-to 6-digit number. If you leave out the leading zeros, they are supplied. The default is 700, meaning a delay of 7 minutes. A value of 0 means that autoinstalled definitions are not written to the global catalog and therefore are not restored at an emergency restart.

For guidance about the performance implications of setting different **AIRDELAY** values, see MVS and DASD: improving performance in Improving performance.

AKPFREQ

The **AKPFREQ** parameter specifies the number of write requests to the CICS system log stream output buffer required before CICS writes an activity keypoint.

AKPFREQ={4000|number}

4000 This is the default. You are recommended to allow **AKPFREQ** to assume its default value.

number

number can be 0 (zero) or any value in the range 50 through 65535. You cannot specify a number in the range 1 - 49.

If you specify AKPFREQ=0, no activity keypoints are written, with the following consequences:

- The CICS system log automatic deletion mechanism does not work so efficiently in this situation. The average system log occupancy would merely increase, maybe dramatically for some users. Without efficient automatic deletion, the log stream spills on to auxiliary storage, and from there onto tertiary storage (unless you control the size of the log stream yourself).
- Emergency restarts are not prevented, but the absence of activity keypoints on the system log affects the performance of emergency restarts because CICS must read backwards through the entire log stream.
- Backout-while-open (BWO) support is seriously affected, because without activity keypointing, tie-up records are not written to the forward recovery logs and the data set recovery point is not updated. Therefore, for forward recovery to take place, all forward recovery logs must be kept since the data set was first opened for update after the last image copy. For more information about the effect of AKPFREQ=0 on BWO, see Effect of disabling activity keypointing.
- Replication support is affected, because without activity keypointing, tie-up records are not written to replication logs. This can affect the performance of the replication engine.

For more information about activity keypointing, see The activity keypoint frequency (AKPFREQ) .

APPLID

The **APPLID** parameter specifies the z/OS Communications Server application identifier for this CICS region.

APPLID={DBDCCICS|*applid*}

Valid values are as follows:

applid This name, 1 through 8 characters, identifies the CICS region in the z/OS Communications Server network. It must match the name field specified in the APPL statement of the z/OS Communications Server VBUILD TYPE=APPL definition. For an example, see Defining specific APPL definitions and APPL parameters to SNA.

If CICS is running in a sysplex, its APPLID must be unique in the sysplex.

This parameter can be used also as the application identifier of this CICS region on IPIC connections.

When you define this CICS region to another CICS region, in an MRO or ISC over SNA CONNECTION definition you specify the APPLID using the NETNAME attribute; in an IPIC IPCONN definition you specify the APPLID using the APPLID attribute.

When sharing a DL/I database with a batch region, the APPLID is used by the batch region to identify the CICS region.

If the CICS region uses XRF, the form of the **APPLID** parameter is as follows:

APPLID=(*generic_applid*,*specific_applid*)

Specifies the generic and specific XRF APPLIDs for the CICS region. Both APPLIDs must be 1 through 8 characters long and the specific APPLID must be unique in the sysplex. If, on CICS startup, the specified specific APPLID is found to duplicate the (specific or only) APPLID of any other CICS region currently active in the sysplex, CICS issues message DFHPA1946 and fails to initialize.

generic_applid

The generic APPLID for both the active and the alternate CICS regions. You must specify the same name for *generic_applid* on the **APPLID** system initialization parameter for both CICS regions. Because IRC uses the *generic_applid* to identify the CICS regions, there can be no IRC connection for an alternate CICS region until takeover has occurred and the alternate CICS region becomes the active CICS region.

When you define this XRF pair to another CICS region, in an MRO or ISC over SNA CONNECTION definition you specify the generic APPLID using the NETNAME attribute; in an IPIC IPCONN definition you specify the generic APPLID using the APPLID attribute.

When sharing a DL/I database with a batch region, this name is used by the batch region to identify the CICS region. CICS passes the generic applid to DBRC, because the alternate system does not sign on to DBRC until it has completed takeover.

Do not confuse the term *generic applid* with *generic resource name*. Generic APPLIDs apply only to CICS regions that use XRF. Generic resource names apply only to z/OS Communications Server generic resource groups.

specific_applid

Specifies the CICS region in the z/OS Communications Server network. It must match the label specified in the z/OS Communications Server VBUILD TYPE=APPL definition. You must specify a different *specific_applid* on the **APPLID** system initialization parameter for the active and for the alternate CICS region. Also, *generic_applid* and *specific_applid* must be different.

The active and alternate CICS regions use the z/OS Communications Server **MODIFY USERVAR** command to set a user application name variable, so users do not need to know which CICS region is active at any instant.

AUTCONN

The **AUTCONN** parameter specifies that the reconnection of terminals after an XRF takeover is to be delayed, to allow time for manual switching.

AUTCONN={0|hhmmss} (alternate)
The delay is *hh* hours, *mm* minutes, and *ss* seconds. The default value of zero means that there is no delay in the attempted reconnection.

The interval specified is the delay before the CXRE transaction runs. CXRE tries to reacquire any XRF-capable (class 1) terminal session that failed to get a backup session, or failed the switch for some other reason. CXRE tries to reacquire other terminals that were in session at the time of the takeover.

Note that the same delay interval applies to the connection of terminals with AUTOCONNECT(YES) specified in the TYPETERM definition, at a warm or emergency restart, whether or not you have coded XRF=YES.

AUTODST

The **AUTODST** parameter specifies whether CICS is to activate automatic dynamic storage tuning for application programs.

AUTODST={NO|YES}
Valid values are as follows:

NO Automatic dynamic storage tuning is not required and CICS does not request this support from Language Environment.

YES Automatic dynamic storage tuning is required. This is activated during CICS startup when Language Environment is being initialized. CICS indicates that it can support dynamic storage tuning to Language Environment, and if Language Environment responds by indicating that it also supports the facility, CICS and Language Environment are synchronized to provide the required support.

For more information, see the appropriate z/OS Language Environment information.

AUTORESETTIME

The **AUTORESETTIME** parameter specifies the action CICS takes for automatic time changes.

AUTORESETTIME={IMMEDIATE|NO|YES}
Valid values are as follows:

IMMEDIATE
CICS issues a PERFORM RESET command to synchronize the CICS time-of-day with the system time-of-day if, at the next task attach, the CICS time-of-day differs from the system time-of-day. CICS issues message DFHIC0801 when the times are synchronized.

NO CICS issues message DFHAP1500 to indicate that a **CEMT PERFORM RESET** command is required to synchronize the CICS time-of-day with the system time-of-day.

YES CICS issues a **PERFORM RESET** command to synchronize the CICS time-of-day with the system time-of-day if, at the next local midnight, the CICS time-of-day differs from the system time-of-day by more than 30 minutes. For example, setting clocks forward or back to adjust for summer and winter time. CICS issues message DFHIC0801 when the times are synchronized.

Note: Setting clocks back might cause end-of-day statistics to be written twice.

AUXTR

The **AUXTR** parameter specifies whether the auxiliary trace destination is to be activated at system initialization.

AUXTR={OFF|ON}

This parameter controls whether any of the three types of CICS trace entry are written to the auxiliary trace data set. The three types are: CICS system trace (see the SYSTR parameter), user trace (see the USERTR parameter), and exception trace entries (that are always made and are not controlled by a system initialization parameter).

OFF Do not activate auxiliary trace.

ON Activate auxiliary trace.

For details of internal tracing in main storage, see INTTR.

AUXTRSW

The **AUXTRSW** parameter specifies whether you want the auxiliary trace autoswitch facility.

AUXTRSW={NO|ALL|NEXT}

Valid values are as follows:

NO Disables the autoswitch facility.

NEXT Enables the autoswitch facility to switch to the next data set at end of file of the first data set used for auxiliary trace. Coding NEXT permits one switch only, and when the second data set is full, auxiliary trace is switched off.

ALL Enable the autoswitch facility to switch to the inactive data set at every end of file. Coding ALL permits continuous switching between the two auxiliary trace data sets, DFHAUXT and DFHBUXT, and whenever a data set is full, it is closed and the other data set is opened.

BMS

The **BMS** system initialization parameter specifies which version of basic mapping support you require in CICS.

BMS=({MINIMUM|STANDARD|FULL }[,COLD][,{UNALIGN |ALIGN}] [,{ DDS|NODDS}])

The function included in each version of BMS is shown in BMS support levels. The parameter **BMS** can be overridden during CICS initialization.

MINIMUM
The minimum version of BMS is included.

STANDARD
The standard version of BMS is included.

FULL The full version of BMS is included. This value is the default.

COLD
CICS deletes delayed messages from temporary storage, and destroys their interval control elements (ICEs). COLD forces the deletion of messages regardless of the value in effect for START. If COLD is not specified, the availability of messages depend on the values in effect for the **START** and **TS** parameters.

UNALIGN
Specifies that all BMS maps assembled before CICS/OS/VS Version 1 Release 6 are unaligned. Results are unpredictable if the stated alignment does not match the actual alignment.

ALIGN
All BMS maps assembled before CICS/OS/VS Version 1 Release 6 are aligned.

DDS BMS is to load suffixed versions of map sets and partition sets. BMS first tries to load a version that has the alternate suffix (if the transaction uses the alternate screen size). If the load fails, BMS tries to load a version that has the default map suffix. If this fails too, BMS tries to load the unsuffixed version. DDS, which stands for "device dependent suffixing", is the default.

You need to use map suffixes only if the same transaction is to be run on terminals with different characteristics (in particular, different screen sizes). If you do not use suffixed versions of map sets and partition sets, CICS need not test for them.

NODDS
BMS is not to load suffixed versions of map sets and partition sets. Specifying NODDS avoids the search for suffixed versions, saving processor time.

Table 2. Versions of BMS

BMS version	Devices supported	Function provided
MINIMUM	All 3270 system display units and printers except SNA character string printers, which are defined as DEVICE(SCSPRINT) on the RDO TYPETERM definition or as TRMTYPE=SCSPRT in DFHTCT	SEND MAP command, RECEIVE MAP command, SEND CONTROL command. Default and alternate screens; extended attributes; map set suffixes; screen coordination with null maps; and block data
STANDARD	All devices supported by BMS. These are listed in BMS support levels	All function of MINIMUM, as well as outboard formats, partitions, controlling a magnetic slot reader, NLEOM mode for 3270 system printers, SEND TEXT command, and Subsystem LDC controls.

Table 2. Versions of BMS (continued)

BMS version	Devices supported	Function provided
FULL	All devices supported by BMS. These are listed in BMS support levels	Same as STANDARD, as well as terminal operator paging, cumulative mapping, page overflow, cumulative text processing, routing, message switching returning BMS-generated data stream to program before output.

BRMAXKEEPTIME

The **BRMAXKEEPTIME** parameter specifies the maximum time (in seconds) that bridge facilities (virtual terminals used by the 3270 bridge) are kept if they are not used.

BRMAXKEEPTIME={86400|number}

The client application can specify this timeout value when it sends a request to run a transaction using the Link3270 bridge. If the client specifies a larger value than the BRMAXKEEPTIME value in the AOR, then CICS will change this parameter in the link parameter list.

number

The maximum timeout value that a client can specify (in seconds), before an unused bridge facility is deleted. The value specified must be in the range 0 to 86400. A value of 0 means that bridge facilities are never kept at the end of a transaction, therefore CICS will not be able to run pseudoconversational transactions. This may be useful if the region is only used for inquiry transactions. The default value is 24 hours (86400 seconds).

CDSASZE

The **CDSASZE** system initialization parameter specifies the size of the CDSA.

CDSASZE={0K|number}

The default size is 0, indicating that the DSA size can change dynamically. A non-zero value indicates that the DSA size is fixed.

number

specify number as an amount of storage in the range 0 to 16777215 bytes in multiples of 262144 bytes (256KB). If the size specified is not a multiple of 256KB, CICS rounds the value up to the next multiple.

You can specify number in bytes (for example, 4194304), or as a whole number of kilobytes (for example, 4096K), or a whole number of megabytes (for example, 4M).

Restriction: You can specify the **CDSASZE** parameter in PARM, SYSIN, or CONSOLE only.

Important: Setting the size of individual dynamic storage areas (DSAs) is not usually necessary and is not recommended. If you specify DSA size values that in combination do not allow sufficient space for the remaining DSAs, CICS fails to initialize. The limit on the storage available for the DSAs in 24-bit storage (below the line) is specified by the DSALIM system initialization

parameter. You must allow at least 256K for each DSA in 24-bit storage for which you have not set a size. See DSA size limits.

CHKSTRM

The **CHKSTRM** parameter specifies that terminal storage-violation checking is to be activated or deactivated.

CHKSTRM={CURRENT|NONE}

Valid values are as follows:

CURRENT
TIOA storage violations are to be checked.

NONE
TIOA storage-violation checking is to be deactivated.

You can also use the CICS-supplied transaction, CSFE, to switch terminal storage-violation checking on and off.

For information about checking for storage violations, see What to do if CICS has stalled.

Restrictions You can specify the **CHKSTRM** parameter in PARM, SYSIN, or CONSOLE only.

CHKSTSK

The **CHKSTSK** parameter specifies that task storage-violation checking at startup is to be activated or deactivated.

CHKSTSK={CURRENT|NONE}

Valid values are as follows:

CURRENT
All storage areas on the transaction storage chain for the current task only are to be checked.

NONE
Task storage-violation checking is to be deactivated.

You can also use the CICS-supplied transaction, CSFE, to switch task storage-violation checking on and off.

For information about checking for storage violations, see What to do if CICS has stalled.

Restrictions You can specify the **CHKSTSK** parameter in PARM, SYSIN, or CONSOLE only.

CICSSVC

The **CICSSVC** parameter specifies the number that you have assigned to the CICS type 3 SVC.

CICSSVC={216|number}

The default number is 216. A CICS type 3 SVC with the specified or default number must be installed in the LPA. For information about installing the CICS SVC, see Installing the CICS SVCs in Installing.

CICS checks if the SVC number supplied corresponds to the correct level of the CICS Type 3 SVC module, DFHCSVC. If the SVC number does not correspond

to the correct level of DFHCSVC, the following can happen, depending on the value specified for the **PARMERR** system initialization parameter:

- CICS is terminated with a system dump
- The operator is allowed to retry using a different SVC number

For details of the **PARMERR** system initialization parameter, see PARMERR.

CILOCK

The **CILOCK** parameter specifies whether or not the control interval lock of a non-RLS VSAM file is to be kept after a successful read-for-update request.

CILOCK={NO|YES}

Valid values are as follows:

NO is the default and specifies that the control interval is to be freed. This allows other tasks to access other records in the same control interval, without an exclusive control conflict occurring. In these cases throughput should be greater. Note that the record lock on the record for which the read-for-update was first issued, still prevents other tasks from updating this record, even though the control interval lock has been released. When the record is rewritten or deleted, the read-for-update is reissued to VSAM as part of the update processing.

If a WRITE is issued by another task during a READ UDPATE, the WRITE receives a DUPREC condition.

YES specifies that the control interval is not to be freed. This means that a subsequent rewrite or delete request does not need to reissue the read-for-update request to VSAM. However, if other tasks attempt to access other records in the same control interval, an exclusive control conflict occurs on this control interval, forcing these tasks to wait until the update request completes.

CLINTCP

The **CLINTCP** parameter specifies the default client code page to be used by the DFHCNV data conversion table, but only if the **CLINTCP** parameter in the DFHCNV macro is set to SYSDEF.

CLINTCP={437|code page}

The *code page* is a field of up to 8 characters and can take the values supported by the **CLINTCP** parameter in the DFHCNV macro. For the list of valid code pages, see CICS-supported conversions. The default is 437.

CLSDSTP

The CLSDSTP system initialization parameter specifies the notification required for an **EXEC CICS ISSUE PASS** command.

CLSDSTP={NOTIFY|NONOTIFY}

This parameter is applicable to both autoinstalled and non-autoinstalled terminals. You can use the notification in a user-written node error program to reestablish the CICS session when a z/OS Communications Server VTAM CLSDST PASS request resulting from an EXEC CICS ISSUE PASS command fails. For more information about the EXEC CICS ISSUE PASS command, see ISSUE PASS.

NOTIFY
CICS requests notification from z/OS Communications Server when the EXEC CICS ISSUE PASS command is executed.

NONOTIFY
CICS does not request notification from z/OS Communications Server.

CLT

The **CLT** parameter specifies the suffix for the command list table (CLT), if this SIT is used by an alternate XRF system.

CLT=xx (alternate)
The name of the table is DFHCLTxx. For information about coding the macros for this table, see Command list table (CLT) .

CMDPROT

The **CMDPROT** parameter specifies whether to allow or inhibit CICS validation of start addresses of storage referenced as output parameters on EXEC CICS commands.

CMDPROT={YES|NO}
Valid values are as follows:

YES CICS validates the initial byte at the start of any storage that is referenced as an output parameter on EXEC CICS commands to ensure that the application program has write access to the storage. This ensures that CICS does not overwrite storage on behalf of the application program when the program itself cannot do so. If CICS detects that an application program has asked CICS to write into an area to which the application does not have addressability, CICS abends the task with an AEYD abend.

The level of protection against bad addresses depends on the level of storage protection in the CICS environment. The various levels of protection provided when you specify CMDPROT=YES are shown in Table 3.

NO CICS does not perform any validation of addresses of the storage referenced by EXEC CICS commands. This means that an application program could cause CICS to overwrite storage to which the application program itself does not have write access.

Table 3. Levels of protection provided by CICS validation of application-supplied addresses

Environment	Execution key of affected programs	Types of storage referenced by applications that cause AEYD abends
Read-only storage (RENTPGM=PROTECT)	CICS-key and user-key	CICS key 0 read-only storage (RDSA and ERDSA).
Subsystem storage protection (STGPROT=YES)	User-key	All CICS-key storage (CDSA and ECDSA)
Transaction isolation (TRANISO=YES)	User-key and ISOLATE(YES)	Task-lifetime storage of all other transactions
Transaction isolation (TRANISO=YES)	User-key and ISOLATE(NO)	Task-lifetime storage of all **except** other user key and ISOLATE(NO) transactions

Table 3. Levels of protection provided by CICS validation of application-supplied addresses (continued)

Environment	Execution key of affected programs	Types of storage referenced by applications that cause AEYD abends
Base CICS (all storage is CICS key 8 storage) (RENTPGM=NOPROTECT; STGPROT=NO; and TRANISO=NO)	CICS-key and user-key	MVS storage only

CMDSEC

The **CMDSEC** parameter specifies whether or not you want CICS to honor the CMDSEC option specified on a transaction's resource definition.

CMDSEC={ASIS|ALWAYS}

Valid values are as follows:

ASIS means that CICS honors the CMDSEC option defined in a transaction's resource definition. CICS calls its command security checking routine only when CMDSEC(YES) is specified in a transaction resource definition.

ALWAYS
CICS overrides the CMDSEC option, and always calls its command security checking routine to issue the appropriate call to the SAF interface.

Note:

1. Specify ALWAYS when you want to control the use of the SPI in all your transactions. Be aware that this might incur additional overhead. The additional overhead is caused by CICS issuing the command security calls on every eligible EXEC CICS command, which are *all* the system programming interface (SPI) commands.
2. If you specify ALWAYS, command checking applies to CICS-supplied transactions such as CESN and CESF. You must authorize all users of CICS-supplied transactions to use the internal CICS resources for the transactions, otherwise you will get unexpected results in CICS-supplied transactions.

Restrictions You can specify the CMDSEC parameter in the SIT, PARM, or SYSIN only.

CONFDATA

The **CONFDATA** parameter specifies whether CICS is to suppress user data that might otherwise appear in CICS trace entries or in dumps.

CONFDATA={SHOW|HIDETC}

This option applies to initial input data received on:

- A z/OS Communications Server RECEIVE ANY operation
- An MRO connection
- An IPIC connection
- FEPI screens and RPLAREAs

This option also applies to the CICS client use of a Virtual Terminal. Data is traced before and after code page conversion and is suppressed if HIDETC is used in combination with CONFDATA YES in the transaction.

SHOW
Data suppression is not in effect. User data is traced regardless of the CONFDATA option specified in transaction resource definitions. This option overrides the CONFDATA option in transaction resource definitions.

HIDETC
CICS is to 'hide' user transport data from CICS trace entries. The action taken by CICS is subject to the individual CONFDATA attribute on the transaction resource definition (see Table 4 on page 26).

If you specify CONFDATA=HIDETC, CICS processes z/OS Communications Server, MRO, IS, and FEPI user data as follows:

- **z/OS Communications Server**: CICS clears the z/OS Communications Server RAIA containing initial input as soon as it has been processed, and before the target transaction has been identified.

  The normal trace entries (FC90 and FC91) are created on completion of the RECEIVE ANY operation with the text `SUPPRESSED DUE TO CONFDATA=HIDETC IN SIT` replacing all the user data except the first 4 bytes of normal data, or the first 8 bytes of function management headers (FMHs).

  CICS then identifies the target transaction for the data. If the transaction definition specifies CONFDATA(NO), CICS traces the user data that it suppressed from the FC90 trace in the trace entry AP FC92. This trace entry is not created if the transaction is defined with CONFDATA(YES).
- **MRO**: CICS does not trace the initial input received on an MRO link.

  The normal trace entries (DD16, DD23, and DD25) are created with the text `SUPPRESSED DUE TO CONFDATA=HIDETC IN SIT` replacing all the user data.

  CICS then identifies the target transaction for the data. If the transaction definition specifies CONFDATA(NO), CICS traces the user data that it suppressed from DD16 in the trace entry AP FC92. This special trace entry is not created if the transaction is defined with CONFDATA(YES).
- **IPIC**: Trace points SO 0201 and SO 0202 suppress buffer data with the message `Trace data suppressed because it may contain sensitive data`. Subsequent trace point SO 029D (buffer continuation) and buffer data from trace points WB 0700 and WB 0701 is suppressed.

  If the transaction definition specifies CONFDATA(NO), IS trace entries are created with the user data, as normal.

  If the transaction definition specifies CONFDATA(YES), user data from IS trace points IS 0602, IS 0702, and IS 0906 is replaced with `SUPPRESSED DUE TO CONFDATA=HIDETC IN SIT`. Data from IS trace points IS 0603 and IS 0703 is not shown.
- **FEPI**: FEPI screens and RPL data areas (RPLAREAs) areas are suppressed from all FEPI trace points if CONFDATA(YES) is specified in the transaction resource definition. The user data in the

FEPI trace points AP 1243, AP 1244, AP 145E, AP 145F, AP 1460, AP 1461, AP 1595, AP 1596, AP 1597, AP 1598, and AP 1599 is replaced with the message `SUPPRESSED DUE TO CONFDATA=HIDETC IN SIT`. If the transaction definition specifies CONFDATA(NO), the FEPI trace entries are created with the user data as normal.

Mirror transactions: The CICS-supplied mirror transaction definitions are specified with CONFDATA(YES). This ensures that, when you specify CONFDATA=HIDETC as a system initialization parameter, CICS regions running mirror transactions suppress user data as described for z/OS Communications Server, MRO, and IS data.

Modified data: By waiting until the transaction has been identified to determine the CONFDATA option, z/OS Communications Server, MRO, or IS data may have been modified (for example, it may have been translated to uppercase).

The interaction between the **CONFDATA** system initialization parameter and the CONFDATA attribute on the transaction resource definition is shown in Table 4.

Table 4. Effect of CONFDATA system initialization and transaction definition parameters

CONFDATA on transaction	CONFDATA system initialization parameter	
	SHOW	HIDETC
NO	Data not suppressed	Data not suppressed
YES	Data not suppressed	Data suppressed

You cannot modify the CONFDATA option while CICS is running. You must restart CICS to apply a change.

Restriction: You can specify the **CONFDATA** parameter in the SIT, PARM, and SYSIN only.

CONFTXT

The **CONFTXT** system initialization parameter specifies whether CICS is to prevent z/OS Communications Server from tracing user data.

CONFTXT={NO|YES}

Valid values are as follows:

NO CICS does not prevent z/OS Communications Server from tracing user data.

YES CICS prevents z/OS Communications Server from tracing user data.

Restriction: You can specify the **CONFTXT** parameter in the SIT, PARM, and SYSIN only.

CPSMCONN

The **CPSMCONN** parameter specifies whether you want CICS to invoke the specified CICSPlex® SM component during initialization of the region.

CPSMCONN={NO|CMAS|LMAS|WUI}

You can initialize the region as one of the following:

- A CICSPlex SM address space (CMAS)

- A CICSPlex SM local managed application system (MAS)
- A CICSPlex SM Web User Interface (WUI) server

NO Do not invoke any CICSPlex SM initialization code in this region.

CMAS
Invoke CICSPlex SM code automatically during CICS initialization to initialize the region as a CMAS. The other information CICSPlex SM needs for a CMAS is taken from the CMAS parameters read from the EYUPARM data set.

Specifying **CPSMCONN**=CMAS is the recommended alternative to specifying the CICSPlex SM CMAS initialization program in a CICS post-initialization program list table (PLTPI).

Note: If you specify **CPSMCONN**=CMAS, ensure that your CICS region startup JCL EXEC statement specifies the name of the CICSPlex SM CMAS program, EYU9XECS. For example:

```
//CMAS    EXEC PGM=EYU9XECS,...,...
```

LMAS Invoke CICSPlex SM code automatically during CICS initialization to initialize the region as a local MAS. The other information CICSPlex SM needs for a MAS is taken from the MAS parameters read from the EYUPARM data set.

Specifying **CPSMCONN**=LMAS is the recommended alternative to specifying the CICSPlex SM MAS initialization program in a CICS post-initialization program list table (PLTPI).

WUI Invoke CICSPlex SM code automatically during CICS initialization to initialize the region as a CICSPlex SM WUI server. The other information CICSPlex SM needs is taken from the MAS and WUI parameters read from the EYUPARM and EYUWUI data sets respectively.

Specifying **CPSMCONN**=WUI is the recommended alternative to specifying the CICSPlex SM MAS and WUI initialization and shutdown programs in initialization and shutdown program list tables (PLTPI and PLTSD).

Note: Using the **CPSMCONN** parameter has the same effect as specifying the relevant CICSPlex SM program in a program list table. This means that MASPLTWAIT and other PLT-related CICSPlex SM parameters are still valid and should be specified as necessary. However, there differences in the way the user ID under which the CICSPlex SM component runs is derived:

- If the CICSPlex SM component is started by a PLT program, it runs under the authority of the PLTPIUSR. For more information, see PLTPIUSR.
- If the CICSPlex SM component is started by **CPSMCONN**, then the ID is derived as follows:
  - If a Program List Table is specified on the PLTPI system initialization parameter, it runs under the authority of the PLTPIUSR. For more information, see PLTPIUSR.
  - If no Program List Table is specified on the PLTPI system initialization parameter, it runs under the authority of the CICS region user ID.

For more information about the user ID under which PLT programs run, see PLT programs.

For more information about the PLTPIUSR parameter, see PLTPIUSR system initialization parameter.

For more information about the CICS region user ID, see Specifying the CICS region userid.

For information about starting CICSPlex SM address spaces, see Installing.

CRLPROFILE

The **CRLPROFILE** parameter specifies the name of the profile that is used to authorize CICS to access the certification revocation lists (CRLs) that are stored in an LDAP server.

CRLPROFILE=*PROFILENAME*

The profile name is specified in the external security manager's LDAPBIND general resource class that contains bind information for an LDAP server. The profile name must be uppercase and can be up to 246 characters in length.

The profile must contain the name of the LDAP server and the distinguished name and password of a user who is authorized to extract certification revocation lists from it. For more information about setting up the profile, see Configuring LDAP for CICS use .

Specifying this parameter means that CICS checks each client certificate during the SSL negotiation for a revoked status using the certificate revocation lists in the LDAP server. If the certificate is revoked, CICS closes the connection immediately. If the **CRLPROFILE** parameter is omitted, CICS does not check the revoked status of certificates during SSL handshakes.

If the **CRLPROFILE** parameter is specified but is invalid, or if the specified profile contains invalid data, or if the LDAP server identified by the profile is unavailable when the CICS region starts, the CICS region disables its own access to the LDAP server and does not check the revoked status of certificates during SSL handshakes. Messages DFHSO0128 and DFHSO0129 report this problem. To restore access, you must fix the error and restart the CICS region.

The bind information for the LDAP server is cached in the SSL environment for the CICS region, which is managed by z/OS System SSL. When you issue the PERFORM SSL REBUILD command, the bind information for the LDAP server is refreshed from the external security manager. The PERFORM SSL REBUILD command cannot restore access to the LDAP server if the CICS region has disabled it. The refresh only takes place for an LDAP server that was available to the CICS region at the time when the command was issued.

CSDACC

The **CSDACC** parameter specifies the type of access to the CSD to be permitted to this CICS region.

CSDACC={READWRITE|READONLY}

This parameter is effective only when you start CICS with a START=COLD parameter. If you code START=AUTO, and CICS performs a warm or emergency restart, the file resource definitions for the CSD are recovered from the CICS global catalog. However, you can redefine the type of access permitted to the CSD dynamically with a **CEMT SET FILE** or **EXEC CICS SET FILE** command.

READWRITE
Read/write access is allowed, permitting the full range of CEDA, CEDB, and CEDC functions to be used.

READONLY
Read access only is allowed, limiting the CEDA and CEDB transactions to only those functions that do not require write access.

CSDBKUP

The **CSDBKUP** parameter specifies whether or not the CSD is eligible for BWO.

CSDBKUP={STATIC|DYNAMIC}
If you want to use BWO, specify CSDBKUP=DYNAMIC.

The **CSDBKUP**, **CSDRECOV**, and **CSDFRLOG** system initialization parameters interact according to how they are specified. For information about their effects when the SIT is assembled and during CICS override processing, see Planning for backup and recovery.

STATIC
All CICS files open for update against the CSD data set must be quiesced before a DFHSM and DFDSS backup of the CSD data set. The files must remain quiesced during the backup.

DYNAMIC
DFHSM and DFDSS are allowed to make a data set back up copy while CICS is updating the CSD.

Note that CSDBKUP=DYNAMIC is valid only if you have also specified CSDRECOV=ALL.

CSDBUFND

The **CSDBUFND** parameter specifies the number of buffers to be used for CSD data.

CSDBUFND=*number*
The minimum you should specify is the number of strings coded on the **CSDSTRNO** parameter plus 1, up to a maximum of 32768. Note that this parameter is used only if you have also coded CSDLSRNO=NONE; if you have coded CSDLSRNO=number, CSDBUFND is is set to a value of 0 and ignored.

If you specify a value for **CSDBUFND** that is less than the required minimum (the CSDSTRNO value plus 1), VSAM automatically changes the number of buffers to the number of strings plus 1 when CICS issues the OPEN macro for the CSD.

This parameter is effective only on a CICS cold or initial start. On a warm or emergency restart, file resource definitions for the CSD are recovered from the global catalog.

CSDBUFNI

The **CSDBUFNI** parameter specifies the number of buffers to be used for the CSD index.

CSDBUFNI=*number*
The minimum you should specify is the number of strings coded on the **CSDSTRNO** parameter, up to a maximum of 32768. This parameter is used only if

you have also coded CSDLSRNO=NONE; if you have coded CSDLSRNO=number, CSDBUFNI is is set to a value of 0 and ignored.

If you specify a value for **CSDBUFNI** that is less than the required minimum (the CSDSTRNO value), VSAM automatically changes the number of buffers to the number of strings when CICS issues the OPEN macro for the CSD.

This parameter is effective only on a CICS cold or initial start. On a warm or emergency restart, file resource definitions for the CSD are recovered from the global catalog.

CSDDISP

The **CSDDISP** parameter specifies the disposition of the data set to be allocated to the CSD.

CSDDISP={OLD|SHR}

If no JCL statement for the CSD exists when it is opened, the open is preceded by a dynamic allocation of the CSD using this disposition. If a DD statement exists in the JCL of the CICS startup job, it takes precedence over this disposition.

OLD The disposition of the CSD is set to OLD if dynamic allocation is performed.

SHR The disposition of the CSD is set to SHR if dynamic allocation is performed.

This parameter is effective only on a CICS cold or initial start. On a warm or emergency restart, file resource definitions for the CSD are recovered from the global catalog.

CSDDSN

The **CSDDSN** parameter specifies the 1-44 character JCL data set name (DSNAME) to be used for the CSD.

CSDDSN=*name*

If no JCL statement exists for the CSD when it is opened, the open is preceded by a dynamic allocation of the CSD using this DSNAME. If a DD statement exists in the JCL of the CICS startup job, it takes precedence over this DSNAME.

This parameter is effective only on a CICS cold or initial start. On a warm or emergency restart, file resource definitions for the CSD are recovered from the global catalog.

CSDFRLOG

The **CSDFRLOG** parameter specifies a number that corresponds to the journal name that CICS uses to identify the forward recovery log stream for the CSD.

CSDFRLOG=number

This parameter is meaningful only if CSDRECOV=ALL and CSDRLS=NO are specified, otherwise it is ignored. If you specify CSDRLS=NO and CSDRECOV=ALL, but omit CSDFRLOG (or specify CSDFRLOG=NO), the SIT assembly fails. However, if you specify an invalid combination as SIT overrides, CICS initialization will fail.

CSDBKUP, **CSDRECOV** and **CSDFRLOG** are ignored if CSDRLS=YES is specified. This is because recovery attributes (that is, the recoverability, the forward recovery LSN, and the BWO eligibility) must be specified in the ICF catalog for data sets that are opened in RLS mode.

The recovery attributes can also be specified (optionally) in the ICF catalog when you specify CSDRLS=NO. If you specify recovery attributes in both the ICF catalog and as system initialization parameters, the ICF catalog values are used (but see the next paragraph).

For a CSD opened in a non-RLS mode (CSDRLS=NO), the **CSDBKUP**, **CSDRECOV** and **CSDFRLOG** system initialization parameters interact according to how they are specified. For information about their effects when the SIT is assembled and during CICS override processing, see Planning for backup and recovery.

This parameter is effective only on a CICS cold or initial start. On a warm or emergency restart, file resource definitions for the CSD are recovered from the global catalog.

number
The journal number that identifies the user journal that CICS is to use for forward recovery of the CSD. CICS journal names are of the form DFHJ*nn* where *nn* is a number in the range 1 through 99. CICS maps the resulting journal name (DFHJ01—DFHJ99) to an MVS log stream.

CSDINTEG

The **CSDINTEG** parameter specifies the level of read integrity for the CSD if it is accessed in RLS mode.

CSDINTEG={UNCOMMITTED|CONSISTENT|REPEATABLE}
If the CSD is not accessed in RLS mode (CSDRLS=NO), a value for CSDINTEG of CONSISTENT or REPEATABLE will be changed to UNCOMMITTED.

UNCOMMITTED
The CSD is read without read integrity. For each read request, CICS obtains the current value of the record as known to VSAM. No attempt is made to serialize this read request with any concurrent update activity for the same record. The record returned may be a version updated by another RDO task but not yet committed, and this record could change if the update is subsequently backed out.

CONSISTENT
CICS reads the CSD with consistent read integrity. If a record is being modified by another RDO task, the READ request waits until the update is complete, the timing of which depends on whether the CSD is recoverable or non-recoverable:

- For a recoverable CSD, the READ request completes when the updating transaction completes its next syncpoint or rollback.
- For a non-recoverable CSD, the READ completes as soon as the VSAM request performing the update completes.

REPEATABLE
CICS reads the CSD with repeatable read integrity. If the record is being modified by another RDO task, the READ request waits until the update is complete, the timing of which depends on whether the CSD is recoverable or non-recoverable:

- For a recoverable CSD, the READ request completes when the updating transaction completes its next syncpoint or rollback.

- For a non-recoverable CSD, the READ completes as soon as the VSAM request performing the update completes.

After the CSD read completes, a shared lock remains held until syncpoint. This guarantees that a CSD record read within an RDO task cannot be modified until the end of the task (for example, a CEDA transaction) that is reading the CSD.

CSDJID

The **CSDJID** parameter specifies the journal identifier of the journal that you want CICS to use for automatic journaling of file requests against the CSD.

CSDJID={NO|number}

This parameter is effective only on a CICS cold or initial start. On a warm or emergency restart, file resource definitions for the CSD are recovered from the global catalog.

NO You do **not** want automatic journaling for the CSD. This is the default.

number

A number in the range 1 through 99 to identify the journal that CICS is to use for automatic journaling for the CSD. Mapping to a log stream works in the same way that CSDFRLOG does, that is, *nn* maps to DFHJnn. 01 no longer maps to the system log.

The automatic journaling options enforced for the CSD when you code CSDJID=number are JNLADD=BEFORE and JNLUPDATE=YES. These options are sufficient to record enough information for a user-written forward recovery utility. No other automatic journaling options are available for the CSD. For information about the options JNLADD=BEFORE and JNLUPDATE=YES, see FILE resources.

CSDLSRNO

The **CSDLSRNO** system initialization parameter specifies whether the CSD is to be associated with a local shared resource (LSR) pool.

CSDLSRNO={1|number|NONE|NO}

This parameter is effective only on a CICS cold or initial start. On a warm or emergency restart, file resource definitions for the CSD are recovered from the global catalog. However, you can redefine the LSR pool attribute for the CSD dynamically with an EXEC CICS SET FILE command.

1 The default LSR pool number is 1.

number

The number of the LSR pool the CSD is to be associated with. The number of the pool must be in the range 1 through 255.

NONE|NO

The CSD is not to be associated with a local shared resource pool.

CSDRECOV

The **CSDRECOV**system initialization parameter specifies whether the CSD is a recoverable file.

CSDRECOV={NONE|ALL|BACKOUTONLY}

The **CSDBKUP**, **CSDRECOV**, and **CSDFRLOG** system initialization parameters interact according to how they are specified, if CSDRLS=NO is specified. If

CSDRLS=YES is specified, these parameters are ignored, because the recovery attributes must be specified in the VSAM catalog (using the BWO, LOG, and LOGSTREAMID parameters on DEFINE CLUSTER or ALTER CLUSTER). If CSDRLS=NO is specified but LOG has been specified in the VSAM catalog, the recovery attributes are taken from the VSAM catalog, and CSDBKUP, CSDRECOV, and CSDFRLOG do not need to be specified. If they are specified, however, the rules given in Planning for backup and recovery must still be followed.

This parameter is effective only on a CICS cold or initial start. On a warm or emergency restart, file resource definitions for the CSD are recovered from the global catalog.

NONE
The CSD is not recoverable.

ALL You want both forward recovery and backout for the CSD. If you code ALL, also specify CSDFRLOG with the journal identification of the journal to be used for forward recovery of the CSD.

Note: If the journal you specify for logstreams associated with CSD recovery (CSDJID, CSDFRLOG, and possibly the log of logs, DFGLGLOG) is a DASD-only log stream, there can be delays when you use the CEDA transaction if the log stream requires a new connection. This delay is because the MVS system logger is formatting the staging data set. Symptoms of the problem are:

```
DFHLG0771 07/08/01 03:30:42 IYOT1 A temporary error condition occurred
during MVS logger operation IXGWRITE for logstream xxxxxx.yyyyyy.zzzzzz.
MVS logger codes: X'00000008', X'00000868'.
```

If the CSD is the only file using those logstreams, CICS disconnects from the log when you end the CEDA transaction. The next time you run a CEDA transaction, CICS reconnects to the log stream and the MVS system logger allocates and formats a new staging data set.

BACKOUTONLY
CSD recovery is limited to file backout only. If you specify backout for the CSD, CICS uses the system log to record before images for backout purposes.

CSDRLS

The **CSDRLS** system initialization parameter specifies whether CICS is to access the CSD in RLS mode.

CSDRLS={NO|YES}
Valid values are as follows:

NO The CSD is opened in non-RLS mode, as specified on the **CSDLSRNO** parameter.

YES The CSD is opened in RLS mode. This enables you to update the CSD concurrently from several CICS regions, provided all the regions specify CSDRLS=YES. If a CICS region opens the CSD in RLS mode, another CICS region cannot open it in non-RLS mode. The first CICS region to open the CSD in a sysplex with SMSVSAM determines the access mode for all regions.

Your CSD must be defined to support RLS access: the IMBED option must not be specified, and recovery attributes must be defined in the VSAM catalog.

Definitions required for VSAM RLS support explains the data set characteristics required to support RLS access. If your CSD does not meet these requirements, it will fail to open.

If you specify both RLS and local shared resource (CSDLSRNO=*number*), RLS takes precedence.

If you specify CSDRLS=YES, the **CSDRECOV**, **CSDFRLOG**, and **CSDJID** parameters are ignored. You must specify the recovery attributes for an RLS-mode CSD in the ICF catalog entry for the CSD.

Note: If you define a recoverable CSD for RLS-mode access, you have to quiesce all RLS activity against the CSD before you can update the CSD using the batch utility program, DFHCSDUP. You can use the **SET DSNAME QUIESCE** command to do this, to ensure that no CEDA, CEDB, or CEDC transactions can run until you unquiesce the data set on completion of the batch job.

CSDSTRNO

The **CSDSTRNO** system initialization parameter specifies the number of concurrent requests that can be processed against the CSD.

CSDSTRNO={6|number}

When the number of requests reaches the **CSDSTRNO** value, CICS automatically queues any additional requests until one of the active requests terminates.

CICS requires two strings per CSD user, and you can increase the **CSDSTRNO** value, in multiples of two, to allow more than one concurrent CEDA user.

See Multiple users of the CSD within a CICS region (non-RLS) before you code this parameter.

This parameter is effective only on a CICS cold or initial start. On a warm or emergency restart, file resource definitions for the CSD are recovered from the global catalog. However, you can redefine the number of strings for the CSD dynamically with an EXEC CICS SET FILE command.

6 The default number of concurrent requests for the CSD is 6.

number
This number must be a multiple of 2, in the range 2 through 254.

CWAKEY

The **CWAKEY** system initialization parameter specifies the storage key for the common work area (CWA) if you are operating CICS with storage protection (STGPROT=YES).

CWAKEY={USER|CICS}

You specify how much storage you want for the CWA on the **WRKAREA** parameter. The permitted values are USER (the default), or CICS:

USER CICS obtains storage for the CWA in user key. This allows a user program executing in any key to modify the CWA.

CICS CICS obtains storage for the CWA in CICS key. This means that only programs executing in CICS key can modify the CWA, and user-key programs have read-only access.

If CICS is running without storage protection, the CWAKEY parameter is ignored, and the CWA is always allocated from CICS-key storage.

DAE

The **DAE** system initialization parameter specifies the default DAE action when new system dump table entries are created.

DAE={NO|YES}

Valid values are as follows:

NO New system dump table entries will be created with DAEOPTION(NODAE). This means that the system dump will not be suppressed by the MVS Dump Analysis and Elimination (DAE) component.

YES New system dump table entries will be created with DAEOPTION(DAE). This means that the system dump is eligible for suppression by the MVS DAE component.

For more information about the DAEOPTION option, see SET SYSDUMPCODE.

DATFORM

The **DATFORM** system initialization parameter specifies the external date display standard that you want to use for CICS date displays.

DATFORM={MMDDYY|DDMMYY|YYMMDD}

An appropriate indicator setting is made in the CSA. It is examined by CICS supplied system service programs that display a Gregorian date. CICS maintains the date in the form 0CYYDDD in the CSA (where C=0 for years 19xx, 1 for years 20xx, and so on; YY=year of century; and DDD=day of year), and converts it to the standard you specify for display.

The DATFORM option selects the order in which the date is to be displayed. It does not select the format of the year. Both YY and YYYY formats are displayed.

MMDDYY
The date is in the form of month-day-year, MMDDYY and MMDDYYYY.

DDMMYY
The date is in the form of day-month-year, DDMMYY and DDMMYYYY.

YYMMDD
The date is in the form of year-month-day, YYMMDD and YYYYMMDD.

DB2CONN

The **DB2CONN** system initialization parameter specifies whether you want CICS to start the DB2 connection automatically during initialization.

DB2CONN={NO|YES}

Valid values are as follows:

NO Do not automatically invoke DFHD2CM0, the CICS DB2 attach program, during initialization.

YES Invoke the CICS DB2 attach program, DFHD2CM0, automatically

during CICS initialization. The other information CICS needs for starting the attachment is taken from CICS DB2 connection resource definitions installed from the CSD.

Specifying DB2CONN=YES is the recommended alternative to specifying the CICS DB2 attach programs in the CICS post-initialization program list table (PLT).

DBCTLCON

The **DBCTLCON** system initialization parameter specifies whether you want CICS to start the DBCTL connection automatically during initialization.

DBCTLCON={NO|YES}

Valid values are as follows:

NO Do not automatically invoke DFHDBCON, the CICS DBCTL attach program, during initialization.

YES Invoke the CICS DBCTL attach program, DFHDBCON, automatically during CICS initialization. The other information CICS needs for starting the attachment, such as the DRA startup table suffix or the DBCTL subsystem name, is taken from an INITPARM system initialization parameter.

Specifying DBCTLCON=YES means you don't need to define the DBCTL attach program in the CICS post-initialization program list table (PLT).

DEBUGTOOL

The **DEBUGTOOL** system initialization parameter specifies whether you want to use debugging profiles to select the programs that will run under the control of a debugging tool.

DEBUGTOOL={NO|YES}

The following debugging tools use debugging profiles:

- Debug Tool, for compiled language application programs (programs written in COBOL, PL/I, C, C++ and Assembler)
- Remote debugging tools (for compiled language application programs and Java™ programs)

Other debugging mechanisms, such as the CICS Execution Diagnostic Facility (CEDF) do not use debugging profiles.

NO Specifies that you do not want to use CICS debugging profiles to select the programs that will run under the control of a debugger tool.

YES
Specifies that you want to use CICS debugging profiles to select the programs that will run under the control of a debugging tool.

For more information, see Debugging profiles in Debugging .

DFLTUSER

The **DFLTUSER** system initialization parameter specifies the RACF userid of the default user; that is, the user whose security attributes are used to protect CICS resources in the absence of other, more specific, user identification.

DFLTUSER={CICSUSER|userid}
For example, except in the case of terminals defined with preset security, the security attributes of the default user are assigned to terminal users who do not sign on.

The specified userid must be defined to RACF if you are using external security (that is, you have specified the system initialization parameter SEC=YES).

The specified userid is signed on during CICS initialization. If it cannot be signed on, CICS fails to initialize.

Restrictions You can specify the DFLTUSER parameter in the SIT, PARM, or SYSIN only.

DIP

The **DIP** system initialization parameter specifies whether the batch data interchange program, DFHDIP, is to be included.

DIP={NO|YES}
This program supports the batch controller functions of the IBM® 3790 Communication System and the IBM 3770 Data Communication System. Support is provided for the transmit, print, message, user, and dump data sets of the 3790 system. For the effect of this parameter, see Defining CICS resource table and module keywords.

DISMACP

The **DISMACP** system initialization parameter specifies whether CICS is to disable any transaction that terminates abnormally with an ASRD or ASRE abend.

DISMACP={YES|NO}
DISMACP=YES has no effect if the ASRD or ASRE abend is handled by an active abend exit. An abend might be caused by a user program invoking a CICS macro, or referencing the CSA, or the TCA.

DOCCODEPAGE

The **DOCCODEPAGE** system initialization parameter specifies the default host code page to be used by the document domain.

DOCCODEPAGE={037|codepage}
The *codepage* is a field of up to 8 characters. If *codepage* value is not specified, the default *doccodepage* is set to *037*.

The standard CICS form of a host code page name consists of the code page number (or more generally CCSID) written using 3 to 5 decimal digits as necessary then padded with trailing spaces to 8 characters. For code page 37, which is fewer than 3 digits, the standard form is 037. CICS accepts any decimal number of up to 8 digits (padded with trailing spaces) in the range 1 to 65535 as a code page name, even if it is not in the standard form.

The DOCCODEPAGE parameter must specify an EBCDIC-based code page if any symbol processing is required, as the delimiters used for symbol and symbol list processing are assumed to be in EBCDIC.

DSALIM

The **DSALIM** system initialization parameter specifies the upper limit of the total amount of storage within which CICS can allocate the individual dynamic storage areas (DSAs) that reside in 24-bit storage (below 16 MB, also known as below the line).

DSALIM={5M|*number*}

Valid values are as follows:

5M The default **DSALIM** value is 5 MB (5,242,880 bytes).

number

A value in the range 2 MB to 16 MB (2,097,152 bytes to 16,777,216 bytes), in multiples of 256 KB (262,144 bytes). If the size specified is not a multiple of 256 KB, CICS rounds the value up to the next multiple.

You can specify *number* in bytes (for example, 4194304), a whole number of kilobytes (for example, 4096K), or a whole number of megabytes (for example, 4M). See Coding conventions for DSA limits in Improving performance

From the storage that you specify using **DSALIM**, CICS allocates the CDSA, UDSA, SDSA, and RDSA. CICS allocates the UDSA in multiples of 1 MB when transaction isolation is active, but in multiples of 256 KB in CICS regions without transaction isolation. The other DSAs in 24-bit storage are allocated in multiples of 256 KB, with or without transaction isolation. For information about the contents of each of the dynamic storage areas, see CICS dynamic storage areas.

The maximum value that you can specify for **DSALIM** is limited by the following factors:

- The configuration of your MVS storage, which governs how much private storage remains below the line.
- The amount of MVS storage, outside the DSAs, that you require to satisfy MVS GETMAIN requests for 24-bit storage (below the line).

For methods to estimate the amount of storage to specify on the **DSALIM** parameter, see Estimating, checking, and setting DSALIM .

Note: If you change the **DSALIM** value while CICS is running, the change is cataloged in the local catalog. If **DSALIM** is specified in the system initialization table, the cataloged value specified by the change overrides the value of the **DSALIM** system initialization parameter on an initial, cold, or warm start. The cataloged value is not used if you specify **DSALIM** as a system initialization parameter override (for example, in SYSIN), or if you reinitialize the CICS catalog data sets.

DSHIPIDL

The **DSHIPIDL** system initialization parameter specifies the minimum time, in hours, minutes, and seconds, that an inactive shipped terminal definition must remain installed in this region.

DSHIPIDL={020000|hhmmss}

When the timeout delete mechanism is invoked, only those shipped definitions that have been inactive for longer than the specified time are deleted.

You can use this parameter in a transaction routing environment, on the application-owning and intermediate regions, to prevent terminal definitions having to be reshipped because they have been deleted prematurely.

The default minimum idle time is 2 hours.

hhmmss
A 1- to 6-digit number in the range 0-995959. Numbers that have fewer than six digits are padded with leading zeros.

DSHIPINT

The **DSHIPINT** system initialization parameter specifies the interval between invocations of the timeout delete mechanism.

DSHIPINT={120000|0|hhmmss}
The timeout delete mechanism removes any shipped terminal definitions that have not been used for longer than the time specified by the DSHIPIDL parameter.

You can use this parameter in a transaction routing environment, on the application-owning and intermediate regions, to control:

- How often the timeout delete mechanism is invoked
- The approximate time of day at which a mass delete operation is to take place, relative to CICS startup

  Note: For more flexible control over when mass delete operations take place, you can use a CEMT SET DELETSHIPPED or EXEC CICS SET DELETSHIPPED command to reset the interval. (The revised interval starts *from the time the command is issued*, **not** from the time the remote delete mechanism was last invoked, nor from CICS startup.)

0
The timeout delete mechanism is not invoked. You might set this value in a terminal-owning region, or if you are not using shipped definitions.

hhmmss
A 1- to 6-digit number in the range 1-995959. Numbers that have fewer than six digits are padded with leading zeros.

DSRTPGM

The **DSRTPGM** system initialization parameter specifies the name of a distributed routing program. The distributed routing program must be specified in the routing region and each target CICS region.

DSRTPGM={NONE|DFHDSRP|*program-name*|EYU9XLOP}
The program can dynamically route:

- Eligible CICS business transaction services (BTS) processes and activities

  For information about which BTS processes and activities are eligible for dynamic routing, see Administering BTS.
- Eligible non-terminal-related EXEC CICS START requests.

  For information about which non-terminal-related START requests are eligible for dynamic routing, see Routing transactions invoked by START commands.

DFHDSRP
The CICS sample distributed routing program.

EYU9XLOP
The CICSPlex SM routing program.

NONE
For eligible CICS BTS processes and activities, no routing program is invoked. BTS processes and activities cannot be dynamically routed.

For eligible non-terminal-related START requests, the CICS sample distributed routing program, DFHDSRP, is invoked.

program-name
The name of a user-written program.

Note: See also the **DTRPGM** parameter, used to name the dynamic routing program.

DTRPGM

The **DTRPGM** system initialization parameter specifies the name of a dynamic routing program.

DTRPGM={DFHDYP|EYU9XLOP|*program-name*}
The program can dynamically route transactions initiated from user terminals, transactions initiated by eligible terminal-related EXEC CICS START commands, and eligible program-link requests. For information about which transactions started by EXEC CICS START commands, and which program-link requests, are eligible for dynamic routing, see the Introduction to CICS dynamic routing.

DFHDYP
The CICS-supplied routing program. This program is the default.

EYU9XLOP
The CICSPlex SM routing program.

program-name
The name of a user-written program.

Note: See also the **DSRTPGM** parameter, used to name the distributed routing program.

DTRTRAN

The **DTRTRAN** system initialization parameter specifies the name of the transaction definition that you want CICS to use for dynamic transaction routing.

DTRTRAN={CRTX|*name*|NO}
This is intended primarily for use in a CICS terminal-owning region, although you can also use it in an application-owning region when you want to daisy-chain transaction routing requests. In a dynamic transaction routing environment, the transaction defined for **DTRTRAN** must be installed in the CICS terminal-owning regions if you want to eliminate the need for resource definitions for individual transactions.

Note: DTRTRAN does not apply to non-terminal **EXEC CICS START** requests where the distributed routing program is invoked.

The transaction name is stored in the catalog for recovery during CICS restarts.

CRTX This is the default dynamic transaction definition. It is the name of the CICS-supplied sample transaction resource definition provided in the CSD group DFHISC.

name The name of your own dynamic transaction resource definition that you want CICS to use for dynamic transaction routing.

NO The dynamic transaction routing program is not invoked when a transaction definition cannot be found.

For information about the CICS-supplied sample transaction resource definition, CRTX, and about defining your own dynamic transaction routing definition, see Dynamic transaction routing.

DUMP

The **DUMP** system initialization parameter specifies whether the CICS dump domain is to take SDUMPs.

DUMP={YES|NO} (active and alternate)

Valid values are as follows:

YES SDUMPs are produced, unless suppressed by the options specified in the CICS system dump table or by the MVS system defaults.

NO SDUMPs are suppressed.

Note: This does not prevent the CICS kernel from taking SDUMPs.

For more information about SDUMPs, see System dumps.

DUMPDS

The **DUMPDS** system initialization parameter specifies the transaction dump data set that is to be opened during CICS initialization.

DUMPDS={AUTO|A|B}

Valid values are as follows:

AUTO

For all emergency or warm starts, CICS opens the transaction dump data set that was **not** in use when the previous CICS run terminated. This information is obtained from the CICS local catalog.

If you specify AUTO, or let it default, code DD statements for both of the transaction dump data sets, DFHDMPA and DFHDMPB, in your CICS startup job stream.

A CICS opens transaction dump data set DFHDMPA.

B CICS opens transaction dump data set DFHDMPB.

DUMPSW

The **DUMPSW** system initialization parameter specifies whether you want CICS to switch automatically to the next dump data set when the first is full.

DUMPSW={NO|NEXT}

Valid values are as follows:

NO Disables the CICS autoswitch facility. If the transaction dump data set opened during initialization becomes full, CICS issues a console

message to notify the operator. If you want to switch to the alternate data set, you must do so manually using the CEMT or EXEC CICS SET DUMPDS SWITCH command.

NEXT Enables the autoswitch facility to switch to the next data set at end of file of the data set opened during initialization. Coding NEXT permits one switch only. If you want to switch to the alternate data set again, you must do so manually using CEMT or EXEC CICS SET DUMPDS SWITCH command. If you specify NEXT, code DD statements for both of the transaction dump data sets, DFHDMPA and DFHDMPB, in your CICS startup job stream.

For more information about transaction dump data sets, see page Printing the transaction dump data sets.

DURETRY

The **DURETRY** system initialization parameter specifies, in seconds, the total time that CICS is to continue trying to obtain a system dump using the SDUMP macro.

DURETRY={30|number-of-seconds|0}

DURETRY allows you to control whether, and for how long, CICS is to reissue the SDUMP macro if another address space in the same MVS system is already taking an SDUMP when CICS issues an SDUMP request.

In the event of an SDUMP failure, CICS responds, depending on the reason for the failure, as follows:

- If MVS is already taking an SDUMP for another address space, and the DURETRY parameter is nonzero, CICS issues an MVS STIMERM macro to wait for five seconds, before retrying the SDUMP. CICS issues a message to say that it is waiting for five seconds before retrying the SDUMP. After five seconds CICS issues another message to say that it is retrying the SDUMP request.
- If the SDUMP fails for any other reason, such as no SYS1.DUMP data sets being available, or I/O errors preventing completion of the dump, CICS issues a message to inform you that the SDUMP has failed, and to give the reason why.

30 30 seconds allows CICS to retry up to 6 times (once every 5 seconds), if the cause of failure is that another region is taking an SDUMP.

number-of-seconds
Code the total number of seconds (up to 32767) during which you want CICS to continue retrying the SDUMP macro if the reason for failure is that another region is taking an SDUMP. CICS retries the SDUMP, once every five seconds, until successful or until retries have been made over a period equal to or greater than the DURETRY value.

0 Code a zero value if you do not want CICS to retry the SDUMP macro.

ECDSASZE

The **ECDSASZE** system initialization parameter specifies the size of the ECDSA.

ECDSASZE={0K|number}

The default size is 0 indicating that the DSA size can change dynamically. A non-zero value indicates that the DSA size is fixed.

number
Specify number as an amount of storage in the range 0 - 1073741824

bytes in multiples of 1048576 bytes (1 MB). If the size specified is not a multiple of 1 MB, CICS rounds the value up to the next multiple.

You can specify number in bytes (for example, 4194304), or as a whole number of KBs (for example, 4096 KB), or a whole number of megabytes (for example, 4 MB).

Restriction: You can specify the **ECDSASZE** parameter in PARM, SYSIN, or CONSOLE only.

Important: Setting the size of individual dynamic storage areas (DSAs) is not usually necessary and is not recommended. If you specify DSA size values that in combination do not allow sufficient space for the remaining DSAs, CICS fails to initialize. The limit on the storage available for the DSAs in 31-bit storage (above the line) is specified by the EDSALIM system initialization parameter. You must allow at least 1 MB for each DSA in 31-bit storage for which you have not set a size. See DSA size limits.

EDSALIM

The **EDSALIM** system initialization parameter specifies the upper limit of the total amount of storage within which CICS can allocate the individual extended dynamic storage areas (EDSAs) that reside in 31-bit (above-the-line) storage; that is, above 16 MB but below 2 GB.

EDSALIM={800M|*number*}

800M The default **EDSALIM** value is 800 MB (838,860,800 bytes).

number

A value in the range 64 MB through 2047 MB, in multiples of 1 MB. If the size specified is not a multiple of 1 MB, CICS rounds the value up to the next multiple.

You can specify *number* in bytes (for example, 52428800), a whole number of kilobytes (for example, 51200K), or a whole number of megabytes (for example, 50M). See Coding conventions for DSA limits in Improving performance.

From the storage that you specify using **EDSALIM**, CICS allocates the ECDSA, EUDSA, ESDSA, ERDSA, and ETDSA. For information about the contents of each of the extended dynamic storage areas, see CICS dynamic storage areas.

The maximum value that you can specify for **EDSALIM** is limited by the following factors:

- The size that you have specified for the CICS region on the MVS **REGION** parameter in the CICS job or procedure.
- The amount of MVS storage, outside the EDSAs, that you require to satisfy MVS GETMAIN requests for 31-bit (above-the-line) storage.

For methods to estimate the amount of storage to specify on the **EDSALIM** parameter, see Estimating, checking, and setting EDSALIM in Improving performance.

Note: If you change the EDSA limit while CICS is running, the change is cataloged in the local catalog. If **EDSALIM** is specified in the system initialization table, the cataloged value specified by the change overrides the value of the **EDSALIM** system initialization parameter on an initial, cold, or warm start. The

cataloged value is not used if you specify **EDSALIM** as a system initialization parameter override (for example, in SYSIN), or if you reinitialize the CICS catalog data sets.

ENCRYPTION (DEPRECATED)

The **ENCRYPTION** system initialization parameter specifies the cipher suites that CICS uses for secure TCP/IP connections.

ENCRYPTION={ALL|TLS12|STRONG}

When a secure connection is established between a pair of processes, the most secure cipher suite that is supported by both is used.

The **ENCRYPTION** system initialization parameter has been deprecated. Use the **MINTLSLEVEL** system initialization parameter instead. For more information about the **MINTLSLEVEL** system initialization parameter, see "MINTLSLEVEL" on page 65. If you specify the **ENCRYPTION** parameter, it will be treated as **MINTLSLEVEL**:

- ENCRYPTION=STRONG will be treated as MINTLSLEVEL=TLS10. This is the default.

  Note: A change in behavior might occur if the client does not handle the negotiation of TLS levels correctly. If this causes a problem, delete the **ENCRYPTION** parameter and use MINTLSLEVEL=TLS10ONLY instead.
- ENCRYPTION=ALL will be treated as MINTLSLEVEL=TLS11.

  Note: A change in behavior might occur if the client supports only TLS 1.0. If this causes a problem, delete the **ENCRYPTION** parameter and use MINTLSLEVEL=TLS10 instead.
- ENCRYPTION=TLS12 will be treated as MINTLSLEVEL=TLS12

ALL Allows the use of TLS version 1.1 and 1.2 in addition to the protocols allowed by STRONG.

TLS12 Sets the level of TLS to 1.2.

STRONG
Sets the level of TLS to 1.0 only. This is the default value.

For more information about cipher suites, see Cipher suites and cipher suite specification files.

CICS can use only the cipher suites that are supported by the underlying z/OS operating system.

To apply FIPS 140-2 standards, set **ENCRYPTION**=TLS12 and **NISTSP800131A**=CHECK. If **NISTSP800131A**=CHECK is set but **ENCRYPTION** is set to a value other than TLS12, it is overridden to **ENCRYPTION**=TLS12 and a warning message is issued.

To apply FIPS 140-2 standards on z/OS Version 2 Release 1 or later, ICSF (Integrated Cryptographic Services Facility) must be active on your system. If you have not already done so, apply APAR OA14956 to z/OS.

For more information about NIST SP800-131A conformance, see Making your CICS TS system conformant to NIST SP800-131A.

EODI

The **EODI** system initialization parameter specifies the end-of-data indicator for input from sequential devices.

EODI={E0|xx}
The characters “xx” represent two hexadecimal digits in the range 01 through FF. The default value is X'E0', which represents the standard EBCDIC backslash symbol (\).

ERDSASZE

The **ERDSASZE** system initialization parameter specifies the size of the ERDSA.

ERDSASZE={0K|number}
The default size is 0 indicating that the DSA size can change dynamically. A non-zero value indicates that the DSA size is fixed.

number
Specify number as an amount of storage in the range 0 to 1073741824 bytes in multiples of 1048576 bytes (1 MB). If the size specified is not a multiple of 1 MB, CICS rounds the value up to the next multiple.

You can specify number in bytes (for example, 4194304), or as a whole number of kilobytes (for example, 4096 KB), or a whole number of megabytes (for example, 4 MB).

Restriction: You can specify the **ERDSAZSE** parameter in PARM, SYSIN, or CONSOLE only.

Important: Setting the size of individual dynamic storage areas (DSAs) is not usually necessary and is not recommended. If you specify DSA size values that in combination do not allow sufficient space for the remaining DSAs, CICS fails to initialize. The limit on the storage available for the DSAs in 31-bit storage (above the line) is specified by the EDSALIM system initialization parameter. You must allow at least 1 MB for each DSA in 31-bit storage for which you have not set a size. See DSA size limits.

ESDSASZE

The **ESDSASZE** system initialization parameter specifies the size of the ESDSA.

ESDSASZE={0K|number}
The default size is 0 indicating that the DSA size can change dynamically. A non-zero value indicates that the DSA size is fixed.

number
Specify number as an amount of storage in the range 0 to 1073741824 bytes in multiples of 1048576 bytes (1 MB). If the size specified is not a multiple of 1 MB, CICS rounds the value up to the next multiple.

You can specify number in bytes (for example, 4194304), or as a whole number of kilobytes (for example, 4096 KB), or a whole number of megabytes (for example, 4 MB).

Restriction: You can specify the **ESDSASZE** parameter in PARM, SYSIN, or CONSOLE only.

Important: Setting the size of individual dynamic storage areas (DSAs) is not usually necessary and is not recommended. If you specify DSA size values that in combination do not allow sufficient space for the remaining DSAs, CICS fails to initialize. The limit on the storage available for the DSAs in 31-bit storage (above the line) is specified by the EDSALIM system initialization

parameter. You must allow at least 1 MB for each DSA in 31-bit storage for which you have not set a size. See DSA size limits.

ESMEXITS

The **ESMEXITS** system initialization parameter specifies whether installation data is to be passed through the RACROUTE interface to the external security manager (ESM) for use in exits written for the ESM.

ESMEXITS={NOINSTLN|INSTLN}

Valid values are as follows:

NOINSTLN

The INSTLN parameter is not used in RACROUTE macros.

INSTLN

CICS-related and installation-supplied data is passed to the ESM using the INSTLN parameter of the RACROUTE macro. For programming information, including the format of the data passed, see How ESM exit programs access CICS-related information. This data is intended for use in exits written for the ESM.

Restrictions You can specify the ESMEXITS parameter in the SIT only.

EUDSASZE

The **EUDSASZE** system initialization parameter specifies the size of the EUDSA.

EUDSASZE={0K|number}

The default size is 0 indicating that the DSA size can change dynamically. A non-zero value indicates that the DSA size is fixed.

number

Specify number as an amount of storage in the range 0 to 1073741824 bytes in multiples of 1048576 bytes (1MB). If the size specified is not a multiple of 1MB, CICS rounds the value up to the next multiple.

You can specify number in bytes (for example, 4194304), or as a whole number of kilobytes (for example, 4096K), or a whole number of megabytes (for example, 4M).

Restriction: You can specify the EUDSAZSE parameter in PARM, SYSIN, or CONSOLE only.

Important: Setting the size of individual dynamic storage areas (DSAs) is not usually necessary and is not recommended. If you specify DSA size values that in combination do not allow sufficient space for the remaining DSAs, CICS fails to initialize. The limit on the storage available for the DSAs in 31-bit storage (above the line) is specified by the EDSALIM system initialization parameter. You must allow at least 1 MB for each DSA in 31-bit storage for which you have not set a size. See DSA size limits.

FCT

The **FCT** system initialization parameter specifies the suffix of the file control table to be used.

FCT={NO|xx|YES}
This parameter is effective only on a CICS cold or initial start. CICS does not load an FCT on a warm or emergency restart and all file resource definitions are recovered from the global catalog.

For information about coding the macros for this table, see File control table (FCT).

You can use a mixture of macro definitions and RDO definitions for files in your CICS region. However, your FCT should contain definitions for only BDAM files to be loaded on a CICS cold start. Other types of files are loaded from their file definitions in RDO groups specified in the **GRPLIST** system initialization parameter. Any definitions in the FCT other than for BDAM files are ignored.

FCQRONLY

The **FCQRONLY** system initialization parameter specifies whether you want CICS to force all file control requests to run under the CICS QR TCB. This parameter applies to file control requests that access VSAM RLS files and local VSAM LSR files. Requests for all other file types always run on the QR TCB.

FCQRONLY={YES|NO}
Valid values are as follows:

NO File Control requests are treated as threadsafe and are run on an open TCB to avoid unnecessary TCB switching. For CONCURRENCY(REQUIRED) programs the request runs on an open TCB. For CONCURRENCY(THREADSAFE) programs the request runs on whatever TCB is being used at the time of the request.

YES File control requests are treated as non-threadsafe. CICS forces all file control requests to run under the CICS QR TCB. With all file requests on the QR TCB, CICS can minimize the amount of locking required at the expense of additional TCB switches if requests are run on open TCBs. YES is the default.

For a program defined as CONCURRENCY(REQUIRED), if the file control request is run under the CICS QR TCB, CICS switches back to the open TCB before handing control back to the application program.

For file-owning regions (FORs), choose an appropriate setting for **FCQRONLY**:

- For FORs where the connections to that region are primarily MRO or ISC connections, these requests run on the QR TCB, and CICS runs the mirror program primarily on the QR TCB. Specify **FCQRONLY=YES** so that all file control requests are processed on the QR TCB. This setting improves performance by avoiding locking, which is unnecessary when all file control requests run on the same TCB.
- For FORs where the connections to that region are primarily IPIC connections, these requests run on open TCBs, and CICS runs the mirror program on an L8 open TCB whenever possible. Specify **FCQRONLY=NO** so that file control requests do not switch to the QR TCB to be processed. This setting improves performance by multi-threading file control requests.

FEPI

The **FEPI** system initialization parameter specifies whether or not you want to use the Front End Programming Interface feature (FEPI).

FEPI={NO|YES}

Valid values are as follows:

NO FEPI support is not required. You should specify NO on this parameter (or allow it to default) if you do not have the feature installed, or if you do not require FEPI support.

YES You require FEPI support, and CICS is to start the CSZI transaction.

For information about what is required to run FEPI see FEPI analysis and planning.

FLDSEP

The **FLDSEP** system initialization parameter specifies one through four field-separator characters, each of which indicates end of field in the terminal input data.

FLDSEP={' '|'xxxx' }

The default is four blanks. The field separator allows you to use transaction identifications of less than four characters followed by one of the separator characters. When less than four characters are coded, the parameter is padded with blanks, so that the blank is then a field separator. None of the specified field separator characters should be part of a transaction identification; in particular, the use of alphabetic characters as field separators is not recommended.

The character specified in the **FLDSEP** parameter must not be the same as any character specified in the **FLDSTRT** parameter. This means that it is invalid to allow both parameters to take the default value. **Restrictions**

If you specify **FLDSEP** in the SIT, the characters must be enclosed in single quotation marks.

If you specify **FLDSEP** as a PARM, SYSIN, or CONSOLE parameter, do **not** enclose the characters in quotation marks, and the characters you choose must not include an embedded blank, or any of these characters:

( ) ' = ,

FLDSTRT

The **FLDSTRT** system initialization parameter specifies a single character to be the field-name-start character for free-form input for built-in functions.

FLDSTRT={' '|'x'}

The default is a blank. The character specified should not be part of a transaction identification; in particular, the use of alphabetic characters is not recommended.

The character specified in the FLDSTRT parameter must not be the same as any character specified in the FLDSEP parameter. This means that it is invalid to allow both parameters to take the default value.

Restrictions

If you specify FLDSTRT in the SIT, the parameter must be enclosed in single quotation marks.

If you specify FLDSTRT as a PARM, SYSIN, or CONSOLE parameter, do **not** enclose the character in quotation marks, and the character you choose must not be a blank or any of the following characters:

```
( ) ' = ,
```

FORCEQR

The **FORCEQR** system initialization parameter specifies whether you want CICS to force all CICS API user application programs that are specified as threadsafe to run under the CICS QR TCB, as if they were specified as quasi-reentrant programs.

FORCEQR={NO|YES}

This parameter applies to all application programs that are restricted to the current CICS programming interfaces (that is, programs that specify API(CICSAPI)), and does not apply to any of the following programs:

- Java programs that are run in a JVM
- C/C++ programs using XPLINK
- OPENAPI programs
- Programs defined with CONCURRENCY(REQUIRED)

None of these programs can run on the QR TCB.

NO CICS honors the CONCURRENCY(THREADSAFE) attribute on program resource definitions, and allows user application programs to run on an open TCB to avoid unnecessary TCB switching.

YES CICS forces all CICSAPI user application programs specified with the CONCURRENCY(THREADSAFE) attribute to run under the CICS QR TCB, as if they were specified as CONCURRENCY(QUASIRENT) programs

FORCEQR=YES allows you, in a test environment, to run incompletely tested threadsafe application programs that have proved to be non-threadsafe.

The **FORCEQR** parameter applies to all programs defined as threadsafe that are not invoked as task-related user exits, global user exits, or user-replaceable modules.

FSSTAFF

The **FSSTAFF** system initialization parameter prevents transactions initiated by function-shipped **EXEC CICS START** requests being started against incorrect terminals.

FSSTAFF={YES|NO}

Specify this parameter in an application-owning region (AOR). You might need to code the function-shipped START affinity (FSSTAFF) parameter in an AOR if all of the following are true:

1. The AOR is connected to two or more terminal-owning regions (TORs) that use the same, or a similar, set of terminal identifiers.
2. One or more of the TORs issues EXEC CICS START requests for transactions in the AOR.
3. The START requests are associated with terminals.
4. You are using shippable terminals, rather than statically defining remote terminals in the AOR.

Consider the following scenario:

Terminal-owning region TOR1 issues an EXEC CICS START request for transaction TRAR, which is owned by region AOR1. It is to be run against terminal T001. Meanwhile, terminal T001 on region TOR2 has been transaction routing to AOR1; a definition of T001 has been shipped to AOR1 from TOR2.

When the START request arrives at AOR1, it is shipped to TOR2, rather than TOR1, for transaction routing from terminal T001.

To prevent this situation, code YES on the FSSTAFF parameter in the AOR.

YES When a START request is received from a terminal-owning region, and a shipped definition for the terminal named on the request is already installed in the AOR, the request is always shipped back to a TOR, for routing, *across the link it was received on*, irrespective of the TOR referenced in the remote terminal definition.

If the TOR to which the START request is returned is **not** the one referenced in the installed remote terminal definition, a definition of the terminal is shipped to the AOR, and the autoinstall user program is called. Your autoinstall user program can then allocate an *alias* termid in the AOR, to avoid a conflict with the previously installed remote definition. For information about writing an autoinstall program to control the installation of shipped definitions, see Writing a program to control autoinstall of shipped terminals.

NO When a START request is received from a terminal-owning region, and a shipped definition for the named terminal is already installed in the AOR, the request is shipped to the TOR referenced in the definition, for routing.

Note:

1. FSSTAFF has no effect:
   - On statically-defined (hard-coded) remote terminal definitions in the AOR. If you use these, START requests are always shipped to the TORs referenced in the definitions.
   - On START requests issued in the local region. It affects only START requests shipped from other regions.
   - When coded on intermediate regions in a transaction-routing path. It is effective only when coded on an application-owning region.
2. If the AOR contains no remote definition of a terminal named on a shipped START request, the "terminal not known" global user exits, XICTENF and XALTENF, are called. For details of these exits, see .

FTIMEOUT

The **FTIMEOUT** system initialization parameter specifies a timeout interval for requests made on files that are opened in RLS mode.

FTIMEOUT={30|number}

The timeout interval is in seconds, from 1 through 4080 (sixty eight minutes) and indicates how long VSAM should wait before terminating a request and returning an exception condition.

The default is 30 seconds.

FTIMEOUT applies to transactions that do not have a deadlock timeout interval active. If a time value is specified for the DTIMOUT keyword of the TRANSACTION definition, this value is used as the file timeout value for that transaction.

GMTEXT

The **GMTEXT** system initialization parameter specifies whether the default logon message text (WELCOME TO CICS) or your own message text is to be displayed on the screen.

GMTEXT={'DFHZC2312 * WELCOME TO CICS ***'|'text'}**

The message text can be displayed by the CSGM (good morning) transaction when a terminal is logged on to CICS through z/OS Communications Server, by the CESN transaction if used to sign on to CICS, or by your own transactions using the **EXEC CICS INQUIRE SYSTEM GMMTEXT** command.

You can use apostrophes to punctuate your message, in addition to using them as message delimiters. However, you must code *two* successive apostrophes to represent a single apostrophe in your text. For example,

```
GMTEXT='User''s logon message text.'
```

The whole message must still be enclosed by a pair of single delimiting apostrophes.

Your message text can be from 1 through 246 characters (bytes), and can extend over two lines by extending the text to column 80 on the first line, and continuing in column 1 of the second line. For example, the following might be used in the SYSIN data set:

```
*          CICS Transaction Server for z/OS, Version 5 Release 4  SYSTEM    *
GMTEXT='An Information Development CICS Terminal-Owning Region (TOR) - C
ICSIDC. This message is to show the use of continuation lines when creating a GM
TEXT parameter in the SYSIN data set'  (for first signon
```

The CSGM transaction displays this as follows (with the time appended to the end of message):

```
An Information Development CICS Terminal-Owning Region (TOR) - C
ICSIDC. This message is to show the use of continuation lines when creating a GM
TEXT parameter in the SYSIN data set 09:56:14
```

The CESN transaction displays this as follows:

```
 Signon for CICS Transaction Server for z/OS, Version 5 Release 4  APPLID CICSHTH1

An Information Development CICS Terminal-Owning Region (TOR) - CICSIDC.
This message is to show the use of continuation lines when creating a GMTEXT
parameter in the SYSIN data set
```

For any transaction other than CESN that displays the text specified by this parameter, you must use a TYPETERM with LOGONMSG(YES) for all terminals requiring the logon message. For information about using TYPETERM, see Autoinstalling model terminal definitions.

GMTRAN

The **GMTRAN** system initialization parameter specifies the name of a transaction ID.

GMTRAN={CSGM|CESN|transaction-id}

Specify the name of the transaction that is:

1. Initiated by ATI when terminals are logged on to CICS by z/OS Communications Server, and LOGONMSG(YES) is specified in the TYPETERM definition.
2. Set to be the next transaction initiated by the terminal operator following expiry of the terminal user's TIMEOUT period (specified in the External Security Manager) and either:
   - LOGONMSG(YES) and SIGNOFF(YES)

     or
   - LOGONMSG(YES), SIGNOFF(LOGOFF) and DISCREQ(NO)

   is specified in the TYPETERM definition.

GMTRAN is initiated when terminals are logged on to CICS by z/OS Communications Server. Do not specify the name of a remote transaction. The transaction must be capable of being automatically initiated (ATI). The default is the transaction CSGM, that displays the text specified in the GMTEXT parameter. Alternatively, you can specify one of the CICS signon transactions, CESL or CESN, which also display the text specified in the GMTEXT parameter. The GMTRAN parameter can be used with the LGNMSG parameter to retrieve z/OS Communications Server logon data.

GNTRAN

The **GNTRAN** system initialization parameter specifies the transaction that you want CICS to invoke when a user's terminal-timeout period expires.

GNTRAN={NO|transaction_id}

Valid values are as follows:

NO The default value, NO, specifies that no special transaction is to be executed when the timeout period expires. Instead, the user is signed off (subject to the SIGNOFF attribute of the TYPETERM resource definition for the terminal, as described below). After the signoff, if the LOGONMSG(YES) option is specified in the TYPETERM resource definition for the terminal, the transaction specified in the **GMTRAN** system initialization parameter is executed.

transaction_id

The name of a timeout transaction to signoff the user at the timed-out terminal. You can specify CESF as the timeout transaction. Specifying your own transaction allows you to specify functions in addition to, or instead of, signoff. For example, your own transaction could issue a prompt for the terminal user's password, and allow the session to continue if the correct password is entered.

The transaction to be used must have been specially written to handle the GNTRAN COMMAREA that is passed to it. Of the CICS-supplied transactions, only CESF has been written to handle the GNTRAN COMMAREA. For more information about writing your own transactions for GNTRAN, see Writing a good night program.

Note: When either the CICS CESF transaction, or your own transaction, attempts to sign off a terminal, the result is subject to the SIGNOFF attribute of the TYPETERM resource definition for the terminal, as follows:

SIGNOFF	Effect
YES	The terminal is signed off, but not logged off.

NO The terminal remains signed on and logged on.
LOGOFF
The terminal is both signed off and logged off.

Note: If GNTRAN fails to attach, and SIGNOFF(LOGOFF) has been specified, the terminal which has reached timeout will be signed off and logged off. GNTRAN will not run and will have no effect.

GRNAME

The **GRNAME** system initialization parameter specifies the z/OS Communications Server generic resource name, as 1 through 8 characters, under which a group of CICS terminal-owning regions in a CICSplex register to z/OS Communications Server.

GRNAME=name
There is no default for the **GRNAME** parameter. If you do not specify a value for this parameter, CICS does not register itself with the z/OS Communications Server generic resources function.

The following restrictions apply when specifying the **GRNAME** parameter:

1. If you are operating a CICSPlex that comprises separate terminal-owning regions and application-owning regions, ensure that you define a z/OS Communications Server generic resource name to the CICS terminal-owning regions only.
2. Generic resource names must be unique within a single network. A generic resource cannot be identical to:
   - A USERVAR
   - An alias name
   - A real LU name

   It is the responsibility of the user to see that these rules are kept.
3. The first character of the **GRNAME** parameter value cannot be a number.

For example, a CICS region with the system initialization parameters:

```
APPLID=CICSHTH1
GRNAME=CICSH###
```

would register to z/OS Communications Server with the applid CICSHTH1 and the generic resource CICSH###. Other LUs in the same sysplex can communicate with the CICS region either through the generic resource or the applid.

Note: If **GRNAME** is specified and you are signing on using a PassTicket, then the PassTicket must be generated using **GRNAME** and not the generic applid.

However, care should be taken with LU6 connections initiated from this side (such as AUTOCONNECT(YES)) because the bind will now contain the generic resource name and may fail if the partner only knows this region by the applid. Binds initiated from the partner are examined to identify the name by which the partner knows this region (generic resource or applid), thus allowing the appropriate connection to be built. For information about defining connections, see Configuring z/OS Communications Server generic resources in Configuring.

GRPLIST

The **GRPLIST** system initialization parameter specifies the names (each 1 - 8 characters) of up to four lists of resource definition groups on the CICS system definition (CSD) file.

GRPLIST={DFHLIST |name|(name[,name2][,name3][,name4])}

The resource definitions in all the groups in the specified lists are loaded during initialization when CICS performs a cold start. If a warm or emergency start is performed, the resource definitions are derived from the global catalog, and the **GRPLIST** parameter is ignored.

Each name can be either a real group list name or a generic group list name that incorporates global filename characters (+ and *). If you specify more than one group list (either by specifically coding two or more group list names or by coding a group list name with global filename characters), the later group lists are concatenated onto the first group list. Any duplicate resource definitions in later group lists override those in earlier group lists.

Use the CEDA command LOCK to protect the lists of resource groups specified on the GRPLIST parameter.

The default is DFHLIST, the CICS-supplied list that specifies the set of resource definitions needed by CICS. If you create your own group list, either add to it the groups specified in DFHLIST (omitting only those for CICS functions that you know you do not need) or specify the DFHLIST name on the **GRPLIST** parameter. Do not code GRPLIST=NO unless you have a group list named NO.

Note:

1. Group lists specified by a generic group list name are concatenated in alphabetic, then numeric, order. For example, the generic list name CICSHT* would concatenate the group lists CICSHT#1, CICSHTAP, CICSHTSD, and CICSHT3V in that order. If the order of concatenation is important (for example, to ensure that a particular resource definition overrides another), consider coding real group list names.
2. If a group list contains resource definitions that are needed by another group list, the group list containing those definitions must be installed first. For example, if list A has TYPETERM definitions needed for TERMINAL definitions in list B, list A must be installed first. Therefore, you might need to specifically name the prerequisite group on the **GRPLIST** parameter.
3. Take care when using generic group list names, because if a group list on your CSD satisfies the generic name, it will be installed. This means that a group list can be installed more than once; for example, if you specify the real group list name and a generic group list name that it satisfies, or if you specify two generic group list names that the group list name satisfies.
4. To override one or more of the group lists specified on the **GRPLIST** system initialization parameter, you must specify all list names (both real and generic) that you want to use, even if you are not changing the names.

For example, to use the four group lists CICSHT#1, CICSHTAP, CICSHTSD, and CICSHT3V, you could specify either of the following system initialization parameters:

```
GRPLIST=(CICSHT*)
GRPLIST=(CICSHT#1,CICSHTAP,CICSHT3V,CICSHTSD)
```

In the first example, the group lists are loaded in the order CICSHT#1, CICSHTAP, CICSHTSD, then CICSHT3V. Resource definitions installed from the CICSHT3V group list override any duplicate definitions installed by the other groups.

In the second example, the group lists are loaded in the order specified. Resource definitions installed from the CICSHTSD group list override any duplicate definitions installed by the other groups.

If you specify `GRPLIST=(CICSHT#1,CICSAP*,CICSHT3V,CICSHTSD)` and you want to replace the list CICSHT3V with the list ANOLST05, specify the override:

```
GRPLIST=(CICSHT#1,CICSAP*,ANOLST05,CICSHTSD)
```

In general, any required resource definitions should appear in one of the group lists specified on the **GRPLIST** system initialization parameter.

For information about resource definitions, groups, lists, and the CSD, see CICS resources.

GTFTR

The **GTFTR** system initialization parameter specifies whether CICS can use the MVS generalized trace facility (GTF) as a destination for trace data.

GTFTR={OFF|ON}

This parameter controls whether any of the three types of CICS trace entry are written to GTF data sets. The three types are: CICS system trace (see the SYSTR parameter), user trace (see the USERTR parameter), and exception trace entries (which are always made and not controlled by a system initialization parameter).

OFF CICS does not use GTF as a destination for CICS trace data.

ON CICS uses GTF as a destination for CICS trace data. To use the GTF data sets for CICS trace data, you must have started GTF with the USR option, in addition to coding GTFTR=ON.

For information about GTF, see the *z/OS MVS Diagnosis: Tools and Service Aids* manual.

HPO

The **HPO** system initialization parameter specifies whether you want to use the z/OS Communications Server authorized path feature of the high performance option (HPO).

HPO={NO|YES}

If you code YES, the CICS type 6 SVC must be link-edited in your MVS nucleus, and defined to MVS in an SVCPARM statement. If the SVC number is not 215 (the default) you must specify the SVC number on the **SRBSVC** parameter.

If HPO=YES and HPO activation is successful, message DFHSI1600 is issued.

If HPO=YES and HPO activation is unsuccessful, message DFHSI1601 is issued and CICS terminates.

For information about installing the CICS type 6 SVC in your MVS system, and about changing the default number, see Selecting the high-performance option.

Restrictions You can specify the HPO parameter in the SIT only.

HTTPSERVERHDR

The **HTTPSERVERHDR** system initialization parameter specifies the value (up to 64 characters) that CICS sets in the server header of HTTP responses.

HTTPSERVERHDR={YES|NO|'value'}

Valid values are as follows:

YES The server header will be set to *IBM_CICS_Transaction_Server/5.3.0(zOS)* in outgoing HTTP responses.

NO CICS will not add a server header to outgoing responses.

value When you specify a value, the server header will include this value in outgoing HTTP responses.

HTTPUSRAGENTHDR

The **HTTPUSRAGENTHDR** system initialization parameter specifies the value (up to 64 characters) that CICS sets in the user-agent header of HTTP requests.

HTTPUSRAGENTHDR={YES|NO|'value'}

Valid values are as follows:

YES The user-agent header will be set to *IBM_CICS_Transaction_Server/5.3.0(zOS)* in outgoing HTTP requests.

NO CICS will not add a user-agent header to outgoing responses.

value When you specify a value, the user-agent header will include this value in outgoing HTTP requests.

ICP

The **ICP** system initialization parameter specifies that you want to perform a cold start for interval control program.

ICP=COLD

If COLD is not specified, the ICP start type will be determined by the **START** and **TS** parameter values. If TS=COLD and START=AUTO is specified then Interval Control Elements created without FROM data will be restored on a WARM start. See Defining CICS resource table and module keywords for further information.

ICV

The **ICV** system initialization parameter specifies the region exit time interval in milliseconds.

ICV={1000|number}

The **ICV** system initialization parameter specifies the maximum time in milliseconds that CICS releases control to the operating system when there are no transactions ready to resume processing. This time interval can be any integer in the range 100 through 3600000 milliseconds (specifying an interval up to 60 minutes). A typical range of operation might be 100 through 2000 milliseconds.

A low value interval can enable much of the CICS nucleus to be retained in dynamic storage, and not be paged-out at times of low terminal activity. This reduces the amount of dynamic storage paging necessary for CICS to process

terminal transactions (thus representing a potential reduction in response time), sometimes at the expense of concurrent batch region throughput.

Large networks with high terminal activity are inclined to run CICS without a need for this value, except to handle the occasional, but unpredictable, period of inactivity. These networks can usually function with a large interval (10000 to 3600000 milliseconds). After a task is initiated, the system recognizes its requests for terminal services and the completion of the services, and this maximum delay interval is overridden.

Small systems, or those with low terminal activity, are subject to paging introduced by other jobs running in competition with CICS. By specifying a low value interval, key portions of the CICS nucleus are referenced more frequently, thus reducing the probability of these pages being paged-out. However, the execution of the logic without performing productive work might be considered wasteful. The need to increase the probability of residency by frequent but unproductive referencing must be weighed against the overhead and response time degradation incurred by allowing the paging to occur. By increasing the interval size, less unproductive work is performed at the expense of performance if paging occurs during the periods of CICS activity.

For information about interval control parameters and performance, see Interval control value parameters: ICV, ICVR, and ICVTSD.

Note: The region exit time interval process contains a mechanism to ensure that CICS does not constantly set and cancel timers (thus degrading performance) while attempting to meet its objectives for a low region exit time interval. This mechanism can cause CICS to release control to the operating system for up to 0.5 seconds when the interval has been set at less than 250; and up to 0.25 seconds more than the region exit time interval when the interval has been set greater than 250.

ICVR

The **ICVR** system initialization parameter specifies the default runaway task time interval in milliseconds as a decimal number.

ICVR={2000|number}

You can specify zero, or a number in the range 250 through 2 700 000, in multiples of 250. CICS rounds down values that are not multiples of 250. This is the RUNAWAY interval used by transactions defined with RUNAWAY=SYSTEM. See TRANSACTION attributes for further information.

CICS might purge a task if it has not given up control after the RUNAWAY interval for the transaction (or ICVR if the transaction definition specified RUNAWAY=SYSTEM). If you code ICVR=0, runaway task control is inoperative for transactions specifying RUNAWAY=SYSTEM in their transaction definition (that is, tasks do not get purged if they appear to be looping). The ICVR value is independent of the ICV value, and can be less than the ICV value. CICS runaway task detection is based upon task time, that is, the interval is decremented only when the task has control of the processor.

Important: Java uses a modified ICVR value. For more information, see CICS task and thread management.

For information about commands that reinitialize the ICVR value, see Investigating loops that cause transactions to abend with abend code AICA.

ICVTSD

The **ICVTSD** system initialization parameter specifies the terminal scan delay value.

ICVTSD={0|number}

The terminal scan delay facility was used in earlier releases to limit how quickly CICS dealt with some types of terminal output requests made by applications, in order to spread the overhead for dealing with the requests. The range is 0 through 5000 milliseconds. Specifying a nonzero value was sometimes appropriate where the CICS system used non-SNA networks. However, with SNA and IPIC networks, setting ICVTSD to 0 is appropriate to provide a better response time and best virtual storage usage.

INFOCENTER

The **INFOCENTER** system initialization parameter specifies the location of the IBM Knowledge Center. You can point either to the online IBM-hosted Knowledge Center or a local installation. If you add this parameter to the Web User Interface (WUI) CICS startup JCL, a link labelled "Information Center" is displayed on WUI views and menus. If you do not code this parameter, CICS does not construct links to the IBM Knowledge Center.

INFOCENTER= *local_path*:*URL*

To point to a local installation of IBM Knowledge Center, specify the path to the local IBM Knowledge Center. This is the server name and port number of the local host, followed by /kc. For example: `INFOCENTER=http://myserver:29127/kc.`

To point to the online IBM Knowledge Center, specify its URL. For example: `INFOCENTER=http://www.ibm.com/support/knowledgecenter` for the IBM Knowledge Center home page, or `INFOCENTER=http://www.ibm.com/support/knowledgecenter/SSGMCP_5.2.0` for the CICS TS 5.2 documentation.

INITPARM

The **INITPARM** system initialization parameter specifies that parameters are to be passed to application programs that use the **ASSIGN INITPARM** command.

INITPARM=(pgmname_1='parmstring_1'[, ... ,pgmname_n='parmstring_n'])

You can use **INITPARM** to pass parameters to PLTPI programs to be executed in the final stages of system initialization. The area giving access to the parameters is specified by the **ASSIGN INITPARM** command. For programming information about the **ASSIGN INITPARM** command, see ASSIGN .

pgmname

The name of a program. This name must be 1 through 8 alphanumeric or national language characters.

parmstring

The parameter string (up to 60 characters enclosed by single quotes) to be passed to the associated program. Any quotes imbedded in the string must be duplicated. For information on coding INITPARM in the SYSIN data set, see Rules for coding CICS system initialization parameters in the SYSIN data set.

You can specify up to 255 pgmname='parmstring' sets.

Note: You can specify the INITPARM keyword and its parameters more than once, see A sample CICS startup job. If you specify INITPARM multiple times

for the same program, the final INITPARM parameter specified is the parameter that the system uses. If you specify INITPARM multiple times for different programs, the INITPARM parameters specified are merged.

INTTR

The **INTTR** system initialization parameter specifies whether the internal CICS trace destination is to be activated at system initialization.

INTTR={ON|OFF}

This parameter controls whether any of the three types of CICS trace entry are written to the internal trace table. The three types are: CICS system trace (see the SYSTR parameter), user trace (see the USERTR parameter), and exception trace entries (which are always made and not controlled by a system initialization parameter).

ON Activate main storage trace.

OFF Do not activate main storage trace.

IRCSTRT

The **IRCSTRT** system initialization parameter specifies whether IRC is to be started up at system initialization.

IRCSTRT={NO|YES}

If IRCSTRT=YES is not coded, IRC can be initialized by issuing a **SET IRC OPEN** command.

ISC

The **ISC** system initialization parameter specifies whether the CICS programs required for multiregion operation (MRO) and intersystem communication over SNA are to be included.

ISC={NO|YES}

For IPIC interconnectivity (IPIC), specify ISC=YES and TCPIP=YES.

JESDI

The **JESDI** system initialization parameter specifies, in a SIT for an alternate XRF system, the JES delay interval.

JESDI={30|number} (alternate)

The minimum is 5 seconds. The alternate CICS region has to ensure that the active CICS region has been canceled before it can take over the resources owned by the active.

Note: You must give careful consideration to the values you specify for the parameters ADI and JESDI so that they do not conflict with your installation's policy on PR/SM RESETTIME and the XCF INTERVAL and OPNOTIFY intervals. You should ensure that the sum of the interval you specify for ADI plus JESDI exceeds the interval specified by the XCF INTERVAL and the PR/SM policy interval RESETTIME.

JVMPROFILEDIR

The **JVMPROFILEDIR** system initialization parameter specifies the name (up to 240 characters long) of a z/OS UNIX directory that contains the JVM profiles for CICS. CICS searches this directory for the profiles it needs to configure JVMs.

JVMPROFILEDIR={/usr/lpp/cicsts/cicsts54/JVMProfiles|*directory*}

The default value of **JVMPROFILEDIR** is the same as the default value of the USSHOME system initialization parameter, which specifies the name and path of the root directory for CICS files on z/OS UNIX, followed by the subdirectory `JVMProfiles`. Changing the **USSHOME** parameter has no effect on the **JVMPROFILEDIR** parameter value.

If you want CICS to load the JVM profiles from a different directory, you must complete one of the following tasks:

- Change the value of the **JVMPROFILEDIR** system initialization parameter.
- Use UNIX soft links to link to your JVM profiles from the directory that is specified by **JVMPROFILEDIR** . Use this method to store your JVM profiles in any place in the z/OS UNIX file system.

For JVM servers that are defined in CICS bundles, the JVM profile is packaged in the CICS bundle along with the JVMSERVER resource definition. CICS does not load these JVM profiles from the directory that is specified by **JVMPROFILEDIR**. Instead, the file path is relative to the root directory of the CICS bundle. For more information see Referencing zFS artifacts in a bundle.

KEYRING

The **KEYRING** system initialization parameter specifies the fully qualified name of the key ring, within the external security manager's database, that contains the keys and X.509 certificates used by CICS support for the secure sockets layer (SSL) and for web services security.

The **KEYRING** parameter is not used by Liberty JVM server's SSL support. If this is your only use of SSL in the CICS region, then you do not have to specify the **KEYRING** parameter.

KEYRING=*keyring-name*

The maximum length of the **KEYRING** parameter is 47 characters, and the key ring name is case-sensitive. For more information about creating a key ring, see Building a key ring manually.

LGDFINT

The **LGDFINT** system initialization parameter specifies the log defer interval to be used by CICS log manager when determining how long to delay a forced journal write request before invoking the MVS system logger.

LGDFINT={5|*number*}

The value is specified in milliseconds.

5 This is the default. When this parameter was first introduced, the default value was 30 milliseconds, but customer experience has shown that 5 is a more realistic value.

number
number can be any value in the range 0 through 65535. You are recommended to allow **LGDFINT** to assume its default value, 5.

You can modify the log defer interval dynamically using the LOGDEFER option of the **CEMT SET SYSTEM** or **EXEC CICS SET SYSTEM** command. However, you are recommended not to modify this value in a production environment without first performing a system evaluation and performance analysis of any changed value.

If you change the log defer interval value dynamically, the new value is not cataloged. The log defer interval value is taken from the **LGDFINT** system initialization parameter in all types of CICS startup.

When a CICS system has many tasks issuing forced log write requests, these tasks will not be delayed for periods close to the **LGDFINT** parameter value. This is because a forced log write request is normally issued while a log deferral is already being performed for another task. The actual interval might also be affected by the need for tasks to wait across a partition exit.

LGNMSG

The **LGNMSG** system initialization parameter specifies whether z/OS Communications Server logon data is to be made available to an application program.

LGNMSG={NO|YES}

Valid values are as follows:

NO z/OS Communications Server logon data is not available to an application program.

YES z/OS Communications Server logon data is available to an application program. The data can be retrieved with an EXEC CICS EXTRACT LOGONMSG command. For programming information about this command, see EXTRACT LOGONMSG.

You can use this parameter with the **GMTRAN** parameter to retrieve the z/OS Communications Server logon data at the time a terminal is logged on to CICS by z/OS Communications Server.

LLACOPY

The LLACOPY system initialization parameter specifies the situations where CICS uses either the LLACOPY macro or the BLDL macro when locating modules in the DFHRPL or dynamic LIBRARY concatenation.

LLACOPY={YES|NO|NEWCOPY}

Valid values are as follows:

YES CICS always uses the LLACOPY macro when locating modules in the DFHRPL or dynamic LIBRARY concatenation.

NO CICS always uses the BLDL macro when locating modules in the DFHRPL or dynamic LIBRARY concatenation.

NEWCOPY
CICS uses the LLACOPY only when a NEWCOPY or a PHASEIN is being performed. At all other times, CICS uses the BLDL macro when locating modules in the DFHRPL or dynamic LIBRARY concatenation.

You can improve the performance of module fetching on your system by allowing library lookaside (LLA) to manage your production load libraries.

LLA reduces the amount of I/O needed to locate and fetch modules from DASD storage. For more information about this, refer to Improving module fetch performance with LLA.

Note:

1. If you code LLACOPY=NO or LLACOPY=NEWCOPY you can still benefit from having LLA managed data sets within your DFHRPL or dynamic LIBRARY concatenation. Modules will continue to be loaded from the virtual lookaside facility (VLF) if appropriate. For more information about VLF and LLA refer to Controlling LLA and VLF through operator commands.
2. If an LLA managed module has been altered, a BLDL macro may not return the new information and a subsequent load will still return the old copy of the module. To load the new module, an LLACOPY must be issued against that module or a MODIFY LLA,REFRESH command must be issued on a system console.
3. If you set LLACOPY to anything other than NO, ensure that the proper RACF security permissions have been set up first. For more information about this refer to Resources protected by the FACILITY general resource class.
4. If an LLACOPY is issued against an LLA managed module, it creates a BLDL macro to interact with the specified DCB. If the directory information does not match the information stored in LLA, the LLA tables are updated to keep both subsystems synchronized. Whilst the LLA tables are being updated, SYSZLLA1.update has an enqueue (lock) held against it until LLA is stopped or the library is removed from LLA management.

LOCALCCSID

The **LOCALCCSID** system initialization parameter specifies the default CCSID for the local region.

LOCALCCSID={037|CCSID}

The CCSID is a value of up to 8 characters. If CCSID value is not specified, the default LOCALCCSID is set to 037. For lists of valid CCSIDs, see:

- CICS-supported conversions
- The relevant appendix in z/OS Unicode Services User's Guide and Reference.

037 The default value for LOCALCCSID.

CCSID
Represents any other valid EBCDIC CCSID value.

LPA

The **LPA** system initialization parameter specifies whether CICS and user modules can be used from the link pack areas.

LPA={NO|YES}

Valid values are as follows:

NO will not load CICS or user modules from the link pack areas.

YES CICS or usermodules installed in the LPA or in the ELPA can be used from there, instead of being loaded into the CICS region.

A list of the CICS modules that are read-only, and hence eligible for residence in the link pack areas (LPA or ELPA), are contained in the SMP/E USERMOD supplied on the distribution tape in the CICSTS54.CICS.SDFHSAMP, in a member called DFH$UMOD. For details of the CICS system initialization parameter PRVMOD that you can use to override LPA=YES for selected modules, see "PRVMOD" on page 83.

MAXOPENTCBS

The **MAXOPENTCBS** system initialization parameter specifies the maximum number, in the range 32 through 4032, of open task control blocks (open TCBs) CICS can create in the pool of L8 and L9 mode TCBs.

If you do not specify the MAXOPENTCBS parameter, the MXT value will be used to set a value for the MAXOPENTCBS parameter. For more information on how the value is set from MXT, see Open TCB management.

If you explicitly specify the MAXOPENTCBS parameter, or change its value dynamically using EXEC CICS SET DISPATCHER, then you must set it to a correct and optimum value. If you change MXT, however you choose to specify it (as a system initialization parameter, using CEMT, or EXEC CICS SET DISPATCHER, or using CICS Explorer®) you should review your explicit setting for MAXOPENTCBS. To revert to allowing CICS to set this parameter automatically after specifying it explicitly, you would have to reinitialize both the local and global catalogs. Before you reinitialize both the global and local catalogs, see The local catalog.

MAXOPENTCBS=*number*

Within this limit, there are no constraints on how many of the TCBs in the pool are L8 TCBs, and how many are L9 TCBs.

- L8 mode TCBs are used in the following circumstances:
  - For CICSKEY OPENAPI application programs.
  - For OPENAPI task-related user exits (TRUEs), for example the CICS-DB2 and WebSphere® MQ Attachment Facilities, and the CICS-DBCTL Database Adapter Transformer (DFHDBAT) when used with IMS™ Version 12 or later. TRUEs always run in CICSKEY.
  - By CICS itself, because CICS uses OPENAPI CICSKEY programs that run on L8 TCBs when accessing doc templates and HTTP static responses that are stored in z/OS UNIX System Services files, or when processing web service requests and parsing XML. CICS also runs some security requests on L8 TCBs. CICS uses L8 TCBs when authenticating a password, password phrase or kerberos token when the original request was on the QR TCB.
- L9 mode TCBs are used for USERKEY OPENAPI application programs.

For more information about open TCBs, see Open TCB management.

MAXSOCKETS

The **MAXSOCKETS** system initialization parameter specifies the maximum number of IP sockets that can be managed by the CICS sockets domain.

MAXSOCKETS={65535|*number***}**

Set a suitable value that does not exceed the maximum value as defined in the **MAXFILEPROC** parameter in SYS1.PARMLIB member BPXPRMxx. If you specify a

value greater than the **MAXFILEPROC** parameter, CICS issues message DFHSO0124, which specifies the value that CICS has used for this parameter. If the CICS region user ID has superuser authority, the **MAXFILEPROC** parameter does not limit the setting of **MAXSOCKETS**.

The maximum number of sockets must be greater than the maximum number of inbound and outbound sockets used by CICS including the number of TCPIPSERVICE resources in service.

MAXSSLTCBS

The **MAXSSLTCBS** system initialization parameter specifies the maximum number of S8 TCBs that can run in the SSL pool.

MAXSSLTCBS={32|*number***}**

The default is 32, but you can specify between 1 and 1024 TCBs.

This value must not exceed the **MAXTHREADS** and **MAXTHREADTASKS** parameter values, that are specified in SYS1.PARMLIB member BPXPRMxx.

For more information about open TCBs, see Open TCB management.

MAXXPTCBS

The **MAXXPTCBS** system initialization parameter specifies the maximum number, in the range 1 through 2000, of open X8 and X9 TCBs that can exist concurrently in the CICS region.

If you do not specify the MAXXPTCBS parameter, the MXT value will be used to set a value for the MAXXPTCBS parameter. For more information on how the value is set from MXT, see Open TCB management.

If you explicitly specify the MAXXPTCBS parameter, or change its value dynamically using EXEC CICS SET DISPATCHER, then you must set it to a correct and optimum value. If you change MXT, however you choose to specify it (as a system initialization parameter, using CEMT, or EXEC CICS SET DISPATCHER, or using CICS Explorer) you should review your explicit setting for MAXXPTCBS. To revert to allowing CICS to set this parameter automatically after specifying it explicitly, you would have to reinitialize both the local and global catalogs. Before you reinitialize both the global and local catalogs, see The local catalog.

MAXXPTCBS=*number*

X8 and X9 are the TCBs that are used to provide support for C and C++ programs compiled with the XPLINK option.

For more information about open TCBs, see Open TCB management.

MCT

The **MCT** system initialization parameter specifies the monitoring control table suffix.

MCT={NO|YES|xx}

If you specify MCT=NO, CICS monitoring builds dynamically a default MCT, ensuring that default monitoring control table entries are always available for use when monitoring is on and a monitoring class (or classes) is active. You can generate an MCT with a single-character suffix only for use by CICS

because single-character suffixes cause an error when the MCT is processed by DFHMNDUP. If you use DFHMNDUP, make sure that you create your MCTs with two-character suffixes.

For information about coding the macros for this table, see Generating a performance dictionary record using DFHMNDUP.

MINTLSLEVEL

The **MINTLSLEVEL** system initialization parameter specifies the minimum TLS protocol that CICS uses for secure TCP/IP connections.

MINTLSLEVEL={TLS10|TLS11|TLS12|MINTLS10ONLY}

When a secure connection is established between a pair of processes, the most secure TLS protocol that is supported by both is used.

TLS10 Sets the minimum level of TLS to 1.0. This is the default value.

Note: If you used ENCRYPTION=STRONG on previous releases, a change in behavior might occur if the client does not handle the negotiation of TLS levels correctly. If this causes a problem, use MINTLSLEVEL=TLS10ONLY instead.

TLS11 Sets the minimum level of TLS to 1.1.

TLS12 Sets the minimum level of TLS to 1.2.

TLS10ONLY
Sets the level of TLS to 1.0 only.

To apply FIPS 140-2 standards, set **MINTLSLEVEL**=TLS12 and **NISTSP800131A**=CHECK. If **NISTSP800131A**=CHECK is set but **MINTLSLEVEL** is set to a value other than TLS12, it is overridden to **MINTLSLEVEL**=TLS12 and a warning message is issued.

To apply FIPS 140-2 standards on z/OS Version 2 Release 1 or later, ICSF (Integrated Cryptographic Services Facility) must be active on your system. If you have not already done so, apply APAR OA14956 to z/OS.

For more information about NIST SP800-131A conformance, see Making your CICS TS system conformant to NIST SP800-131A.

MN

The **MN** system initialization parameter specifies whether monitoring is to be switched on or off at initialization.

MN={OFF|ON}

Use the individual monitoring class system initialization parameters to control which monitoring classes are to be active (see the MNEXC, MNPER, and MNRES parameter descriptions.) The default status is that the CICS monitoring facility is **off**. The monitoring status is recorded in the CICS global catalog for use during warm and emergency restarts.

OFF Switch off monitoring.

ON Turn on monitoring. However, unless at least one individual class is active, no monitoring records are written.

Note:

1. If the monitoring status is ON, CICS accumulates monitoring data continuously. For each monitoring class that is active, CICS writes the

monitoring data to a system management facilities (SMF) data set. If the monitoring status is OFF, CICS does not accumulate or write any monitoring data, even if any of the monitoring classes are active.

2. You can change the monitoring status and the monitoring class settings at any time, as follows:
   - During a warm restart by coding system initialization parameters in PARM, SYSIN, or through the system console.
   - While CICS is running, using:
     - The monitoring facility transaction CEMN.
     - The CEMT SET MONITOR command.
     - The EXEC CICS SET MONITOR command.

   When you change the status of monitoring, the change takes effect immediately. If you change the monitoring status from OFF to ON, monitoring starts to accumulate data and write monitoring records to SMF for all tasks that start after the status change is made, for all active monitoring classes. If the status is changed from ON to OFF, monitoring stops writing records immediately and does not accumulate monitoring data for any tasks that start after the status change is made.
3. The monitoring status setting can be manipulated independently of the class settings. This means that, even if the monitoring status is OFF, you can change the monitoring class settings, and the changes take effect for all tasks that are started after the monitoring status is next set to ON.

MNCONV

The **MNCONV** system initialization parameter specifies whether conversational tasks have separate performance class records produced for each pair of terminal control I/O requests.

MNCONV={NO|YES}

Any clock (including user-defined) that is active at the time such a performance class record is produced is stopped immediately before the record is written. After the record is written, such a clock is reset to zero and restarted. Thus a clock whose activity spans more than one recording interval within the conversational task appears in multiple records, each showing part of the time, and the parts add up to the total time that the clock is active. The high watermark fields (which record maximum levels of storage used) are reset to their current values. All other fields are set to X'00', except for the key fields (transid, termid). The monitoring converse status is recorded in the CICS global catalog for use during warm and emergency restarts.

MNEXC

The **MNEXC** system initialization parameter specifies whether the monitoring exception class is to be made active during initialization.

MNEXC={OFF|ON}

The monitoring exception class status is recorded in the CICS global catalog for use during warm and emergency restarts.

OFF Set the exception monitoring class to “not active”.

ON Set the exception monitoring class to “active”.

For programming information about exception monitoring records, see The schedule flag word.

MNFREQ

The **MNFREQ** system initialization parameter specifies the interval for which CICS automatically produces a transaction performance class record for any long-running transaction.

MNFREQ={0|hhmmss}

The monitoring frequency value is recorded in the CICS global catalog for use during warm and emergency restarts. CICS can produce a performance class monitoring record in this way only when the long-running transaction is running on the QR or CO TCBs.

0 No frequency monitoring is active.

hhmmss

The interval for which monitoring produces automatically a transaction performance class record for any long-running transaction. Specify a 1 to 6 digit number in the range 000100–240000. Numbers that are fewer than six digits are padded with leading zeroes.

MNIDN

The **MNIDN** system initialization parameter specifies whether the monitoring identity class is to be made active during CICS initialization.

MNIDN={OFF|ON}

The monitoring identity class status is recorded in the CICS global catalog for use during warm and emergency restarts.

OFF Set identity monitoring class to not active.

ON Set identity monitoring class to active.

MNPER

The **MNPER** system initialization parameter specifies whether the monitoring performance class is to be made active during CICS initialization.

MNPER={OFF|ON}

The monitoring performance class status is recorded in the CICS global catalog for use during warm and emergency restarts.

OFF Set the performance monitoring class to "not active".

ON Set the performance monitoring class to active.

For programming information about performance monitoring records, see CICS monitoring facility: Performance and tuning.

MNRES

The **MNRES** system initialization parameter specifies whether transaction resource monitoring is to be made active during CICS initialization.

MNRES={OFF|ON}

The transaction resource monitoring class status is recorded in the CICS global catalog for use during warm and emergency restarts.

OFF Set transaction resource monitoring to not active.

ON Set transaction resource monitoring to active.

Transaction resource monitoring applies to CICS file resources when you specify the FILE=*nn* option on the DFHMCT TYPE=INTIAL macro.

MNSYNC

The **MNSYNC** system initialization parameter specifies whether you want CICS to produce a transaction performance class record when a transaction takes an implicit or explicit syncpoint (unit-of-work).

MNSYNC={NO|YES}

No action is taken for syncpoint rollbacks. The monitoring syncpoint status is recorded in the CICS global catalog for use during warm and emergency restarts.

MNTIME

The **MNTIME** system initialization parameter specifies whether you want the time stamp fields in the performance class monitoring data to be returned to an application using the **EXEC CICS COLLECT STATISTICS MONITOR(taskno)** command in either GMT or local time.

MNTIME={GMT|LOCAL}

The monitoring time value is recorded in the CICS global catalog for use during warm and emergency restarts.

For programming information on the **EXEC CICS COLLECT STATISTICS** command, see COLLECT STATISTICS.

MQCONN

The **MQCONN** system initialization parameter specifies whether you want CICS to start a connection to WebSphere MQ automatically during initialization.

MQCONN={NO|YES}

NO Do not automatically call DFHMQCOD, the CICS-WebSphere MQ adapter program, during initialization.

YES

Call the CICS-WebSphere MQ adapter program, DFHMQCOD, automatically during CICS initialization. The **MQCONN** parameter always uses program DFHMQCOD to start the CICS-WebSphere MQ connection. It cannot be customized to use a user-supplied attach program of a different name.

When you specify **MQCONN=YES**, the information that CICS needs to start the connection to WebSphere MQ, such as the name of a WebSphere MQ queue manager or queue-sharing group, is taken from the MQCONN resource definition for the CICS region.

An MQCONN resource definition must be installed before CICS can start the connection to WebSphere MQ. When you start the connection automatically at CICS initialization, for an initial or cold start, the MQCONN resource definition must be present in one of the groups named in the list or lists named by the **GRPLIST** system initialization parameter. For a warm or emergency start of CICS, the MQCONN resource definition must have been installed by the end of the previous CICS run.

When you specify **MQCONN=YES**, you do not need to define the CICS-WebSphere MQ adapter program in the CICS post initialization program list table (PLT).

MROBTCH

The **MROBTCH** system initialization parameter specifies the number of events that must occur before CICS is posted for dispatch because of the batching mechanism.

MROBTCH={1|number}

The number can be in the range 1 through 255, and the default is 1.

Use this batching mechanism to spread the overhead of dispatching CICS over several tasks. If the value is greater than 1 and CICS is in a system wait, CICS is not posted for dispatch until the specified number of events has occurred. Events include MRO requests from connected systems or DASD I/O and CHANGE_MODE processing. For these events, CICS is dispatched as soon as one of the following occurs:

- The current batch fills up (the number of events equals MROBTCH)
- An ICV interval expires

Therefore, ensure that the time interval you specify in the ICV parameter is low enough to prevent undue delay to the system.

If CICS is dispatched for another reason, the current batch is dealt with in that dispatch of CICS.

Note: During periods of low utilization, a value of MROBTCH greater than 1 might cause increased transaction response times. Transactions that issue file I/O requests might be delayed because of increased FCIOWAIT value. For more information about the effect of MROBTCH on performance, see Batching requests (MROBTCH).

MROFSE

The **MROFSE** system initialization parameter specifies whether you want to extend the lifetime of the long-running mirror to keep it allocated until the end of the task rather than after a user syncpoint for function shipping applications.

MROFSE={NO|YES}

Valid values are as follows:

NO The lifetime of the MRO long-running mirror is not extended.

YES The mirror task remains available to the application until the end of the application's task. This extended long-running mirror saves the overhead of re-attaching the mirror task following a user syncpoint.

This parameter is ignored for DPL requests (that is a DPL causes the session to be freed at the next syncpoint even if is has been kept for a previous sequence of syncpoints).

It should be used with caution especially if DPL requests with SYNCONRETURN or TRANSID are used. For additional information, see Long-running mirror tasks for MRO

Do not specify this value in the front-end region when long running tasks might be used to function-ship requests. This because a SEND session is unavailable for allocation to other tasks when unused. Specifying MROFSE=YES could prevent the connection from being released when contact has been lost with the back-end region, until the task terminates or issues a function-shipped request.

MROLRM

The **MROLRM** system initialization parameter specifies whether you want to establish an MRO long-running mirror task.

MROLRM={NO|YES}

Valid values are as follows:

NO The MRO long-running mirror task is not required.

YES The mirror transaction remains available to the application issuing the remote request. This long-running mirror saves the overhead of re-establishing communication with the mirror transaction if the application makes more function shipping requests in this unit of work.

For information about long-running mirror tasks, see Long-running mirror tasks for MRO.

MSGCASE

The **MSGCASE** system initialization parameter specifies how you want the message domains to display mixed case messages.

MSGCASE={MIXED|UPPER}

Messages handled by the CICS message domain and the CPSM message domain are in mixed case.

MIXED
This is the default in the SIT; all messages displayed by the CICS message domain or the CPSM message domain remain in mixed case.

UPPER
The message domain displays all mixed case messages in uppercase only.

Mixed case output is not displayed correctly on Katakana display terminals and printers. Uppercase English characters appear correctly as uppercase English characters, but lowercase appears as Katakana symbols. If you have any Katakana terminals connected to your CICS region, specify MSGCASE=UPPER.

If you want to use uppercase English for your CICS-WebSphere MQ components, you must set MSGCASE=UPPER, and ensure that ASSIGN NATLANGINUSE returns E (US English).

MSGLVL

The **MSGLVL** system initialization parameter specifies the message level that controls the generation of messages to the console and JES message log.

MSGLVL={1|0}

Valid values are as follows:

1 All messages are printed or displayed.
0 Only critical errors or interactive messages are printed or displayed.

MXT

The **MXT** system initialization parameter specifies the maximum number, in the range 10 through 2000, of user tasks that can exist in a CICS system at the same time. The MXT value does not include CICS system tasks.

MXT={250|number}

CICS queues requests for tasks above this number but does not action (attach) them until the number of tasks attached drops below the MXT limit. See Setting the maximum task specification (MXT).

Review the size specified on the z/OS **REGION** and **MEMLIMIT** parameters for the CICS address space. The increase in CICS use of virtual storage above 16 MB but below 2 GB (above the line) means that you probably need to increase the **REGION** parameter. The increase in CICS use of virtual storage above 2 GB (above the bar) means that you probably need to increase the **MEMLIMIT** parameter. See Setting the limits for CICS storage in Improving performance.

For CICS regions that operate with transaction isolation, the transaction isolation facility increases the allocation of some virtual storage above 16 MB but below 2 GB.

- If the CICS region operates with transaction isolation, CICS allocates storage for task-lifetime storage in multiples of 1 MB for user-key tasks that run above 16 MB but below 2 GB. 1 MB is the minimum unit of storage allocation for the extended user dynamic storage area (EUDSA) when transaction isolation is active. However, although storage above 16 MB but below 2 GB is allocated in multiples of 1 MB, MVS paging activity affects only the storage that is used (referenced), and unused parts of the 1 MB allocation are not paged.
- If the CICS region operates without transaction isolation, CICS allocates user-key task-lifetime storage above 16 MB but below 2 GB in multiples of 64 KB.

The subspace group facility uses more real storage, because MVS creates a page and segment table from real storage for each subspace. The CICS requirement for real storage varies depending on the transaction load at any one time. As a guideline, each task in the system requires 9 KB of real storage, and this should be multiplied by the number of concurrent tasks that can be in the system at any one time (governed by the MXT system initialization parameter).

Note: If the MAXOPENTCBS or MAXXPTCBS system initialization parameters have not been specified, then setting MXT will affect the MAXOPENTCBS and MAXXPTCBS settings.

For more information about open TCBs, see Open TCB management.

Important: Before you change the **MXT** system initialization parameter, review the information in Open TCB pools.

Note: From CICS Transaction Server for z/OS, Version 5 Release 4, tasks that are internally initiated in a MAS by CICSPlex SM no longer execute as user tasks. As a result these tasks no longer qualify under the MXT and transaction class limits .

NATLANG

The **NATLANG** system initialization parameter specifies the single-character code for the language to be supported in this CICS run.

NATLANG=(E|C|K)

Valid code values are as follows:

E English, which is the *system* default (that is, is provided even if you do not specifically code E).

C Simplified Chinese, a Double-Byte Character Set language. Translation is performed by IBM.

K Japanese, a Double-Byte Character Set language. Translation is performed by IBM.

English language support is provided, even if you do not specifically code E for English.

Globalization is not available to CICS console messages, which continue to be in English only.

NCPLDFT

The NCPLDFT system initialization parameter specifies the name of the default named counter pool to be used by the CICS region on calls it makes to a named counter server.

NCPLDFT={DFHNC001|name}

If CICS cannot determine, from the named counter options table, the pool name required by an EXEC CICS named counter command, CICS uses the default name specified on the NCPLDFT parameter.

Note: This parameter is relevant to references to a named counter server made through the EXEC CICS API only. It not used by the named counter call interface.

DFHNC001
This is the default name that CICS uses as the named counter pool name if you omit the NCPLDFT system initialization parameter.

name Specifies the 8-character name to be used by CICS as the default pool name in connection with named counter API commands, when the name cannot be resolved by the named counter options table.

NEWSIT

The **NEWSIT** system initialization parameter specifies whether CICS is to load the specified SIT, and enforce the use of all system initialization parameters, modified by any system initialization parameters provided by PARM, SYSIN, or the system console, even in a warm start.

NEWSIT={YES|NO}

Enforcing the use of system initialization parameters in this way overrides any parameters that may have been stored in a warm keypoint at shutdown.

However, there are some exceptions. The following system initialization parameters are always ignored in a warm start, even if they are supplied by PARM, SYSIN, or the console:

- CSDACC
- CSDBUFND
- CSDBUFNI
- CSDDISP
- CSDDSN
- CSDFRLOG
- CSDINTEG

- CSDJID
- CSDLSRNO
- CSDRECOV
- CSDRLS
- CSDSTRNO
- FCT
- GRPLIST

In a warm restart, CICS uses the **installed** resource definitions saved in the CICS global catalog at warm shutdown, and therefore the CSD, FCT, and GRPLIST parameters are ignored. (At CICS startup, you can only modify installed resource definitions, including file control table entries, or change to a new FCT, by performing a cold start of CICS with START=COLD.)

For more information about the use of the NEWSIT parameter, see Controlling start and restart.

Restrictions

You can specify the NEWSIT parameter in PARM, SYSIN, or CONSOLE only.

NISTSP800131A

The **NISTSP800131A** system initialization parameter specifies whether the CICS region is to check for conformance to the NIST SP800-131A standard.

NISTSP800131A={NOCHECK|CHECK}

NOCHECK

Conformance checking is not required in this CICS region. This is the default value.

CHECK

The CICS region is required to check for conformance with the NIST SP800-131A security standard. If this value is set, CICS issues a message if an actual or potential violation is detected.

This option also causes the CICS SSL environment to use only TLS v 1.2 with FIPS 140-2 standards applied.

The checks that are performed are as follows:

Web services

If the <wsse_handler> tag is specified in a pipeline configuration file, it implies that the pipeline is to be used for web services security. Because not all of the encryption algorithms that can be used for web services security conform to SP800-131A, installing a pipeline that uses web services means that CICS might be non-conformant. CICS issues message DFHXS1300, which warns of potential nonconformance.

If you receive message DFHXS1300, check whether you are using DFHWSSE as the web services security handler. If you are not using DFHWSSE, inspect your security handler to check which encryption and signing algorithms it uses. If these algorithms are SP800-131A conformant, you can ignore the message. If they are not conformant, consider whether to use conformant algorithms instead. Otherwise, if the CICS region that issues the message must be conformant, consider moving

the web service security workload to a different CICS region where conformance is not required.

CICS also checks for certain things that are not conformant to SP800-131A. If any of these situations are found, CICS issues message DFHXS1301:

- An <algorithm> element exists within the <authentication> element of the <wsse_handler> definition in the pipeline configuration file. The only algorithms that can be used are SHA-1 routines, which are not conformant with NIST SP800-131A.
- A <sign_body> element exists in the pipeline configuration file. The only algorithms that can be used are SHA-1 routines, which are not conformant with SP800-131A.
- An <encrypt_body> element in the pipeline configuration file. Of the four algorithms that can be used, three are conformant with SP800-131A but one is not. If the nonconformant algorithm is specified, DFHXS1301 is issued.

If you receive message DFHXS1301, consider not performing the encryption or signing operations in this CICS region. If the nonconformant algorithm is specified in the <encrypt_body> element, consider using a conformant algorithm.

Sockets

If SSL is active, setting **NISTSP800131A**=CHECK forces **MINTLSLEVEL**=TLS12 if it is not already set. If **MINTLSLEVEL**=TLS12 is forced, message DFHSO0144 is issued. Sockets domain initializes the SSL environment with the FIPS option on and the System SSL started task runs in FIPS mode. The effect of this is that SSL allows fewer ciphers to be used on a successful handshake.

To use FIPS with z/OS Version 2 Release 1 or later, ICSF (Integrated Cryptographic Services Facility) must be active on your system. If you have not already done so, apply APAR OA14956 to z/OS.

If SSL is inactive because no **KEYRING** parameter is specified, then setting NISTSP800131A has no effect on sockets domain.

JVM servers

When a JVM server is started, CICS sets the Java properties to make Java NIST SP800-131A conformant.

If you set **NISTSP800131A**=CHECK, you should also set **MINTLSLEVEL**=TLS12. However, if you do not do so, CICS overrides the value of **MINTLSLEVEL** to **MINTLSLEVEL**=TLS12 and issues a warning message.

NONRLSRECOV

The **NONRLSRECOV** system initialization parameter specifies whether VSAM catalog recovery options should override those specified on the CICS FILE resource definition for all non-RLS files. Default behavior, with **NONRLSRECOV=VSAMCAT**, will take recovery attributes from the catalog if they are present, and from the file definition otherwise. RLS files must always specify recovery options on the catalog.

To open a data set in non-RLS mode, ensure that **NONRLSRECOV=FILEDEF**, and alternative recovery attributes will be used.

Where VSAM log replication is being used, this parameter must be set to **NONRLSRECOV=VSAMCAT**.

NONRLSRECOV={VSAMCAT|FILEDEF}
Recovery options do not apply to read-only files. Valid values are as follows:

VSAMCAT
By default, CICS uses the recovery options that are specified on the VSAM catalog for non-RLS files. These recovery options include the LOG, LOGSTREAMID, and BWO options. If no recovery options are set, CICS uses the attributes on the FILE resource.

FILEDEF
For non-RLS files, CICS ignores any recovery options on the catalog and uses the values specified in the FILE resource instead. The recovery attributes for the CSD are set by the appropriate system initialization parameters. This option is not compatible with the LOGREPLICATE option, which can only be specified on the VSAM catalog.

NQRNL

The **NQRNL** system initialization parameter controls resource name list (RNL) processing by z/OS global resource serialization, which can cause the scope value of a resource to change. CICS uses z/OS global resource serialization to provide sysplex-wide protection of application resources.

NQRNL={NO|YES}

For more information on global resource serialization and RNL processing, see z/OS MVS Planning: Global Resource Serialization.Valid values are as follows:

NO NO is the default value in CICS. When **NQRNL=NO** is specified, the sysplex-wide **EXEC CICS ENQ** and **EXEC CICS DEQ** commands use the **RNL=NO** parameter on the z/OS ENQ or DEQ requests. Use **NQRNL=NO** when you are sure that you want the request to be processed only by z/OS global resource serialization using only the scope specified in the ENQMODEL resource definition in CICS. When **NQRNL=NO** is specified, the sysplex-wide **EXEC CICS ENQ** and **EXEC CICS DEQ** requests are ignored by alternative serialization products.

YES When **NQRNL=YES** is specified, the sysplex-wide **EXEC CICS ENQ** and **EXEC CICS DEQ** commands use the **RNL=YES** parameter on the z/OS ENQ or DEQ requests. This parameter allows z/OS global resource serialization to perform RNL processing, searching an appropriate RNL to determine the scope of the resource. This setting is the default in z/OS.

For more information on using ENQMODEL definitions in CICS to specify the scope for resources, see ENQMODEL resources.

OFFSITE

The **OFFSITE** system initialization parameter specifies whether CICS is to restart in off-site recovery mode; that is, a restart is taking place at a remote site.

OFFSITE={NO|YES}
For a successful off-site restart, the log records of the failed CICS region must be available at the remote site. CICS does not provide a facility for shipping log records to a remote backup site, but you can use a suitable vendor product

to perform this function. See the relevant product documentation for other procedures you need to follow for a remote site restart.

See Administering restart and recovery for more information about remote site recovery.

NO CICS will not perform the special restart processing required for remote site recovery.

YES CICS will perform an off-site restart at a remote site following a disaster at the primary site. CICS performs this special processing for an off-site restart, because some information (for example, a VSAM lock structure) is not available at the remote site.

CICS performs an emergency restart, even if the global catalog indicates that CICS can do a warm start. OFFSITE=YES is valid with START=AUTO only, and CICS initialization is terminated if you specify START=COLD or INITIAL.

Restrictions

You can specify the OFFSITE parameter in PARM, SYSIN, or CONSOLE only.

OPERTIM

The OPERTIM system initialization parameter specifies the write-to-operator timeout value, in the range 0 through 86400 seconds (24 hours).

OPERTIM={120|number}

This is the maximum time in seconds that CICS waits for a reply before returning control to this transaction. You can change the write-to-operator timeout value when issuing messages to the console from an application by using the timeout option on the **WRITE OPERATOR** command. See WRITE OPERATOR for details.

OPNDLIM

The OPNDLIM system initialization parameter specifies the open destination and close destination request limit.

OPNDLIM={10|number} (Not required for currently supported releases of z/OS Communications Server.)

This limit is used to restrict the number of concurrent OPNDSTs and CLSDSTs to prevent the z/OS Communications Server from running out of space in the CICS region. The limit may be any value in the range 0 through 999. When large values are used for OPNDLIM, the value on the EDSALIM system initialization parameter and the value on the MVS REGION parameter may need to be adjusted to ensure that enough operating system storage is available.

PARMERR

The **PARMERR** system initialization parameter specifies what action you want to follow if CICS detects incorrect system initialization parameter overrides during initialization.

PARMERR={INTERACT|IGNORE|ABEND}

When specified as an override, this parameter affects only subsequent system initialization parameter overrides. Errors in earlier system initialization

parameter overrides are dealt with according to the **PARMERR** system initialization parameter value in the SIT.

INTERACT
Enables the operator to communicate with CICS through the console and correct parameter errors.

Note: INTERACT is overridden with IGNORE in the following cases:

- If errors are found in PARM or SYSIN for system initialization parameter overrides that are not allowed to be entered from the console
- In certain circumstances, in response to invalid data when you have been trying to correct a previous invalid system initialization parameter keyword or value

IGNORE
CICS ignores errors, and tries to complete initialization.

ABEND
CICS abends.

PDI

The PDI system initialization parameter specifies the XRF primary delay interval, in seconds, in a SIT for an active CICS region.

PDI={30|decimal-value}
The minimum delay that you can specify is 5 seconds. This is the time that must elapse between the (apparent) loss of the surveillance signal in the alternate CICS region, and any reaction by the active CICS region. The corresponding parameter for the alternate CICS region is ADI. PDI and ADI need not have the same value.

PDIR

The **PDIR** system initialization parameter specifies a suffix for the PDIR list.

PDIR={NO|YES|xx}
A PDIR is a list of program specification blocks (PSBs) that define, for DL/I, the use of databases by application programs. A PDIR is applicable only if you are using DL/I remote support. See also The global catalog. Specifying a value other than NO implies to CICS that remote DLI support is required.

For information about coding the macros for this table, see the Macro resource definition in Reference -> System definition.

PGAICTLG

The PGAICTLG system initialization parameter specifies whether autoinstalled program definitions should be cataloged.

PGAICTLG={MODIFY|NONE|ALL}

While CICS is running, you can set dynamically whether autoinstalled programs should be cataloged, by using either the EXEC CICS SET SYSTEM or CEMT SET SYSTEM command.

The setting that you specify for cataloging of autoinstalled programs has no effect on programs that are autoinstalled by a task for an application that is deployed on a platform. These programs are never cataloged.

MODIFY
Autoinstalled program definitions are cataloged only if the program definition is modified by a SET PROGRAM command subsequent to the autoinstall.

NONE
Autoinstalled program definitions are not cataloged. This gives a faster CICS restart (warm and emergency) compared with the MODIFY or ALL options, because CICS does not reinstall definitions from the global catalog. Definitions are autoinstalled on first reference.

ALL Autoinstalled program definitions are written to the global catalog at the time of the autoinstall, and following any subsequent modification.

PGAIEXIT

The **PGAIEXIT** system initialization parameter specifies the name of the program autoinstall exit program.

PGAIEXIT={DFHPGADX|name}
While CICS is running, you can set the name of the program autoinstall exit program dynamically, by using either the EXEC CICS SET SYSTEM or CEMT SET SYSTEM command.

PGAIPGM

The **PGAIPGM** system initialization parameter specifies the state of the program autoinstall function at initialization.

PGAIPGM={INACTIVE|ACTIVE}
While CICS is running, you can set the status of program autoinstall dynamically, by using either the IBM CICS Explorer or CICSPlex SM Web User Interface.

INACTIVE
The program autoinstall function is disabled.

ACTIVE
The program autoinstall function is enabled.

PGCHAIN

The PGCHAIN system initialization parameter specifies the character string that is identified by terminal control as a BMS terminal page-chaining command.

PGCHAIN=character(s)
The character string can be 1 through 7 characters. For more information about the character string, see “PGRET” on page 79.

PGCOPY

The **PGCOPY** system initialization parameter specifies the character string that is identified by terminal control as a BMS command to copy output from one terminal to another.

PGCOPY=character(s)
The character string can be 1 through 7 characters. For more information about the character string, see “PGRET” on page 79.

PGPURGE

The **PGPURGE** system initialization parameter specifies the character string that is identified by terminal control as a BMS terminal page-purge command.

PGPURGE=character(s)
It can be 1 through 7 characters. For more information about the character string, see "PGRET."

PGRET

The **PGRET** system initialization parameter specifies the character string that is recognized by terminal control as a BMS terminal page-retrieval command.

PGRET=character(s)
The character string can be 1 through 7 characters.

1. Each character string is unique with respect to the leading characters of every other transaction identification defined in the CSD. A command requested by a single character precludes the use of all other transaction identifications starting with this character.
2. In pseudoconversational mode, each character string is unique with respect to the leading characters of any terminal input message.
3. A field-separator or other suitable delimiter may be specified in each character string to separate this command code from the remainder of the paging command when entered by an operator. For example:

   ```
   PGCHAIN = X/
   PGCOPY = C/
   PGPURGE = T/
   PGRET = P/
   ```

   This reduces the risk of creating a nonunique command. (See Note 1.)

   Restrictions

   If you specify PGCHAIN, PGCOPY, PGPURGE, or PGRET in the SIT, the characters you choose must not include any of the following: ( ) '

   If you specify PGCHAIN, PGCOPY, PGPURGE, or PGRET as a PARM, SYSIN, or console parameter, do not enclose the characters in quotation marks. The characters you choose must not include an embedded blank or any of the following: ( ) ' =
4. PGCHAIN, PGCOPY, PGPURGE, and PGRET are required only if full function BMS is being used. For information about the BMS page retrieval transaction CSPG, see CSPG - page retrieval .
5. CICS always processes a paging command entered by the operator before initiating a transaction invoked by an EXEC CICS RETURN command with the TRANSID option.

PLTPI

The **PLTPI** system initialization parameter specifies a program list table, which contains a list of programs to be run in the final stages of system initialization.

PLTPI={NO|xx|YES}
For information about coding the macros for the program list table, see Program list table (PLT).

For information about writing initialization programs, see Writing initialization programs. You can use the **INITPARM** system initialization parameter to pass parameters to those programs.

PLTPISEC

The **PLTPISEC** system initialization parameter specifies whether you want CICS to perform command security or resource security checking for PLT programs during CICS initialization.

PLTPISEC={NONE|CMDSEC|RESSEC|ALL}

The PLT programs run under the authority of the userid specified on PLTPIUSR, which must be authorized to the appropriate resources defined by PLTPISEC.

NONE
You do not want any security checking on PLT initialization programs.

CMDSEC
You want CICS to perform command security checking only.

RESSEC
You want CICS to perform resource security checking only.

ALL You want CICS to perform both command and resource security checking.

Restriction: You can specify the PLTPISEC parameter in the SIT, PARM, or SYSIN only.

PLTPIUSR

The **PLTPIUSR** system initialization parameter specifies the user ID that CICS uses for security checking for PLT programs that run during CICS initialization.

PLTPIUSR=*userid*

All PLT programs run under the authority of the specified user ID, which must be authorized to all the resources referenced by the programs, as defined by the PLTPISEC parameter.

PLT programs are run under the CICS internal transaction, CPLT. Before the CPLT transaction is attached, CICS performs a surrogate user check against the CICS region userid (the userid under which the CICS region is executing). This is to ensure that the CICS region is authorized as a surrogate for the userid specified on the PLTPIUSR parameter. This ensures that you cannot arbitrarily specify any PLT userid in any CICS region; each PLT userid must first be authorized to the appropriate CICS region.

If you do not specify the PLTPIUSR parameter, CICS runs PLTPI programs under the authority of the CICS region userid, in which case CICS does not perform a surrogate user check. However, the CICS region userid must be authorized to all the resources referenced by the PLT programs.

Restriction: You can specify the PLTPIUSR parameter in the SIT, PARM, or SYSIN only.

PLTSD

The **PLTSD** system initialization parameter specifies a program list table that contains a list of programs to be run during system termination.

PLTSD={NO|xx|YES}

The default value is NO. See Defining CICS resource table and module keywords.

PRGDLAY

The **PRGDLAY** system initialization parameter specifies the BMS purge delay time interval that is added to the specified delivery time to determine when a message is to be considered undeliverable and therefore purged.

PRGDLAY={0|hhmm}

This time interval is specified in the form *hhmm* (where *hh* represents hours from 00 to 99 and *mm* represents minutes from 00 to 59). If **PRGDLAY** is not coded, or is given a zero value, a message remains eligible for delivery either until it is purged or until there is a cold start of temporary storage.

Note: If you specify **PRGDLAY** as a SIT override, you must still specify a 4-character value (for example 0000).

The **PRGDLAY** facility requires the use of full function BMS. Note also that you must code a **PRGDLAY** value if you want the ERRTERM | ERRTERM(name) parameter on **EXEC CICS ROUTE** commands to be operative.

The **PRGDLAY** value determines the interval between terminal page clean-up operations. A very low value causes the CSPQ transaction to be initiated continuously, and can have a detrimental effect on task-related resources. A zero value stops CSPQ initiating terminal page clean-up. However, this can cause messages to stay in the system forever, resulting in performance problems with long AID queues or lack of temporary storage. The actual purge delay time interval specified is dependent on individual system requirements.

PRINT

The **PRINT** system initialization parameter specifies the method of requesting printout of the contents of a 3270 screen.

PRINT={NO|YES|PA1|PA2|PA3}

Valid values are as follows:

NO Screen copying is not required.

YES Screen copying can be requested by terminal control print requests only.

PA1, PA2, or PA3

Screen copying can be requested by terminal control print request, or by using the PA (program attention) key specified.

The PA key specified by this parameter must not be specified by the TASKREQ option of the RDO TRANSACTION definition or be used for 3270 single keystroke retrieval.

When YES, PA1, PA2, or PA3 is specified, transaction CSPP is initiated which invokes program DFHP3270. The transaction and programs are defined in the CSD group DFHHARDC. In the case of 3270 and LUTYPE2 logical units, the resources defined in CSD group DFHVTAMP are required.

The 3270 print-request facility allows either the application program or the terminal operator to request a printout of data currently displayed on the 3270 display.

If CSPP is invoked to print the screen contents at an associated z/OS Communications Server printer, the screen size of the printer is chosen according to the screen size defined in the profile for the transaction CSPP. The CICS-supplied definitions use the default screen size. Therefore, if you want

DFHP3270 to use the alternate screen size of the printer, you must alter the screen size defined in the profile for the transaction CSPP. For information about defining profiles for transactions, see Define FILE attributes.

For a z/OS Communications Server 3270 display without the printer-adapter feature, the PRINT request prints the contents of the display on the first available 3270 printer specified by PRINTER and ALTPRINTER options of the RDO TERMINAL definition. For a printer to be considered available, it must be in service and not currently attached to a task. It is not necessary for the printer to be on the same control unit.

In an MRO environment, the printer must be owned by the same system as the z/OS Communications Server 3270 display.

For the 3275 with the printer-adapter feature, the PRINT request prints the data currently in the 3275 display buffer on the 3284 Model 3 printer attached to the 3275.

The format of the print operation depends on the size of the display buffer. For a 40-character wide display, the print format is a 40-byte line, and for an 80-character wide display the format is an 80-byte line.

For the 3270 compatibility mode logical unit of the 3790 (if the logical unit has the printer-adapter feature specified), the PRINT request prints the contents of the display on the first printer available to the 3790. The allocation of the printer to be used is under the control of the 3790.

For 3274, 3276, and LUTYPE2 logical units with the printer-adapter feature, the PRINT request prints the contents of the display on the first printer available to the 3270 control unit. The printer to be allocated depends on the printer authorization matrix.

For the 3270 compatibility mode logical unit without the printer-adapter feature, see the preceding paragraph on z/OS Communications Server 3270 displays without the printer-adapter feature.

PRTYAGE

The **PRTYAGE** system initialization parameter specifies the number of milliseconds to be used in the priority aging algorithm that is used to increment the priority of a task.

PRTYAGE={1000|value}

The value can be in the range 0 through 65535, and 1000 is the default.

The priority aging factor is used to increase the effective priority of a task according to the amount of time it is held on a ready queue. The value represents the number of milliseconds that must elapse before the priority of a waiting task can be adjusted upwards by 1. For example, if you code PRTYAGE=3000, a task has its priority raised by 1 for every 3000 milliseconds it is held on the ready queue. Thus a high value for PRTYAGE results in a task being promoted very slowly up the priority increment range, and a low value enables a task to have its priority incremented quickly.

If you specify a value of 0, the priority aging algorithm is not used (task priorities are not modified by age) and tasks on the ready queue are handled according to the user assigned priority.

PRVMOD

The **PRVMOD** system initialization parameter specifies the names of those modules that are not to be used from the LPA.

PRVMOD={name|(name,name...name)}

The operand is a list of 1- to 8-character module names. This enables you to use a private version of a CICS nucleus module in the CICS address space, and not a version that might be in the LPA. For information about **PRVMOD**, see Using modules from DFHRPL.

Restriction: You can specify the **PRVMOD** parameter in PARM, SYSIN, or CONSOLE only.

PSBCHK

The **PSBCHK** system initialization parameter specifies whether CICS is to perform PSB authorization checks for remote terminal users who use transaction routing to initiate a transaction in this CICS region to access an attached IMS system.

PSBCHK={NO|YES}

Valid values are as follows:

NO The remote link is checked, but no check is made against the remote terminal. This value is the default.

YES The remote link is checked, and the remote terminal is also checked if RESSEC(YES) is coded in the definition of the transaction in the CSD.

Note: If you require DL/I security checking, you must specify the XPSB system initialization parameter as XPSB=YES or XSPB=name. For further information about the XPSB system initialization parameter, see XPSB.

Restriction: You can specify the **PSBCHK** parameter in the SIT, PARM, or SYSIN only.

PSDINT

The **PSDINT** system initialization parameter specifies the persistent session delay interval, which states if, and for how long, z/OS Communications Server holds sessions in a recovery-pending state.

PSDINT={0|hhmmss}

0 If a failure occurs, z/OS Communications Server sessions are terminated. Zero is the default, and means that persistent sessions support is not exploited.

hhmmss

The time for which z/OS Communications Server retains sessions if a failure occurs, from 1 second up to the maximum of 23 hours 59 minutes and 59 seconds. Specify a 1 to 6-digit time in hours, minutes, and seconds. If you specify fewer than six digits, CICS pads the value with leading zeros. Thus, a value of 500 is taken as 5 minutes exactly.

You can override this value while CICS is running. Overriding the value changes the action taken by z/OS Communications Server if a failure occurs. The changed interval is not stored in the CICS global catalog, and therefore is not restored in an emergency restart.

z/OS Communications Server holds all sessions in a recovery-pending state for up to the interval specified, unless they are unbound through path failure or z/OS Communications Server operator action, or other-system action in the case of intelligent LUs. The interval you specify must be able to cover the time from a CICS failure to the time when the z/OS Communications Server ACB is opened by CICS during a subsequent emergency restart.

- If you specify SNPS (the default) or MNPS for the **PSTYPE** system initialization parameter for the CICS region, set a nonzero value for the persistent session delay interval, so that sessions are retained.
- If you specify NOPS (no persistent sessions support) for the **PSTYPE** system initialization parameter, a zero value is required for the persistent session delay interval.

When choosing your **PSDINT** value, take account of the types and numbers of sessions involved. You must exercise care when specifying large **PSDINT** values because of the problems such a value might give in some environments, in particular:

- Dial up sessions, for which real costs might be incurred.
- LU6.2 sessions to other host systems. If these sessions are retained in recovery pending state, the other host systems might experience excessive queuing delays. This point applies to LU6.1 sessions that are retained until restart, when they are unbound.

PSTYPE

The **PSTYPE** system initialization parameter specifies whether CICS uses z/OS Communications Server single-node persistent sessions (SNPS), multinode persistent sessions (MNPS), or does not use z/OS Communications Server persistent sessions support (NOPS).

PSTYPE={SNPS|MNPS|NOPS}

The default setting, SNPS (single-node persistent sessions), means that persistent sessions support is available, so that z/OS Communications Server sessions can be recovered after a CICS failure and restart. MNPS (multinode persistent sessions) means that, in addition to the SNPS support, z/OS Communications Server sessions can also be recovered after a z/OS Communications Server or z/OS failure in a sysplex (across LPARs).

For single-node persistent sessions support, you require z/OS Communications Server V3.4.1 or later, which supports persistent LU-LU sessions. CICS Transaction Server for z/OS, Version 5 Release 4 functions with releases of z/OS Communications Server earlier than V3.4.1, but in the earlier releases sessions are not retained in a bound state if CICS fails. For multinode persistent sessions support, you require z/OS Communications Server V4.R4 or later, and z/OS Communications Server must be in a Parallel Sysplex® with a coupling facility.

If you specify SNPS or MNPS, set a nonzero value for the **PSDINT** system initialization parameter, which specifies the retention time for session information. The default is zero, which means that sessions are not retained.

If you do not require persistent sessions support, specify NOPS. A CICS region that is used only for development or testing might not require this support. Removing persistent sessions support where it is not required reduces resource consumption, and can enable you to increase the number of CICS regions in an LPAR. If you specify NOPS, a zero value is required for the **PSDINT** system initialization parameter.

PVDELAY

The **PVDELAY** system initialization parameter specifies the persistent verification delay as a value in the range 0 through 10080 minutes (up to 7 days).

PVDELAY={30|number}

PVDELAY defines how long entries can remain in the signed-on-from lists for those connections for which persistent verification is specified in a connection resource definition. If you specify PVDELAY=0, entries are deleted immediately after use.

For information about the use of PVDELAY, see CICSPlex SM workload management.

QUIESTIM

The **QUIESTIM** system initialization parameter specifies a timeout value for data set quiesce requests.

QUIESTIM={240|number}

In a busy CICSplex, it is possible for the default timeout to expire before the quiesce request has been processed by all the CICS regions, even though there is nothing wrong. If the quiesce operation is not completed when the timeout period expires, SMS VSAM cancels the quiesce. If you find that timeout is occurring too frequently, increase the timeout value.

Specify the timeout value as a number of seconds. The default value is 240 seconds (4 minutes)

The maximum timeout value you can specify is 3600 (1 hour).

RACFSYNC

The **RACFSYNC** system initialization parameter specifies whether CICS listens for type 71 ENF events.

RACFSYNC={YES|NO}

RACF sends a type 71 ENF signal to listeners when a **CONNECT**, **REMOVE**, or **REVOKE** command changes a user's resource authorization. When CICS receives a type 71 ENF event for a user ID, all cached user tokens for the user ID are invalidated, irrespective of the setting of the **USRDELAY** parameter. Subsequent requests from that user ID force a full RACF RACROUTE VERIFY request, which results in a refresh of the user's authorization level. CICS will also make DB2 threads for the associated userid issue a full signon when they are next reused. User tokens for tasks that are currently running are not affected.

Note: Specify the **RACFSYNC=NO** parameter only under direction from IBM Service.

YES CICS listens for type 71 ENF events.

NO CICS does not listen for type 71 ENF events.

Note: In the configuration where type 71 signals are issued for large numbers of users simultaneously, combined with large numbers of connections to DB2, the temporary performance overhead might be significant when the full signon processing across all affected DB2 threads is completed. To reduce the impact of type 71 ENF processing, it is recommended that updates to large numbers of RACF users be made during off-peak periods.

Restrictions: You can specify the **RACFSYNC** parameter only in the system initialization table (SIT), the **PARM** parameter of the **EXEC PGM=DFHSIP** statement, or the SYSIN data set.

RAMAX

The **RAMAX** system initialization parameter specifies the size in bytes of the I/O area allocated for each RECEIVE ANY issued by CICS, in the range 0 through 32767 bytes.

RAMAX={256|number}

If you are using APPC, do not code a value less than 256; otherwise, the results are unpredictable.

For information about coding this parameter, see Setting the size of the receive-any input areas.

RAPOOL

The **RAPOOL** system initialization parameter specifies the number of concurrent receive-any requests that CICS is to process from the z/OS Communications Server for SNA.

RAPOOL={50|value1|(value1,value2,FORCE)}

value1 is the number of fixed request parameter lists (RPLs), receive any control elements (RACEs), and receive any input areas (RAIAs) that are to be generated whether or not CICS uses the high performance option (HPO). *value1*, in the range 1 through 999, is also the number that are active in a non-HPO system; *value2*, in the range 0 through 999, is the number that are active in an HPO system. The default for *value1* in the DFHSIT macro is 50. The default for *value2* is calculated from *value1* as follows:

If value1 = 1, value2 = 1

If value1 ≤ 5, value2 = (value1 minus 1)

If value1 ≥ 6 and ≤ 50, value2 = 5

If value1 > 50, value2 is 10 per cent of value1

Note: Code *value1* equal to or greater than *value2*; if you code *value1* less than *value2*, CICS forces *value2* equal to *value1*.

If you omit the RAPOOL parameter altogether, RAPOOL=(50,5) is assumed. CICS maintains *n* z/OS Communications Server RECEIVE ANYs, where *n* is either the RAPOOL "number active" value, or the MXT value minus the number of active tasks, whichever is the smaller. For example, in a non-HPO system:

If RAPOOL=2, MXT=50, active tasks = 45, then RECEIVE ANY = 2

If RAPOOL=10, MXT=50, active tasks = 45, then RECEIVE ANY = 5

If RAPOOL=10, MXT=50, active tasks = 35, then RECEIVE ANY = 10

In an HPO system:

If RAPOOL=(20,10), MXT=50, active tasks = 45, then RECEIVE ANY = 5

FORCE tells CICS to free up Receive_Any_RPLs if they are stalled. CICS decides that the Receive_Any_RPLs are stalled if all the RA RPLs have been posted but the TCTTE for each one is waiting for a response from a z/OS Communications Server terminal or session for 10 dispatches of the TCP (CSTP) task.

This typically happens only if a protocol error has occurred, and sessions are waiting for a response; for example, to a BID SHUTD request from CICS.

Each session is unbound, the Receive_Any data is lost and the RA RPL is reissued thus allowing z/OS Communications Server activity to continue: Message DFHZC4949 is issued for each session affected.

Consider increasing the size of the RAPOOL before resorting to the use of FORCE.

If FORCE is not specified and a Receive_Any stall occurs, DFHZC2118 is written to the console for each session affected.

If FORCE is specified in the SIT, and RAPOOL is supplied as an override, you must again specify FORCE as otherwise it defaults to FORCE not specified.

The number of RECEIVE ANYs needed depends on the expected activity of the system, the average transaction lifetime, and the MAXTASK value specified. For information about coding this parameter, see the Setting the size of the receive-any pool in Improving performance.

RDSASZE

The **RDSASZE** system initialization parameter specifies the size of the RDSA.

RDSASZE={0K|number}

The default size is 0, indicating that the DSA size can change dynamically. A non-zero value indicates that the DSA size is fixed.

number

specify number as an amount of storage in the range 0 to 16777215 bytes in multiples of 262144 bytes (256 KB). If the size specified is not a multiple of 256 KB, CICS rounds the value up to the next multiple.

You can specify number in bytes (for example, 4194304), or as a whole number of kilobytes (for example, 4096 KB), or a whole number of megabytes (for example, 4 MB).

Restriction: You can specify the RDSAZSE parameter in PARM, SYSIN, or CONSOLE only.

Important: Setting the size of individual dynamic storage areas (DSAs) is not usually necessary and is not recommended. If you specify DSA size values that in combination do not allow sufficient space for the remaining DSAs, CICS fails to initialize. The limit on the storage available for the DSAs in 24-bit storage (below the line) is specified by the DSALIM system initialization parameter. You must allow at least 256K for each DSA in 24-bit storage for which you have not set a size. See DSA size limits.

RENTPGM

The **RENTPGM** system initialization parameter specifies whether you want CICS to allocate the read-only DSAs, RDSA and ERDSA, from read-only key-0 protected storage.

RENTPGM={PROTECT|NOPROTECT}

The permitted values are PROTECT (the default), or NOPROTECT:

PROTECT

CICS obtains the storage for the read-only DSAs from key-0 protected storage.

NOPROTECT
CICS obtains the storage from CICS-key storage, effectively creating two more CICS DSAs (CDSA and ECDSA). This allows programs eligible for the read-only DSAs to be modified by programs that execute in CICS key.

In production CICS regions, RENTPGM=PROTECT provides the correct level of protection for modules in the RDSA and ERDSA. Specifying RENTPGM=NOPROTECT is only appropriate for development regions.

For more information, see Storage protection.

RESP

The **RESP** system initialization parameter specifies the type of request that CICS terminal control receives from logical units.

RESP={FME|RRN}
Valid values are as follows:

FME Function management end is the default.

RRN Reached recovery node.

RESSEC

The **RESSEC** system initialization parameter specifies whether you want CICS to honor the RESSEC option specified on a transaction's resource definition.

RESSEC={ASIS|ALWAYS}
Valid values are as follows:

ASIS CICS honors the RESSEC option defined in a transaction's resource definition. CICS calls its resource security checking routine only when RESSEC(YES) is specified in a transaction resource definition. This is normally a sufficient level of control, because often you will need only to control the ability to execute a transaction.

ALWAYS
CICS overrides the RESSEC option, and always calls its resource security checking routine to issue the appropriate call to the SAF interface.

Use this option only if you need to control or audit all accesses to CICS resources. Using this option can significantly degrade performance.

Restrictions You can specify the RESSEC parameter in the SIT, PARM, or SYSIN only.

RLS

The **RLS** system initialization parameter specifies whether CICS is to support VSAM record-level sharing (RLS).

RLS={NO|YES}
Valid values are as follows:

NO RLS support is not required in this CICS region. Files whose definitions specify RLSACCESS(YES) will fail to open, with an error

indicating that RLS access is not supported. You should not specify RLS=NO if you have files that you want to open in RLS access mode (including the CSD).

YES RLS support is required in this CICS region. During initialization, CICS automatically registers with an SMSVSAM control ACB to enable RLS access to files opened with RLSACCESS(YES).

RLSTOLSR

The **RLSTOLSR** system initialization parameter specifies whether CICS is to include files that are to be opened in RLS mode when calculating the number of buffers, strings, and other resources for an LSR pool.

RLSTOLSR={NO|YES}

CICS performs this calculation only when you have not explicitly defined an LSRPOOL resource definition that corresponds to an LSRPOOLNUM in a file definition. CICS calculates and builds a default LSR pool only when it is opening the first file in LSR mode that references the default pool.

NO CICS is not to include files opened in RLS mode, and which also specify an LSRPOOLNUM, when it is building default LSR pools. Files defined with RLSACCESS(YES) are ignored when CICS is scanning file entries looking for files that specify an LSR pool it is about to build using default values.

If the LSR pools referenced by LSRPOOLNUMs in your file resource definitions are defined explicitly by LSRPOOL resource definitions, you must specify RLSTOLSR=NO.

YES

CICS is to include in its calculation, when building default LSR pools, files that specify both RLSACCESS(YES) and an LSRPOOLNUM.

Note that an LSR pool built including files that are opened in RLS mode is larger than necessary initially. This option is provided to ensure that, if files are later switched to LSR, the LSR pool is adequate for the extra files. You should specify RLSTOLSR=YES only if both of the following conditions are true:

- You do not define LSR pools explicitly, relying instead on CICS obtaining a default set of values for you.
- You have files that are sometimes accessed in RLS mode and sometimes accessed in non-RLS mode (although this is not advised).

The RLSTOLSR parameter is provided to support files that are normally opened in RLS mode, but which can be closed and then switched to LSR mode.

If LSR pools are not defined explicitly using LSRPOOL resource definitions, CICS calculates the resources needed for an LSR pool using default attributes. CICS performs this calculation when opening the first file that specifies an LSR pool that is not explicitly defined. To calculate a default LSR pool, CICS scans all the file entries to count all the files that specify the same LSRPOOLNUM. The size of an LSR pool built dynamically in this way remains fixed until all files that reference the LSR pool are closed. After all files have been closed, another request to open a file with the same LSRPOOLNUM causes CICS to recalculate the size.

If you add files to the system *after* the LSR calculation has been performed there may be insufficient storage available to enable CICS to open a file that specifies a default pool. This situation could occur if files are opened initially in RLS mode and later closed and reopened in LSR mode. There are two ways to ensure that enough resources are built into the LSR pool to support subsequent switches of files from RLS to LSR:

- You can explicitly define LSRPOOL resource definitions that correspond to the LSRPOOLNUMs on file definitions, removing the need for CICS to calculate default values.
- You can specify RLSTOLSR=YES to force CICS to include RLS files when calculating defaults.

RMTRAN

The **RMTRAN** system initialization parameter specifies the name of the transaction that you want an alternate CICS to initiate when logged-on class 1 terminals, which are defined with the attribute RECOVNOTIFY(TRANSACTION) specified, are switched following a takeover.

RMTRAN=({CSGM|name1}[,{CSGM |name2}])

This parameter is applicable only on an alternate CICS region.

If you do not specify a name here, CICS uses the CSGM transaction, the default CICS good morning transaction.

name1 This is the transaction that CICS initiates at terminals that do **not** remain signed-on after the takeover (that is, they are still connected to CICS, but are signed off).

name2 This is the transaction that CICS initiates at terminals that remain signed-on after the takeover. If you specify only name1, CICS uses the CSGM transaction as the default for name2.

If you are using z/OS Communications Server persistent sessions, the name2 transaction is ignored and the name1 transaction is always initiated.

RRMS

The **RRMS** system initialization parameter specifies whether CICS is to register as a resource manager with recoverable resource management services (RRMS).

RRMS={NO|YES}

Valid values are as follows:

NO You do not require RRMS support.

YES You require RRMS support to enable DPL requests to be coordinated by resource recovery services (RRS).

Note: If you specify RRMS=YES, ensure that the DFHRXSVC module is available during CICS initialization. This module, which provides RRMS authorized services, is supplied in the SDFHLINK library. For information about this link list library, see CICS- and CICSPlex SM-supplied modules required in the MVS linklist.

RST

The **RST** system initialization parameter specifies a recoverable service table suffix.

RST={NO|xx|YES}

If you are running CICS with XRF=YES, and you are using DBCTL, you must specify an RST if you want XRF support for DBCTL.

For information about coding the macros for this table, see Recoverable service table (RST).

RSTSIGNOFF

The **RSTSIGNOFF** system initialization parameter specifies whether all users signed-on to the active CICS region are to remain signed-on following a persistent sessions restart or an XRF takeover.

RSTSIGNOFF={NOFORCE|FORCE}

It applies to the following events:

- A persistent sessions restart, where PSDINT=*value* and PSTYPE=SNPS or MNPS are specified, and the restart follows a CICS abnormal or immediate shutdown.
- A persistent sessions restart, where PSDINT=*value* and PSTYPE=MNPS are specified, and terminal sessions are recovered as a result of a z/OS Communications Server restart.
- An XRF takeover, where XRF=YES is specified.

NOFORCE

Do not sign off users, unless FORCE is specified on either:

- The RSTSIGNOFF parameter in the TYPETERM definition referenced by the user's terminal definition.
- The XRFSOFF parameter in the CICS segment of the user's RACF profile.

Thus for a user to remain signed on after a persistent sessions restart or an XRF takeover, NOFORCE must be specified as a system initialization parameter, on the TYPETERM definition, and in the CICS segment.

FORCE

Sign off all users regardless of the options specified on:

- The RSTSIGNOFF attribute in the TYPETERM definition referenced by the user's terminal definition.
- The **XRFSOFF** parameter in the CICS segment of the user's RACF profile.

See The CICS segment for information about user profile options in the CICS segment, and see TYPETERM resources for information about the TYPETERM resource definition.

RSTSIGNTIME

The **RSTSIGNTIME** parameter specifies the timeout delay interval for signon retention during a persistent sessions restart or an XRF takeover.

RSTSIGNTIME={500|decimal-value}

You can specify a 1-to-6 digit time in hours, minutes and seconds, up to the maximum time of 23 hours 59 minutes 59 seconds. If you specify less than six digits, CICS pads the value with leading zeros. Thus a value of 500 is taken as five minutes exactly.

RSTSIGNTIME is counted from the time when CICS failed. Note that the time of failure cannot be determined with complete accuracy.

If you specify NOFORCE on all the appropriate parameters to enable a user to remain signed on, but the persistent sessions restart or XRF takeover takes longer than the specified on the **RSTSIGNTIME** parameter, CICS ensures users do not remain signed on after the delay period expires.

500 Five minutes is the default value.

time This is the time, in the range 0 through 23 hours 59 minutes 59 seconds, during which CICS permits users to remain signed on during a persistent sessions restart or an XRF takeover. The period is measured as follows:

- For a persistent sessions restart, the period is the time from the CICS failure and the time when the user starts working on the terminal. If the specified time expires before the user starts working on the terminal, users signed on at the time CICS failed are not signed on again after restart.
- For an XRF takeover, the period is the time from when the takeover is initiated to the time at which the alternate CICS has completed takeover and is ready to process user transactions. If the takeover takes longer than the specified period, all users signed on at the time the takeover was initiated are signed off.

A value of 0 means there is no timeout delay, and terminals are not signed on after a persistent sessions restart or XRF takeover, which means that RSTSIGNTIME=0 has the same effect as coding RSTSIGNOFF=FORCE.

When XRF is in use with non-XRF-capable terminals, take into account any AUTCONN delay period when setting the value for RSTSIGNTIME. For example, you might need to increase the time specified on RSTSIGNTIME to allow for the delay up to the start of the CXRE transaction imposed by the AUTCONN parameter; otherwise, terminals could be signed off too early.

RUWAPOOL

The **RUWAPOOL** parameter specifies the option for allocating a storage pool the first time a program invoked by Language Environment runs in a task.

RUWAPOOL={NO|YES}

Valid values are as follows:

NO CICS disables the option and provides no RUWA storage pool. Every EXEC CICS LINK to a program that runs under Language Environment results in a GETMAIN for RUWA storage.

YES CICS creates a pool of storage the first time a program invoked by Language Environment runs in a task. This provides an available storage pool that reduces the need to GETMAIN and FREEMAIN run-unit work areas (RUWAs) for every EXEC CICS LINK request.

Note: This applies only to application programs running with the Language Environment run-time option ALL31(ON). RUWAPOOL=YES has no effect on application programs running with the Language Environment run-time option ALL31(OFF).

SDSASZE

The **SDSASZE** system initialization parameter specifies the size of the SDSA.

SDSASZE={0K|number}

The default size is 0, indicating that the DSA size can change dynamically. A non-zero value indicates that the DSA size is fixed.

number

specify number as an amount of storage in the range 0 to 16777215 bytes in multiples of 262144 bytes (256 KB). If the size specified is not a multiple of 256 KB, CICS rounds the value up to the next multiple.

You can specify number in bytes (for example, 4194304), or as a whole number of kilobytes (for example, 4096 KB), or a whole number of megabytes (for example, 4 MB).

Restrictions You can specify the **SDSAZSE** parameter in PARM, SYSIN, or CONSOLE only.

CAUTION:
Setting the size of individual dynamic storage areas (DSAs) is not usually necessary and is not recommended. If you specify DSA size values that in combination do not allow sufficient space for the remaining DSAs, CICS fails to initialize. The limit on the storage available for the DSAs in 24-bit storage (below the line) is specified by the DSALIM system initialization parameter. You must allow at least 256K for each DSA in 24-bit storage for which you have not set a size.

SDTRAN

The **SDTRAN** system initialization parameter specifies the name of the shutdown transaction to be started at the beginning of normal and immediate shutdown.

SDTRAN={CESD|*name_of_shutdown_tran*|NO}

The shutdown transaction enables CICS to shut down in a controlled manner, within a reasonable period of time. For example, you can use it to purge and backout long-running tasks, while ensuring that as many tasks as possible commit or back out cleanly. For information about the CICS-supplied program, DFHCESD, started by the default shutdown transaction, CESD, and how to use it as the basis for your own transaction, see Shutdown assist program (DFHCESD).

Note:

1. The transaction runs under the userid authority of the issuer of the shutdown command.
2. If the program named by the shutdown transaction cannot be loaded, CICS waits indefinitely for all user tasks to complete. *This happens on an immediate, as well as on a normal, shutdown.*

CESD Starts the CICS-supplied program DFHCESD.

name_of_shutdown_transaction

The 1- to 4-character name of your own shutdown transaction.

NO Do not run a shutdown transaction. On a normal shutdown, CICS waits indefinitely for all user tasks to complete.

SEC

The **SEC** system initialization parameter specifies what level of external security you want CICS to use.

SEC={YES|NO}

Valid values are as follows:

YES You want to use full external security. CICS requires the appropriate level of authorization for the access intent: a minimum of READ permission for read intent, and a minimum of UPDATE permission for update intent.

Note: You must also ensure that the default user ID (CICSUSER or another user ID specified on the **DFLTUSER** system initialization parameter) has been defined to RACF.

If command security checking is defined for CICS SP-type commands, then specifying SEC=YES means that the appropriate level of authority is checked for; therefore:

- A check for READ authority is made for **INQUIRE** and **COLLECT** commands.
- A check for UPDATE authority is made for **SET**, **PERFORM**, and **DISCARD** commands.

NO You do not want CICS to use an external security manager. All users have access to all resources, whether determined by attempts to use them or by the **QUERY SECURITY** command. Users are not allowed to sign on or off.

Note: With MRO bind-time security, even if you specify SEC=NO, the CICS region user ID is still sent to the secondary CICS region, and bind-time checking is still carried out in the secondary CICS region. For information about MRO bind-time security, see Security checking using the Query Security command.

Define whether to use RACF for resource level checking by using the **XDCT**, **XFCT**, **XHFS**, **XJCT**, **XPCT**, **XPPT**, **XPSB**, **XRES**, and **XTST** system initialization parameters. Define whether to use RACF for transaction-attach security checking by using the **XTRAN** system initialization parameter. Define whether RACF session security can be used when establishing APPC sessions by using the **XAPPC** system initialization parameter.

For programming information about the use of external security for CICS system commands, see Security checking.

Restrictions You specify the **SEC** parameter in the SIT system initialization parameter, PARM option, or SYSIN control statement.

Note: If you are using preset terminal security and you perform a warm start with SEC=NO and then again with SEC=YES, you must reinstall the terminal definition to preserve the preset user ID that is replaced by the default user ID when security is switched off. See Preset terminal security for details.

SECPRFX

The **SECPRFX** system initialization parameter specifies whether CICS prefixes the resource names in any authorization requests to the external security manager.

SECPRFX={NO|YES|*prefix*}

Valid values are as follows:

NO CICS does not use prefixes on any resource names.

YES CICS prefixes all resource names with the CICS region user ID. This is the user ID under which the CICS job runs. It is one of the following:

- If CICS is a batch job, it is the user ID corresponding to the USER parameter of the CICS JOB statement.
- If CICS is a started task, it is the user ID associated with the name of the started procedure in the RACF ICHRIN03 table.
- If CICS is a started job, it is the user ID specified in the user parameter of the STDATA segment of a STARTED general resource class profile.

For more information, see RACF security overview.

prefix CICS prefixes all resource names with the string you specify. It can be any string of 1 to 8 uppercase alphanumeric characters except NO or YES, and it must start with an alphabetic character.

The SECPRFX parameter is effective only if you specify YES for the SEC system initialization parameter.

Restriction: You can specify the SECPRFX parameter in the SIT, PARM, or SYSIN only.

SIT

The **SIT** system initialization parameter specifies the suffix, if any, of the system initialization table that you want CICS to load at the start of initialization.

SIT=*xx*

If you omit this parameter, CICS loads the unsuffixed table, DFHSIT, which is pregenerated with all the default values. This default SIT (shown in The default system initialization table) is in CICSTS54.CICS.SDFHAUTH, and its source, named DFHSIT$$, is in CICSTS54.CICS.SDFHSAMP.

Restriction: You can specify the system initialization parameter anywhere in PARM or SYSIN, or as the **first** parameter entry at the CONSOLE.

SKRxxxx

The **SKRxxxx** system initialization parameter specifies that a single-keystroke-retrieval operation is required.

SKRxxxx='page-retrieval-command'

'xxxx' specifies a key on the 3270 keyboard which, during a page retrieval session, is to be used to represent a page retrieval command. The valid keys you can specify as a system initialization parameter or as an override are PA1 through PA3, and PF1 through PF24. Therefore it is possible to specify up to 27 keys in total.

The 'page-retrieval-command' value represents any valid page retrieval command, and must be enclosed in apostrophes. It is concatenated to the character string coded in the PGRET parameter. The combined length must not exceed 16 characters.

Note: If full function BMS is used, all PA keys and PF keys are interpreted for page retrieval commands, even if some of these keys are not defined.

SNPRESET

The **SNPRESET** system initialization parameter specifies whether preset userid terminals share a single access control environment element (ACEE) that is associated with the userid, or a unique ACEE for every terminal.

SNPRESET={UNIQUE|SHARED}

UNIQUE
When signing on a preset userid terminal, the ACEE will be built with entry port information. Every preset terminals will have a unique ACEE that is associated with the userid and terminal. This is the default.

If you audit data that is based on the terminal of a preset userid, use SNPRESET=UNIQUE.

SHARED
When signing on a preset userid terminal, the ACEE will be built without entry port information. All preset terminals with the same userid will use the same ACEE.

If you do not need information that is based on the terminal of a preset userid, you can save storage by selecting SNPRESET=SHARED.

Note: Set SNPRESET=SHARED only when you have a large number of preset userid terminals and do not have security definitions that rely on the netname of these terminals.

In the event of a security violation with SNPRESET=SHARED, the netname of the terminal will not appear in the DFHXS1111 message.

SNSCOPE

The **SNSCOPE** system initialization parameter specifies whether a userid can be signed on to CICS more than once, within the scope of a single CICS region, a single MVS image, and a sysplex.

SNSCOPE={NONE|CICS|MVSIMAGE|SYSPLEX}
The signon SCOPE is enforced with the MVS ENQ macro where there is a limit on the number of outstanding MVS ENQs per address space. If this limit is exceeded, the MVS ENQ is rejected and CICS is unable to detect if the user is already signed on. When this happens, the signon request is rejected with message DFHCE3587. You can use the ISGADMIN macro to set or reset the MVS ENQ limit. For more information, see the *z/OS MVS Programming: Authorized Assembler Services Reference*.

NONE
Each user ID can be used to sign on for any number of sessions on any CICS region. This is the compatibility option, providing the same signon scope as in releases of CICS before CICS Transaction Server for z/OS, Version 5 Release 4 .

CICS Each user ID can be signed on once only in the same CICS region. A signon request is rejected if the userid is already signed on to the same CICS region. However, the user ID can be used to signon to another CICS region in the same, or another, MVS image.

MVSIMAGE
Each userid can be signed on once only, and to only one of the set of CICS regions in the same MVS image that also specify SNSCOPE=MVSIMAGE. A signon request is rejected if the user is already signed on to another CICS region in the same MVS image.

SYSPLEX
Each user ID can be signed on once only, and to only one of the set of CICS regions within an MVS sysplex that also specify SNSCOPE=SYSPLEX. A signon is rejected if the user is already signed on to another CICS region in the same MVS sysplex.

The signon scope (if specified) applies to all user IDs signing on by an explicit signon request (for example, by an EXEC CICS SIGNON command or the CESN transaction). SNSCOPE is restricted to users signing on at local terminals, or signing on after using the CRTE transaction to connect to another system.

Signon scope specified by SNSCOPE does not apply to:

- Non-terminal users.
- The CICS default userid, specified by the **DFLTUSER** system initialization parameter.
- Preset user IDs, specified in the USERID option of the **DEFINE TERMINAL** command.
- User IDs for remote users, received in attach headers.
- User IDs for link security. For information about which userid is used for link security on a specific connection, see Security-related system initialization parameters.
- The user ID specified on the **PLTPIUSR** system initialization parameter.
- The CICS region user ID.

Restrictions You can specify the **SNSCOPE** parameter in the SIT, PARM, or SYSIN only.

SOTUNING

The **SOTUNING** system initialization parameter specifies whether performance tuning for HTTP connections will occur to protect CICS from unconstrained resource demand.

For more information about performance tuning for HTTP connections, see CICS HTTP support: Performance and tuning.

SOTUNING={YES|520}
Valid values are as follows:

YES

As a region becomes overloaded, CICS will pause listening for new HTTP connection requests. While new connection requests are not being accepted, those pending requests will queue outside of CICS in the TCP/IP backlog queue. This backlog queue will increase when the region is over capacity, allowing feedback to TCP/IP port sharing and Sysplex Distributor, promoting a balanced sharing of workload with other regions that are sharing the same IP endpoint.

Because requests are no longer queuing in CICS, MXT will not be exceeded by a surge of HTTP requests, but the number of times MXT

is reached might greatly increase. As CICS processes requests from the backlog queue, the number of active tasks will fluctuate, potentially reaching MXT multiple times.It does not mean that MXT needs to be increased, but indicates that the current value is correctly protecting CICS from unconstrained resource demand.

If the region continues to become overloaded, it will close existing persistent connections after their next request completes and new connections will be marked as non-persistent until the region is no longer under stress.

In addition, CICS will also periodically close persistent connections to allow more efficient sharing of workload across regions that share IP endpoints.

520

The CICS TS V5.2 behavior is used, and performance tuning for HTTP connections does not take place.

Note: If sharing IP endpoints, ensure that all regions have the same **SOTUNING** value or uneven loading might occur.

SPCTR

The **SPCTR** system initialization parameter specifies the level of special tracing required for CICS as a whole.

SPCTR={(1,2)|(1[,2][,3])|ALL|OFF}

Specifies the level of special tracing for all CICS components used by a transaction, terminal, or both. If you want to set different tracing levels for an individual component of CICS, use the **SPCTRxx** system initialization parameter.

It is possible to select up to 4 levels of tracing using this parameter. However, most CICS components only use levels 1, 2, and 3, and some do not have trace points at all these levels. The exception is the SM component (storage manager domain), which also has level 4 tracing. Use the **SPCTRxx** system initialization parameter to set special tracing levels above 3 for this component.

number

The level numbers for the level of special tracing you want for all CICS components.

ALL Enables the special tracing facility for all available levels.

OFF Disables the special tracing facility.

SPCTRxx

The **SPCTRxx** system initialization parameter specifies the level of special tracing for a particular CICS component used by a transaction, terminal, or both.

SPCTRxx={(1,2)|(1[,2][,3][,4])|ALL|OFF}

You identify the component by coding a value for *xx* in the keyword. You code one SPCTRxx keyword for each component that you want to define selectively. For a CICS component that is being specially traced and that does not have its trace level set by SPCTRxx, the trace level is that set by **SPCTR** (which, in turn, defaults to (1,2)). The CICS component codes that you can specify for *xx* on the SPCTRxx keyword are shown in the following table:

Code	Component name
AP	Application domain
BA	Business application manager
BF*	Built-in function
BM*	Basic mapping support
BR*	3270 bridge
CP*	Common programming interface
DC*	Dump compatibility layer
DD	Directory manager domain
DH	Document handling domain
DI*	Data interchange
DM	Domain manager domain
DP	Debugging profiles domain
DS	Dispatcher domain
DU	Dump domain
EC*	Event capture and emission
EI*	Exec interface
EJ	Enterprise Java domain
EM	Event manager domain
EP	Event processing domain
FC*	File control
GC	Global catalog domain
IC*	Interval control
IE	ECI over TCP/IP domain
IS*	ISC or IRC
KC*	Task control
KE	Kernel
LC	Local catalog domain
LD	Loader domain
LG	Log manager domain
LM	Lock domain
ME	Message domain
ML	Markup language domain
MN	Monitoring domain
MP	Managed platform domain
NQ	Enqueue domain
OT	Object transaction domain
PA	Parameter domain
PC*	Program control
PG	Program manager domain
PI	Pipeline domain
PT	Partner domain

Code	Component name
RA	Resource manager adapters
RI*	Resource manager interface (RMI)
RL	Resource life-cycle domain
RM	Recovery manager domain
RS	Region status domain
RX	RRS-coordinated EXCI domain
RZ	Request streams domain
SC*	Storage control
SH	Scheduler services domain
SJ	JVM domain
SM	Storage manager domain
SO	Sockets domain
ST	Statistics domain
SZ*	Front End Programming Interface
TC*	Terminal control
TD*	Transient data
TI	Timer domain
TR	Trace domain
TS	Temporary storage domain
UE*	User exit interface
US	User domain
WB	Web domain
WU™	CICS Management Client Interface (CMCI) domain
W2	Web 2.0 domain
XM	Transaction manager domain
XS	Security manager domain

number
The level numbers for the level of special tracing you want for the required CICS component. You can use level numbers 1, 2, 3, and 4, depending on the component.

Most CICS components only use levels 1, 2 and 3, and some do not have trace points at all these levels. The exception is the SM component (storage manager domain) that has level 4 tracing. This level of tracing is intended for IBM field engineering staff.

ALL You want all the available levels of special CICS tracing switched on for the specified component.

OFF Switches off all levels of special CICS tracing for the CICS component indicated by xx.

For details of using trace, see CICS trace.

Restrictions You can specify the **SPCTRxx** parameter in PARM, SYSIN, or CONSOLE only.

SPOOL

The **SPOOL** system initialization parameter specifies whether the system spooling interface is required.

SPOOL={NO|YES}
Valid values are as follows:

NO The system spooling interface is not required.

YES The system spooling interface is required.

The CICS spool interface uses the MVS exit, IEFDOIXT, which is provided in the SYS1.LINKLIB library. For further information about the MVS exit IEFDOIXT, see z/OS MVS Installation Exits.

SRBSVC

The **SRBSVC** system initialization parameter specifies the number that you have assigned to the CICS type 6 SVC.

SRBSVC={215|number}
The default number is 215.

For information on changing the SVC number, see Installing the CICS SVCs in Installing. A CICS type 6 SVC with the specified (or default) number must have been link-edited with the system nucleus.

SRT

The **SRT** system initialization parameter specifies the system recovery table suffix.

SRT={1$|YES|NO|*xx*}
If SRT=YES is coded, the default DFHSRT1$ table is used.

Restriction: SRT=YES can only be specified when assembling the SIT table; it cannot be specified as an override parameter.

If SRT=NO is coded, the system recovery program (DFHSRP) does not attempt to recover from a program check or from an operating system abend. However, CICS issues ESPIE macros to intercept program checks to perform clean up operations before CICS terminates. Therefore, you must provide a SRT if you require recovery from either program checks or abnormal terminations, or both. For information about coding the macros for this table, see SRT - system recovery table.

SRVERCP

The **SRVERCP** system initialization parameter specifies the default server code page to be used by the DFHCNV data conversion table but only if the SRVERCP parameter in the DFHCNV macro is set to SYSDEF.

SRVERCP={037|*code page*}
The *code page* is a field of up to 8 characters and can take the values supported by the SRVERCP parameter in the DFHCNV macro. For the list of valid code pages, see CICS-supported conversions. The default is 037.

SSLCACHE

The **SSLCACHE** system initialization parameter specifies whether session IDs for SSL sessions are to be cached locally or at sysplex level for reuse by the CICS region. The SSL cache allows CICS to perform partial handshakes with clients that it has previously authenticated.

SSLCACHE={CICS|SYSPLEX}

CICS The SSL environment for the CICS region includes a local cache of session IDs for SSL sessions between CICS and clients. z/OS System SSL manages the SSL environment. This cache is replaced by a new cache when the PERFORM SSL REBUILD command is issued.

SYSPLEX
A cache of SSL sessions is held at sysplex level for multiple CICS regions, in the coupling facility. This cache is not affected when the PERFORM SSL REBUILD command is issued. Sysplex caching is only allowed if multiple CICS socket-owning regions accept SSL connections at the same IP address. You must activate the z/OS System SSL started task GSKSRVR to have a sysplex cache.

SSLDELAY

The **SSLDELAY** system initialization parameter specifies the length of time in seconds for which CICS retains session ids for secure socket connections.

SSLDELAY={600|*number*}
Session ids are tokens that represent a secure connection between a client and an SSL server.

While the session id is retained by CICS within the SSLDELAY period, CICS can continue to communicate with the client without the significant overhead of an SSL handshake. The value is a number of seconds in the range 0 through 86400.

START

The **START** system initialization parameter specifies the type of start for the system initialization program.

START=({AUTO|INITIAL|COLD|STANDBY}[,ALL])
The value specified for START, or the default of AUTO, becomes the default value for each resource.

AUTO
CICS performs a warm, emergency, cold or initial start, according to the status of two control records on the global catalog:

- The recovery manager (RM) control record written by the previous execution of CICS
- The RM autostart override record written by a run of the recovery manager utility program, DFHRMUTL

Note: If the global catalog does *not* contain the RM control record:

- If it contains an RM autostart override record with option AUTOINIT, CICS performs an initial start.
- If it does not contain an RM autostart override record with option AUTOINIT, CICS does not start.

If you code START=AUTO, you must do one of the following:

- Provide the global catalog and system log from the previous execution of CICS. For an emergency restart to be successful, you must also have coded an activity keypoint value (see the "AKPFREQ" on page 15 parameter) on the previous execution of CICS.
- Provide a global catalog against which you have run the DFHRMUTL utility program, specifying SET_AUTO_START=AUTOINIT.

You may choose to leave the START parameter set to AUTO for all types of startup other than XRF standby, and use the DFHRMUTL program to reset the startup mode to COLD or INITIAL when necessary, using SET_AUTO_START=AUTOCOLD or SET_AUTO_START=AUTOINIT, respectively. For information about the DFHRMUTL utility program, see Recovery manager utility (DFHRMUTL).

INITIAL

The status of CICS resource definitions saved in the global catalog at the previous shutdown is ignored, and all resource definitions are reinstalled, either from the CSD or CICS control tables.

You should rarely need to specify START=INITIAL; if you want to reinstall definitions of local resources from the CSD, use START=COLD instead.

Examples of times when an initial start is necessary are:

- When bringing up a new CICS system for the first time.
- After a serious software failure, when the system log has been corrupted.
- If the global catalog is cleared or initialized.
- When you want to run CICS with a dummy system log. (If the system log is defined as a dummy, it is ignored.)

COLD

The status of CICS resource definitions saved in the global catalog at the previous shutdown is ignored, and all resource definitions (except those for the system log) are reinstalled, either from the CSD or CICS control tables.

Resynchronization information in the global catalog relating to remote systems or to RMI-connected resource managers is preserved. The CICS system log is scanned during startup, and information regarding unit of work obligations to remote systems, or to non-CICS resource managers (such as DB2) connected through the RMI, is preserved. (That is, any decisions about the outcome of local UOWs, needed to allow remote systems or RMI resource managers to resynchronize their resources, are preserved.)

Note that, on a cold start, the following are *not* preserved:

- Updates to *local* resources that were not fully committed or backed out during the previous execution, *even if the updates were part of a distributed unit of work*.
- Resynchronization information for remote systems connected by LU6.1 links, or for earlier releases of CICS systems connected by MRO.

- Any program LIBRARY definitions that had been dynamically defined. Only the static DFHRPL concatenation will remain, together with any LIBRARY definitions in the grouplist specified at startup or installed via BAS at startup.

If you want to reinstall resource definitions from the CSD, use START=COLD rather than START=INITIAL.

STANDBY
Coding START=STANDBY, but only when you have also specified XRF=YES, defines this CICS as the alternate CICS region in an XRF pair. In other words, you **must** specify START=STANDBY for the system that starts off as the alternate. (To start an active CICS region, specify AUTO or COLD, as you would without XRF.)

(option,ALL)
The ALL option is a special option you can use on the START parameter when you supply it as a system initialization parameter at CICS startup; you cannot code it in the SIT. If you specify START=(AUTO,ALL), CICS initializes all resources according to the type of startup that it selects (warm, emergency, initial, or cold). The ALL option overrides any individual settings in other system initialization parameters.

However, if you do not use the ALL option, you can individually cold start those resources that have a COLD operand. For details of resources that have a COLD option, see Defining CICS resource table and module keywords.

Restrictions You can specify START=(option,ALL) in PARM, SYSIN, or CONSOLE only.

For more information about the types of CICS startup, see Controlling start and restart.

STARTER

The **STARTER** system initialization parameter specifies whether the generation of starter system modules (with $ and # suffixes) is permitted, and various MNOTES are suppressed.

STARTER={NO|YES}
Use this parameter only when service is being performed on starter system modules.

Restriction: You can specify the **STARTER** parameter in the SIT only.

STATEOD

The **STATEOD** system initialization parameter specifies the end-of-day time in the format hhmmss.

STATEOD={0|hhmmss}
The default is 0, which is midnight.

End-of-day time is expressed in local time and must be in the range 00:00:00-23:59:59. That is, the hh value cannot exceed 23, and the mm and ss values can be specified in the range 00 to 59. If you leave out leading zeros, the DFHSIT macro inserts them (for example, 100 becomes 000100—that is, 1 minute 00 seconds past midnight).

This parameter is the equivalent of the ENDOFDAY option on the CEMT and EXEC CICS SET STATISTICS command, which you can use to modify the value set by STATEOD.

STATINT

The **STATINT** system initialization parameter specifies the recording interval for system statistics in the format hhmmss.

STATINT={010000|*hhmmss* }
The default is 1 hour.

The interval must be at least one minute and cannot be more than 24 hours. The minutes and seconds part of the value can be specified in the range 00 to 59. If you leave out leading zeros, the DFHSIT macro inserts them. For example, 3000 becomes 003000 (that is, an interval of 30 minutes).

This parameter is the equivalent of the INTERVAL option on the CEMT and **EXEC CICS SET STATISTICS** command, which you can use to modify the value set by STATINT.

STATRCD

The **STATRCD** system initialization parameter specifies the interval statistics recording status at CICS initialization.

STATRCD={OFF|ON}
This status is recorded in the CICS global catalog for use during warm and emergency restarts. Statistics collected are written to the SMF data set.

OFF Interval statistics are not collected (no action is taken at the end of an interval).

End-of-day statistics are collected at the logical end of day and on shutdown. Unsolicited statistics are written to SMF when resources are discarded or closed.

ON Interval statistics are collected.

On a cold start of a CICS region, interval statistics are recorded by default at hourly intervals. All intervals are timed using the end-of-day time as a base starting time (not CICS startup time). The default end-of-day time is midnight, so the default settings result in collections at 00.00, 01.00, 02.00, 03.00, and so on, regardless of the time that you start CICS.

On a warm or emergency restart, the statistics recording status is restored from the CICS global catalog.

You can change the statistics recording status at any time in the following ways:

- During a warm or emergency restart by coding the **STATRCD** system initialization parameter.
- While CICS is running by using the CEMT or EXEC CICS SET STATISTICS command.

Whatever the value of the **STATRCD** system initialization parameter, you can ask for requested statistics and requested reset statistics to be collected. You can get statistics “on demand” for all, or for specified, resource types by using CEMT PERFORM STATISTICS or the **PERFORM STATISTICS** command. See and PERFORM STATISTICS.

The period covered for statistics requested in this way is from the last reset time up to the time that you issue the PERFORM STATISTICS command. The last reset time is one of the following:

- The beginning of the current interval.
- The logical end-of-day collection time.
- The time that you last issued a SET or PERFORM STATISTICS command specifying the RESETNOW option.

For information about the statistics utility program DFHSTUP, see Statistics utility program (DFHSTUP).

STGPROT

The **STGPROT** system initialization parameter specifies whether you want storage protection to operate in the CICS region.

STGPROT={NO|YES}

The permitted values are YES (the default), or NO.

YES If you specify YES, or allow this system initialization parameter to default, CICS operates with storage protection, and observes the storage keys and execution keys that you specify in various system and resource definitions.

NO If you specify NO, CICS does not operate any storage protection.

The **STGPROT** system initialization parameter affects the storage key for the following CICS dynamic storage areas (DSAs):

- UDSA
- SDSA
- EUDSA
- ESDSA
- GUDSA
- GSDSA

When CICS operates with storage protection, the storage for these DSAs is allocated from user-key storage. When CICS operates without storage protection, the storage for these DSAs is allocated from CICS-key storage.

The **STGPROT** system initialization parameter does not affect the storage key for the following CICS DSAs:

- RDSA. The storage for this DSA is affected by the setting for the **RENTPGM** system initialization parameter.
- ERDSA. The storage for this DSA is affected by the setting for the **RENTPGM** system initialization parameter.
- CDSA, ECDSA, ETDSA, and GCDSA. The storage for these DSAs is always allocated from CICS-key storage.

For more information, see Storage protection and CICS dynamic storage areas.

STGRCVY

The **STGRCVY** system initialization parameter specifies whether CICS should try to recover from a storage violation.

STGRCVY={NO|YES}

Valid values are as follows:

NO CICS does not try to repair any storage violation that it detects.

YES CICS tries to repair any storage violation that it detects.

In both cases, CICS continues unless you have specified in the dump table that CICS should terminate.

In normal operation, CICS sets up four task-lifetime storage subpools for each task. Each element in the subpool starts and ends with a 'check zone' that includes the subpool name. At each freemain, and at end-of-task, CICS checks the check zones and abends the task if either has been overwritten.

Terminal input-output areas (TIOAs) have similar check zones, which are set up with identical values. At each freemain of a TIOA, CICS checks the check zones and abends the task if they are not identical.

If you specify **STGRCVY**(YES), CICS resets the check zones correctly and the task continues running.

If you specify **STGRCVY**(NO), CICS abends the task if it is still running. The storage is not reusable and is not returned to the DSA for the remainder of the CICS cycle. If an error is detected when the task ends, no abend is issued. Any sync point that has taken place could save data that is corrupted.

STNTR

The **STNTR** system initialization parameter specifies the level of standard tracing required for CICS as a whole.

STNTR={1|(1[,2][,3])|ALL|OFF}

It is possible to use this parameter to select up to 32 levels of tracing. However, most CICS components use only levels 1, 2 and 3, and some do not have trace points at all these levels.

The exception is the SM component (storage manager domain). The SM component also has level 4 tracing. If you need to set standard tracing levels above 3 for this component, use the **STNTRxx** system initialization parameter, rather than the **STNTR** system initialization parameter

CAUTION:
Before you globally activate tracing levels 3 and ALL, which sets these tracing levels for the SM component, read the warnings in the information about the STNTRxx system initialization parameter.

number
Code the level numbers for the level of standard tracing you want for all CICS components. The options are: 1, (1,2), or (1,2,3). The default, 1, specifies standard tracing for level 1 for all CICS components.

ALL Enables standard tracing for all levels.

OFF Disables standard tracing.

For information about the differences between special and standard CICS tracing, see CICS trace.

STNTRxx

The **STNTRxx** system initialization parameter specifies the level of standard tracing you require for a particular CICS component.

STNTRxx={1|(1[,2][,3][,4])|ALL|OFF}

You identify the component by coding a value for *xx* in the keyword. You code one STNTRxx keyword for each component you want to define selectively. For

a CICS component that is being specially traced, and that does not have its trace level set by STNTRxx, the trace level is that set by STNTR (which, in turn, defaults to 1). You can select up to three levels of tracing, but some CICS components do not have trace points at all these levels.

The CICS component codes that you can specify for *xx* on this STNTRxx keyword are shown in the following table:

Code	Component name
AP	Application domain
BA	Business application manager
BF*	Built-in function
BM*	Basic mapping support
BR*	3270 bridge
CP*	Common programming interface
DC*	Dump compatibility layer
DD	Directory manager domain
DH	Document handling domain
DI*	Data interchange
DM	Domain manager domain
DP	Debugging profiles domain
DS	Dispatcher domain
DU	Dump domain
EC*	Event capture and emission
EI*	Exec interface
EJ	Enterprise Java domain
EM	Event manager domain
EP	Event processing domain
FC*	File control
GC	Global catalog domain
IC*	Interval control
IE	ECI over TCP/IP domain
IS*	ISC or IRC
KC*	Task control
KE	Kernel
LC	Local catalog domain
LD	Loader domain
LG	Log manager domain
LM	Lock domain
ME	Message domain
ML	Markup language domain
MN	Monitoring domain
MP	Managed platform domain
NQ	Enqueue domain

Code	Component name
OT	Object transaction domain
PA	Parameter domain
PC*	Program control
PG	Program manager domain
PI	Pipeline domain
PT	Partner domain
RA	Resource manager adapters
RI*	Resource manager interface (RMI)
RL	Resource life-cycle domain
RM	Recovery manager domain
RS	Region status domain
RX	RRS-coordinated EXCI domain
RZ	Request streams domain
SC*	Storage control
SH	Scheduler services domain
SJ	JVM domain
SM	Storage manager domain
SO	Sockets domain
ST	Statistics domain
SZ*	Front End Programming Interface
TC*	Terminal control
TD*	Transient data
TI	Timer domain
TR	Trace domain
TS	Temporary storage domain
UE*	User exit interface
US	User domain
WB	Web domain
WU	CICS Management Client Interface (CMCI) domain
W2	Web 2.0 domain
XM	Transaction manager domain
XS	Security manager domain

number

The level numbers for the level of standard tracing you want for the CICS component indicated by xx. Level numbers 1, 2, 3 and 4 can be used, depending on the component.

Most CICS components only use levels 1, 2, and 3, and some do not have trace points at all these levels. The exceptions are the SM (storage manager domain) and SJ (JVM server) components, which also have level 4 tracing. This level of tracing is intended for IBM field engineering staff.

Attention: Selecting tracing levels 3, 4, or ALL for standard tracing for the storage manager (SM) component, or the temporary storage domain (TS), degrades the performance of your CICS region. This is because options 3, 4, and ALL turn on trace flags that are used by the SM domain for field engineering purposes.

SM trace flag 3 deactivates the quickcell mechanism, and SM trace flag 4 forces subpool element chaining on every CICS subpool. Furthermore, after these settings have been activated during system initialization, they cannot be unset, either through a PLTPI program or by using the CETR trace transaction, because they are not used for tracing as such. Thus, a significant performance overhead is incurred if these storage manager trace levels are selected for standard tracing.

For information about the effects of trace levels 3 and 4, see CICS trace.

ALL You want all the available levels of standard tracing switched on for the specified component.

Attention: Selecting ALL for standard tracing for the storage manager (SM) component, or the temporary storage domain (TS), degrades the performance of your CICS region. This is because ALL switches on trace flags that are used by the SM domain for field engineering purposes.

Selecting ALL for standard tracing for the JVM domain (SJ) component is not recommended. JVM trace can produce a large amount of output, so activate JVM trace only for special transactions (using the **SPCTRSJ** system initialization parameter), rather than turning it on globally for all transactions.

OFF Switches off all levels of standard CICS tracing for the CICS component indicated by xx.

Restriction: You can specify the **STNTRxx** parameter in PARM, SYSIN, or CONSOLE only.

SUBTSKS

The **SUBTSKS** system initialization parameter specifies the number of task control blocks (TCBs) you want CICS to use for running tasks in concurrent mode.

SUBTSKS={0|1}

Specifies whether there is to be a concurrent mode TCB so that CICS can perform management functions as system subtasks.

0 If you specify 0 (the default), CICS runs under the following two TCBs:

- The quasi-reentrant mode TCB. CICS runs all user applications under this TCB.
- The resource-owning mode TCB. CICS runs tasks that open and close files under this TCB.

1 If you specify 1, CICS runs under the two TCBs listed previously, and uses an additional TCB, a concurrent mode TCB, to perform system subtasking.

SUFFIX

The **SUFFIX** system initialization parameter specifies the last two characters of the name of this system initialization table.

SUFFIX=*xx*
The first 6 characters of the name of the SIT are fixed as DFHSIT. You can specify the last two characters of the name, using the SUFFIX parameter. Because the SIT does not have a TYPE=INITIAL macro statement like other CICS resource control tables, you specify its SUFFIX on the TYPE=CSECT macro statement.

The suffix allows you to have more than one version of the SIT. Any one or two characters (other than NO and DY) are valid. You select the version of the table to be loaded into the system during system initialization by coding SIT=xx in the PARM parameter or the SYSIN data set. (You can, in some circumstances, specify the SIT using the system console, but this is not recommended.)

Restriction: You can specify the **SUFFIX** parameter in the SIT only.

SYDUMAX

The **SYDUMAX** system initialization parameter specifies the limit on the number of system dumps that can be taken per dump table entry.

SYDUMAX={999|number}
If this number is exceeded, subsequent system dumps for that particular entry will be suppressed. The **SYDUMAX** parameter applies for new or added system dump codes. It does not override the limit on the number of system dumps for existing dump table entries.

number
A number in the range 0 through 999. The default, 999, enables an unlimited number of dumps to be taken.

SYSIDNT

The **SYSIDNT** system initialization parameter specifies a 1- to 4-character name that is known only to your CICS region.

SYSIDNT={CICS|name}
If your CICS region also communicates with other CICS regions, the name you choose for this parameter to identify your local CICS region must not be the same name as an installed CONNECTION resource definition for a remote region.

The value for SYSIDNT, whether specified in the SIT or as an override, can only be updated on a cold start. After a warm start or emergency restart, the value of SYSIDNT is that specified in the last cold start.

For information about the SYSIDNT of a local CICS region, see The local CICS region name.

SYSTR

The **SYSTR** system initialization parameter specifies the setting of the master system trace flag.

SYSTR={ON|OFF}

Valid values are as follows:

ON The master trace flag is set, causing CICS to write trace entries of system activity for the individual CICS components. Trace entries are captured and written only for those components for which the trace level is 1 or greater, as specified on the STNTR or STNTRxx system initialization parameters. Entries are written only to those trace destinations that are active.

OFF The master trace flag is unset, and no standard trace entries are captured, overriding any trace levels specified by the STNTR or STNTRxx system initialization parameters.

Note: Setting the master trace flag OFF affects only standard tracing and has no effect on special tracing, which is controlled separately by SPCTR or SPCTRxx trace levels and the CETR transaction.

See CICS trace for more information about controlling CICS trace.

TAKEOVR

The **TAKEOVR** system initialization parameter specifies the action to be taken by the alternate CICS region, following the apparent loss of the surveillance signal in the active CICS region.

TAKEOVR={MANUAL|AUTO|COMMAND} (alternate)

Use this parameter in the SIT for an alternate CICS region. This parameter also specifies the level of operator involvement.

If both active and alternate CICS regions are running under different MVS images in the same sysplex, and an MVS failure occurs in the MVS image of the active CICS region, the TAKEOVR option is overridden.

- If the MVS images are running in a PR/SM environment, CICS XRF takeover to an alternate CICS region on a separate MVS image completes without the need for any operator intervention.
- If the MVS images are not running in a PR/SM environment, the CICS takeover is still initiated automatically, but needs operator intervention to complete, because XCF outputs a WTOR (IXC402D). Sysplex partitioning does not complete until the operator replies to this message, and CICS waits for sysplex partitioning to complete before completing the XRF takeover.

MANUAL

The operator is asked to approve a takeover if the alternate CICS region cannot detect the surveillance signal of the active CICS region.

The alternate CICS region does not ask the operator for approval if the active CICS region signs off abnormally, or if there is an operator or program command for takeover. In these cases, there is no doubt that the alternate CICS region should take over, and manual involvement by the operator would be an unnecessary overhead in the takeover process.

You could use this option, for instance, to ensure manual takeover of a master or coordinator region in MRO.

AUTO

No operator approval, or intervention, is needed for a takeover.

COMMAND
Takeover occurs only when a CEBT PERFORM TAKEOVER command is received by the alternate CICS region. It ensures, for instance, that a dependent alternate CICS region, in MRO, is activated only if it receives the command from the operator, or from a master or coordinator region.

TBEXITS

The **TBEXITS** system initialization parameter specifies the names of your backout exit programs for use during emergency restart backout processing.

TBEXITS=([name1][,name2][,name3] [,name4][,name5][,name6])
The order in which you code the names is significant. If you do not want to use all the exits, code commas in place of the names you omit. For example:

```
TBEXITS=(,,EXITF,EXITV)
```

The program names for *name1* through *name6* apply to global user exit points as follows:

- *name1* and *name2* are the names of programs to be invoked at the XRCINIT and XRCINPT global user exit points (but note that XRCINIT and XRCINPT are invoked only for user log records).
- *name3* is the name of the program to be invoked at the file control backout failure global user exit point, XFCBFAIL.
- *name4* is the name of the program to be invoked at the file control logical delete global user exit point, XFCLDEL.
- *name5* is the name of the program to be invoked at the file control backout override global user exit point, XFCBOVER.
- *name6* is the name of the program to be invoked at the file control backout global user exit point, XFCBOUT.

This exit is invoked (if required) during backout of a unit of work, regardless of whether the backout is taking place at emergency restart, or at any other time.

The XFCBFAIL, XFCLDEL, and XFCBOVER global user exit programs are enabled on all types of CICS start if they are named on the TBEXITS system initialization parameter.

If no backout exit programs are required, you can do one of the following:

- Omit the **TBEXITS** system initialization parameter altogether.
- Code the parameter as TBEXITS=(,,,,,).

TCP

The **TCP** system initialization parameter specifies whether the pregenerated non-z/OS Communications Server terminal control program, DFHTCP, is to be included.

TCP={YES|NO}
You must code TCP=YES if you intend using card reader/line printer (sequential) devices.

TCPIP

The **TCPIP** system initialization parameter specifies whether CICS TCP/IP services are to be activated at CICS startup.

TCPIP={YES|NO}

YES The IPIC, HTTP, and ECI over TCP/IP services can process work. This is the default value.

NO The IPIC, HTTP, and ECI over TCP/IP services cannot be enabled.

For IPIC, you must specify TCPIP=YES and ISC=YES.

Note: The **TCPIP** system initialization parameter affects only CICS internal TCP/IP Services defined by TCPIPSERVICE resource definitions. It has nothing to do with the TCP/IP socket interface for CICS feature of z/OS Communications Server.

TCSACTN

The **TCSACTN** system initialization parameter specifies the required action that CICS terminal control should take if the terminal control shutdown wait threshold expires.

TCSACTN={NONE|UNBIND|FORCE}

For details of the wait threshold, see the **TCSWAIT** system initialization parameter. **TCSACTN** only takes effect when **TCSWAIT** is coded with a value in the range 1 through 99. This is a global default action. On a terminal-by-terminal basis, you can code a DFHZNEP routine to override this action.

NONE

No action is taken. This can be overridden by DFHZNEP.

- To report hung terminals and not attempt to force-close them specify the TCSWAIT=mm (with an appropriate time interval) and TCSACTN=NONE system initialization parameters.
- To attempt to force-close some hung terminals, and only report others, specify the TCSWAIT=mm (with an appropriate time interval) and TCSACTN=NONE system initialization parameters, and code a DFHZNEP routine that selects the required terminals and sets TWAOCN on for them.

UNBIND

CICS terminal control attempts to close the session by issuing a z/OS Communications Server VTAM CLSDST and sending an SNA UNBIND command to the hung terminal. This can be overridden by DFHZNEP.

- To attempt to force-close all hung terminals specify the TCSWAIT=mm (with an appropriate time interval) and TCSACTN=UNBIND system initialization parameters.

FORCE

CICS terminal control attempts to forceclose the CICS z/OS Communications Server ACB if there are any hung terminals or parallel connection sessions. All CICS z/OS Communications Server terminals and sessions are released and CICS normal shutdown continues. This parameter will only take effect if all LU Type 6.2 parallel connections, if any, have successfully completed CNOS close processing.

- To attempt to force-close the CICS z/OS Communications Server ACB if there are any hung terminals, specify the TCSWAIT=mm (with an appropriate time interval) and TCSACTN=FORCE system initialization parameters.

TCSWAIT

The **TCSWAIT** system initialization parameter specifies the required CICS terminal control shutdown wait threshold.

TCSWAIT={4|number|NO|NONE|0}

The wait threshold is the time, during shutdown, that CICS terminal control allows to pass before it considers terminal shutdown to be hung. If all z/OS Communications Server sessions shutdown and close before the threshold expires then the CICS shutdown process moves on to its next stage, and the terminal control wait threshold then no longer applies. If, however, some of the z/OS Communications Server sessions do not complete shutdown and close, then CICS takes special action with these sessions. For details of this special action see the description of the TCSACTN system initialization parameter. The wait threshold only applies to z/OS Communications Server sessions; that is, z/OS Communications Server terminals and z/OS Communications Server intersystem connections. The wait time is specified as a number of minutes, in the range 1 through 99. As a special case, TCSWAIT=NO may be specified to indicate that terminal control shutdown is never to be considered hung, no matter how long the shutdown and close process takes. TCSWAIT=NONE and TCSWAIT=0 are alternative synonyms for TCSWAIT=NO, and all three have the same effect (internally they are held as the one value 0 (zero)).

The value that you specify on the TCSWAIT system initialization parameter should be large enough so that under normal circumstances all z/OS Communications Server terminals and connections shutdown in an orderly fashion. To help choose this value, consider using a value slightly larger than the elapsed time between the following two CICS terminal control shutdown messages:

```
DFHZC2305 Termination of VTAM sessions beginning
DFHZC2316 VTAM ACB is closed
```

Note: VTAM is now z/OS Communications Server.

TCT

The **TCT** system initialization parameter specifies which terminal control table, if any, is to be loaded.

TCT={NO|xx|YES}

For guidance about coding the macros for this table, see Terminal control table (TCT) .

If you reassemble the TCT after starting CICS, any changes are applied when you next start CICS, even if it is a warm or emergency startup.

If you have z/OS Communications Server-connected terminals only, you can specify TCT=NO. If you do this, a dummy TCT called DFHTCTDY, is loaded during system initialization. For more information about DFHTCTDY, see The dummy TCT, DFHTCTDY. If you code TCT=NO, you must specify a CSD group list in the **GRPLIST** parameter.

TCTUAKEY

The **TCTUAKEY** system initialization parameter specifies the storage key for the terminal control table user areas (TCTUAs) if you are operating CICS with storage protection (STGPROT=YES).

TCTUAKEY={USER|CICS}

The permitted values are USER (the default), or CICS:

USER CICS obtains the amount of storage for TCTUAs in user key. This allows a user program executing in any key to modify the TCTUA.

CICS CICS obtains the amount of storage in CICS key. This means that only programs executing in CICS key can modify the TCTUA, and user-key programs have read-only access.

If CICS is running without storage protection, the TCTUAKEY parameter only designates which DSA (User or CICS) the storage comes from. The TCTUAs are accessed in CICS-key whether they are in the UDSA or CDSA.

TCTUALOC

The **TCTUALOC** system initialization parameter specifies where terminal user areas (TCTUAs) are to be stored.

TCTUALOC={BELOW|ANY}

Valid values are as follows:

BELOW
The TCTUAs are stored in 24-bit storage (below the 16 MB line). If you require the terminal user areas to be in 24-bit storage, because you have application programs that are not capable of 31-bit addressing, specify this setting.

ANY The TCTUAs are stored anywhere in virtual storage. CICS stores TCTUAs in 31-bit storage (above the 16 MB line) if possible. This setting is the default.

For more information about TCTUAs, see The TCTUA.

For details about defining terminals using RDO, see Model TERMINAL definitions in group DFHTERM.

TD

The **TD** system initialization parameter specifies the number of VSAM buffers and strings to be used for intrapartition transient data (TD).

TD=({3|*decimal_value_1*}[,{ 3|*decimal_value_2*}])

Valid values are as follows:

decimal_value_1
The number of buffers to be allocated for the use of intrapartition transient data. The value must be in the range 1 through 32 767. The default value is 3.

CICS obtains, above the 16 MB line, storage for the TD buffers in units of the page size (4 KB). Because CICS optimizes the use of the storage obtained, TD may allocate more buffers than you specify, depending on the control interval (CI) size you have defined for the intrapartition data set.

For example, if the CI size is 1536, and you specify 3 buffers (the default number), CICS allocates 5 buffers. This is because 2 pages (8192 bytes) are required to obtain sufficient storage for three 1536-byte buffers, a total of only 4608 bytes, which would leave 3584 bytes of spare storage in the second page. In this case, CICS allocates another 2 buffers (3072 bytes) to minimize the amount of unused storage. In this way, CICS uses storage that would otherwise be unavailable for any other purpose.

decimal_value_2
The number of VSAM strings to be allocated for the use of intrapartition transient data. The value must be in the range 1 through 255, and must not exceed the value specified in *decimal_value_1*. The default value is 3.

For example, TD=(8,5) specifies 8 buffers and 5 strings.

The order in which you code the values is significant, so if you want to omit a value, you must code a comma in the place of that value. For example, TD=(,2) specifies the default for the number of buffers and explicitly specifies the number of strings.

TDINTRA

The **TDINTRA** system initialization parameter specifies whether CICS is to initialize with empty intrapartition TD queues.

TDINTRA={NOEMPTY|EMPTY}
Valid values are as follows:

NOEMPTY
CICS recovers all the intrapartition TD queues to the state they were in at the previous termination of CICS, as in a normal emergency restart. The TD queue resource definitions are recovered from the CICS global catalog.

EMPTY
CICS initializes with all the intrapartition TD queues empty. This option must be used when CICS is initializing in remote site recovery mode (OFFSITE=YES).

You can optionally use this option to preform a cold start of your intrapartition TD queues to initialize them as empty.

The option is significant only on warm and emergency restarts—cold starts always initialize with empty queues. Note that the EMPTY option may cause data integrity problems because all indoubt log records associated with logically recoverable TD queues are discarded.

The TD queue resource definitions are recovered from the CICS global catalog.

TRANISO

The **TRANISO** system initialization parameter specifies, together with the **STGPROT** system initialization parameter, whether you want transaction isolation in the CICS region.

TRANISO={NO|YES}
The permitted values are NO (the default), or YES.

NO This is the default. If you specify NO, or allow this parameter to default, CICS operates without transaction isolation, and all storage in the CICS address space is addressable. If you specify STGPROT=YES and TRANISO=NO, CICS storage protection is active without transaction isolation.

YES Transaction isolation is required. This ensures that the user-key task-lifetime storage of transactions defined with the ISOLATE(YES) option is isolated from the user-key programs of other transactions.

If you specify TRANISO=YES and STGPROT=YES, CICS operates with transaction isolation. YES is the default for the STGPROT system initialization parameter.

If you specify TRANISO=YES, but STGPROT=NO is specified, CICS issues an information message during initialization, and operates without transaction isolation. If STGPROT=NO and TRANISO=YES are specified in the system initialization table, an error occurs during assembly (MNOTE 8).

Notes:

1. VSAM nonshared resources (NSR) are not supported for transactions that use transaction isolation. You should specify ISOLATE(NO) when you define transactions that access VSAM files using NSR. You can also function ship the file request to a remote region. The DFHMIRS program that carries out the request is defined with an EXECKEY of CICS. A CICS-key program has read and write access to CICS-key and user-key storage of its own task and all other tasks, whether or not transaction isolation is active.
2. Storage protection, transaction isolation, and command protection protect storage from user application code. They add no benefit to a region where no user code is executed; that is, a pure terminal-owning region (TOR) or a pure file-owning region (FOR) (where no distributed program link (DPL) requests are function-shipped).
3. Transaction isolation does not apply to 64-bit storage.

TRAP

The **TRAP** system initialization parameter specifies whether the FE global trap exit is to be activated at system initialization.

TRAP={OFF|ON}

This exit is for diagnostic use under the guidance of service personnel.

TRDUMAX

The **TRDUMAX** system initialization parameter specifies the limit on the number of transaction dumps that may be taken per Dump Table entry.

TRDUMAX={999|number}

If this number is exceeded, subsequent transaction dumps for that particular entry will be suppressed.

number

A number in the range 0 through 999. The default, 999, enables an unlimited number of dumps to be taken.

TRTABSZ

The **TRTABSZ** system initialization parameter specifies the size, in kilobytes, of the internal trace table.

TRTABSZ={12288| *number-of-kilobytes*}

12288 The default size (12 MB) of the internal trace table.

number-of-kilobytes
The number of kilobytes of storage to be allocated for the internal trace table, in the range 16 KB through 1048576 KB (1 GB). The table is page aligned and occupies a whole number of pages. If the value specified is not a multiple of the page size (4 KB), it is rounded up to the next multiple of 4 KB.

The CICS internal trace table is allocated at an early stage during CICS initialization, and it exists for the whole of the CICS run.

CICS obtains MVS 64-bit (above-the-bar) storage (outside the CICS DSAs) for the internal trace table.

If you change the size of the internal trace table, check your current setting for the z/OS parameter **MEMLIMIT**. **MEMLIMIT** limits the amount of 64-bit storage that the CICS address space can use. Your setting for **TRTABSZ** must remain within **MEMLIMIT**, and you must also allow for other use of 64-bit storage in the CICS region.

For information about the **MEMLIMIT** value for CICS, and instructions to check the value of **MEMLIMIT** that currently applies to the CICS region, see Estimating, checking, and setting MEMLIMIT in Improving performance. For further information about **MEMLIMIT** in z/OS, see Limiting the use of memory objects in the z/OS MVS Programming: Extended Addressability Guide.

TRTRANSZ

The **TRTRANSZ** system initialization parameter specifies the size, in kilobytes, of the transaction dump trace table.

TRTRANSZ={1024 | *number-of-kilobytes*}

When a transaction dump is taken, CICS obtains MVS storage in 64-bit (above-the-bar) storage for the transaction dump trace table.

1024 1024 KB is the default size of the transaction dump trace table.

number-of-kilobytes
The number of kilobytes of storage to be allocated for the transaction dump trace table, in the range 16 -1048576 KB (1 GB).

Trace entries are of variable lengths. The average length of a trace entry is approximately 100 bytes. 1 KB is equal to 1024 bytes.

When you set this parameter, check your current setting for the z/OS parameter **MEMLIMIT**. **MEMLIMIT** limits the amount of 64-bit storage that the CICS address space can use. Your setting for **TRTRANSZ** must remain within **MEMLIMIT**, and you must also allow for other facilities in the CICS region that use 64-bit storage. See Estimating, checking, and setting MEMLIMIT in Improving performance. For information about **MEMLIMIT** in z/OS, see Limiting the use of memory objects in the z/OS MVS Programming: Extended Addressability Guide.

TRTRANTY

The **TRTRANTY** system initialization parameter specifies which trace entries should be copied from the internal trace table to the transaction dump trace table.

TRTRANTY={TRAN|ALL}

Valid values are as follows:

TRAN Only the trace entries associated with the transaction that is abending will be copied to the transaction dump trace table.

ALL All of the trace entries from the internal trace table will be copied to the transaction dump trace table. If the internal trace table size is larger than the transaction dump trace table size, the transaction dump trace table could wrap. This results in only the most recent trace entries being written to the transaction dump trace table.

TS

The **TS** system initialization parameter specifies whether you want to perform a cold start for temporary storage, as well as the number of VSAM buffers and strings to be used for auxiliary temporary storage.

TS=([COLD][,{0|3|decimal-value-1 }][,{3|decimal-value-2}])

Valid values are as follows:

COLD
The type of start for the temporary storage facility. COLD forces a cold start regardless of the value of the START parameter. If COLD is omitted, the TS start type is determined by the value of START.

0 No buffers are required; that is, only MAIN temporary storage is required.

decimal-value-1
The number of buffers to be allocated for the use of auxiliary temporary storage. The value must be in the range 3 through 32 767.

decimal-value-2
The number of VSAM strings to be allocated for the use of auxiliary temporary storage. The value must be in the range 1 through 255, and must not exceed the value specified in decimal-value-1. The default value is 3.

For example, TS=(,8,5) specifies 8 buffers and 5 strings.

The operands of the TS parameter are positional. You must code commas to indicate missing operands if others follow. For example, TS=(,8) specifies the number of buffers and allows the other operands to default.

TSMAINLIMIT

The **TSMAINLIMIT** system initialization parameter specifies a limit for the storage that is available for main temporary storage queues to use. You can specify an amount of storage in the range 1 - 32768 MB (32 GB), but this amount must not be greater than 25% of the value of the z/OS parameter **MEMLIMIT**. The default is 64 MB.

TSMAINLIMIT={64M|*nnnnn*M|*nn*G}

64M The default setting in megabytes.

***nnnnn*M**
An amount of storage in megabytes. The allowed range is 1 - 32768 MB.

***nn*G** An amount of storage in gigabytes. The allowed range is 1 - 32 GB.

For example, TSMAINLIMIT=2G makes 2 GB of storage available to main temporary storage queues.

When you set this parameter, check your current setting for the z/OS parameter **MEMLIMIT**. **MEMLIMIT** limits the amount of 64-bit storage that the CICS address space can use. Your setting for **TSMAINLIMIT** must not be greater than 25% of the **MEMLIMIT** value.

If you set the **TSMAINLIMIT** system initialization parameter to greater than 25% of the **MEMLIMIT** value, message DFHTS1608 is issued and CICS terminates. For information about the **MEMLIMIT** value for CICS, and instructions to check the value of **MEMLIMIT** that currently applies to the CICS region, see Estimating, checking, and setting MEMLIMIT in Improving performance. For further information about **MEMLIMIT** in z/OS, see Limiting the use of memory objects in the z/OS MVS Programming: Extended Addressability Guide.

TST

The **TST** system initialization parameter specifies the temporary storage table suffix.

TST={NO|YES|*xx*}

NO CICS uses only RDO support for temporary storage queues, and does not load a TST.

YES CICS uses an unsuffixed version of the table, named DFHTST.

xx CICS uses a table named DFHTST*xx*. See Defining CICS resource table and module keywords for information on defining the temporary storage table suffix.

Note: To use a TST in combination with TSMODEL resource definitions, you must assemble the TST load module with the MIGRATE option. If the TST is not assembled with the MIGRATE option, CICS loads the TST only and does not provide any RDO support for temporary storage queues, and any attempts to install TSMODEL resource definitions are rejected.

For information about coding the macros for this table, see Temporary storage table (TST).

UDSASZE

The **UDSASZE** system initialization parameter specifies the size of the UDSA.

UDSASZE={0K|number}
The default size is 0, indicating that the DSA size can change dynamically. A non-zero value indicates that the DSA size is fixed.

number
specify number as an amount of storage in the range 0 to 16777215 bytes in multiples of 262144 bytes (256KB). If the size specified is not a multiple of 256KB (or 1MB if transaction isolation is active), CICS rounds the value up to the next multiple.

You can specify number in bytes (for example, 4194304), or as a whole number of kilobytes (for example, 4096K), or a whole number of megabytes (for example, 4M).

Restriction: You can specify the UDSAZSE parameter in PARM, SYSIN, or CONSOLE only.

Important: Setting the size of individual dynamic storage areas (DSAs) is not usually necessary and is not recommended. If you specify DSA size values that in combination do not allow sufficient space for the remaining DSAs, CICS fails to initialize. The limit on the storage available for the DSAs in 24-bit storage (below the line) is specified by the DSALIM system initialization parameter. You must allow at least 256K for each DSA in 24-bit storage for which you have not set a size. See DSA size limits.

UOWNETQL

The **UOWNETQL** system initialization parameter specifies a qualifier for the NETUOWID for units of work initiated on the local CICS region.

UOWNETQL=user_defined_value

UOWNETQL is required only if the z/OS Communications Server VTAM=NO is coded. The specified value is used in the following circumstances:

- CICS is performing a cold start and VTAM=NO has been specified.
- CICS is performing a cold start and the z/OS Communications Server ACB has failed to open.
- CICS is being started with VTAM=NO and the z/OS Communications Server ACB has not been opened since the last cold start of CICS.
- CICS is being started, the z/OS Communications Server ACB has failed to open, and the z/OS Communications Server ACB has not been opened since the last cold start of CICS.

If any of the above conditions apply and UOWNETQL is not specified, a dummy default UOWNETQL of 9UNKNOWN is used. This dummy UOWNETQL is invalid because the first character is a number. UOWNETQL is given this invalid name to avoid a conflict with any real, valid netid.

If any of the above conditions apply, UOWNETQL, or its default value, is used as the IPIC NETWORKID of this CICS region. It is also used as the default NETWORKID on IPCONN definitions for IPIC connections to other CICS regions.

The value you code can be from 1 to 8 characters long, and must consist of uppercase letters (A through Z), or numbers in the range 0 through 9. The first character must be a letter.

USERTR

The **USERTR** system initialization parameter specifies whether the master user trace flag is to be set on or off.

USERTR={ON|OFF}

If the user trace flag is off, the user trace facility is disabled, and EXEC CICS ENTER TRACENUM commands receive an INVREQ condition if EXCEPTION is not specified. If the program does not handle this condition the transaction will abend AEIP.

For programming information about the user trace facility using EXEC CICS ENTER TRACENUM commands, see ENTER TRACENUM.

USRDELAY

The **USRDELAY** system initialization parameter specifies the maximum time, in the range 0 - 10080 minutes (up to seven days), that an eligible user ID and its associated attributes are to be retained in the CICS region after use. A user ID that is retained in the user table can be reused.

USRDELAY={30|*number*}

For a user ID to be retained in the CICS region and eligible for reuse in the **USRDELAY** period, one of the following statements must apply to the user ID:

- The user ID was received from remote systems.
- The user ID was specified on the SECURITYNAME attribute in the CONNECTION resource.
- The user ID was specified on the USERID attribute in the SESSIONS resource.
- The user ID was specified on the USERID attribute in the definition of an intrapartition transient data queue.
- The user ID was specified on the USERID option on a START command.
- The user ID was specified on the USERID attribute for a non-terminal task, such as the alias tasks that are attached for processing HTTP requests.

Within the **USRDELAY** period, a user ID in any one of these categories can be reused in one of the other categories, provided that the request for reuse has the same qualifiers. If a user ID is qualified by a different group ID, APPLID, or terminal ID, a retained entry is not reused, except when changing the terminal ID on LU6.2 when the retained entry is used.

If a user ID is unused for more than the **USRDELAY** limit, it is removed from the system, and the message DFHUS0200 is issued. You can suppress this message in an XMEOUT global user exit program.

If you specify **USRDELAY=0**, all eligible user IDs are deleted immediately after use and cannot be reused. With **USRDELAY=0** set, the message DFHUS0200 is not issued.

When you specify **USRDELAY=0**, CICS drives a full sign-on for each incoming request (with I/O to the external security manager) and a full sign-off at the end of each transaction. This setting provides the highest level of security, but in some scenarios performance might be a higher priority. For example, if the CICS region communicates with other CICS regions and the connections carry high volumes of transaction routing or function shipping activity, multiple instances of sign-on and sign-off might be required for a single task. Select a **USRDELAY** value that gives the optimum balance of performance and security for the type of work carried out in each CICS region.

When a value other than 0 is specified for **USRDELAY**, the ability to change the attributes of the user or revoke the user ID becomes more difficult because the user ID and its attributes are retained in the region until the **USRDELAY** value has expired. For example, if you have specified **USRDELAY=30** for a user ID, but that user ID continues to run transactions every 25 minutes, the **USRDELAY** value never expires and any changes made to the user ID never come into effect.

If you previously specified low values for the **USRDELAY** system initialization parameter in your CICS regions to ensure that CICS detected changes to RACF

profiles quickly, you might want to increase this value, because CICS is notified immediately if RACF profile changes occur. The primary impact of a high **USRDELAY** value is that the amount of storage used for RACF control blocks is increased.

USSCONFIG

The **USSCONFIG** system initialization parameter specifies the name and path of the root directory for CICS Transaction Server configuration files on z/OS UNIX.

USSCONFIG={/var/cicsts/dfhconfig | *directory*}
Specifies the directory in which z/OS UNIX configuration files are stored. The default value is `/var/cicsts/dfhconfig`.

USSHOME

The **USSHOME** system initialization parameter specifies the name and path of the root directory for CICS Transaction Server files on z/OS UNIX.

USSHOME={/usr/lpp/cicsts/cicsts54 | *directory* | NONE}
The value for the **USSHOME** system initialization parameter must match the directory that you specified for CICS Transaction Server files on z/OS UNIX when you installed CICS using the DFHISTAR installation job. The default value for the **USSHOME** system initialization parameter is /usr/lpp/cicsts/cicsts54, which matches the default values for the DFHISTAR installation job. The maximum length of the **USSHOME** system initialization parameter is 255 characters.

If you changed any of the **TINDEX**, **PATHPREFIX**, or **USSDIR** parameters in the DFHISTAR installation job, you must specify a value for the **USSHOME** system initialization parameter to match the name and path that you specified for the root directory using those DFHISTAR parameters.

If you specify **USSHOME=NONE** instead of specifying a directory name, CICS does not use any default root directory in the UNIX System Services file system. In this case, some CICS functions that request data from this directory might produce unpredictable results.

VTAM (z/OS Communications Server)

The **VTAM** system initialization parameter specifies whether the z/OS Communications Server access method is to be used.

VTAM={YES|NO}
The default is VTAM=YES.

VTPREFIX

The **VTPREFIX** system initialization parameter specifies the first character to be used for the terminal identifiers (termids) of autoinstalled virtual terminals.

VTPREFIX={\|*character*}
Virtual terminals are used by the External Presentation Interface (EPI) and terminal emulator functions of the CICS Client products.

Termids generated by CICS for autoinstalled Client terminals consist of a 1-character prefix and a 3-character suffix. The default prefix is '\'. The suffix can have the values 'AAA' through '999'. That is, each character in the suffix

can have the value 'A' through 'Z' or '0' through '9'. The first suffix generated by CICS has the value 'AAA'. This is followed by 'AAB', 'AAC', ... 'AAZ', 'AA0', 'AA1', and so on, up to '999'.

Each time a Client virtual terminal is autoinstalled, CICS generates a 3-character suffix that it has not recorded as being in use.

By specifying a prefix, you can ensure that the termids of Client terminals autoinstalled on this system are unique in your transaction routing network. This prevents the conflicts that could occur if two or more terminal-owning regions (TORs) ship definitions of Client virtual terminals to the same application-owning region (AOR).

If such a naming conflict does occur—that is, if a Client virtual terminal is shipped to an AOR on which a remote terminal of the same name is already installed—the autoinstall user program is invoked in the AOR. Your user program can resolve the conflict by allocating an alias terminal identifier to the shipped definition. For details of writing an autoinstall user program to install shipped definitions, see Writing a program to control autoinstall of shipped terminals . However, you can avoid potential naming conflicts by specifying a different prefix, reserved for virtual terminals, on each TOR on which Client virtual terminals are to be installed.

You must not use the characters + - * < > = { } or blank.

Note:

1. When specifying a prefix, ensure that termids generated by CICS for Client terminals do not conflict with those generated by your autoinstall user program for user terminals, or with the names of any other terminals or connections.
2. Client terminal definitions are not recovered after a restart. Immediately after a restart, no Client terminals are in use, so when CICS generates suffixes it begins again with 'AAA'. This means that CICS does **not** always generate the same termid for any given Client terminal. This in turn means that server applications should not assume that a particular CICS-generated termid always equates to a particular Client terminal.

   If your server programs do make this assumption, you can use your autoinstall user program to allocate alias termids, by which the virtual terminals will be known to CICS, in a consistent manner.
3. Clients can override CICS Transaction Server for z/OS-generated termids.

WEBDELAY

The **WEBDELAY** system initialization parameter specifies two Web delay periods.

WEBDELAY=(5|time_out,60|keep_time)

These periods are:

1. A timeout period. The maximum time, in minutes, in the range 1-60, that a transaction started through the Web 3270 bridge interface, is allowed to remain in terminal wait state before it is automatically purged by CICS.
2. The terminal keep time. The time, in minutes, in the range 1-6000, during which state data is kept for a CICS Web 3270 bridge transaction, before CICS performs clean-up.

WLMHEALTH

The **WLMHEALTH** system initialization parameter specifies the time interval and the health adjustment value to be used by CICS on z/OS Workload Manager Health API (IWM4HLTH) calls, which CICS makes to inform z/OS WLM about the health state of a CICS region.

WLMHEALTH={(20|*interval*[,25|*number*])|OFF}]

interval
Specifies the amount of time, in seconds, between calls that CICS makes to the z/OS Workload Manager Health API (IWM4HLTH). The value specified must be in the range 0 through 600. The default value is 20.

number
Specifies the health adjustment value that CICS provides to the z/OS Workload Manager Health API (IWM4HLTH) in each call at the specified interval. The value must be in the range 1 through 100. The default value is 25.

The health indicator is a number which shows, in percent, how well the server is performing. The specified health adjustment value adjusts the health indicator of the region each time CICS invokes the z/OS WLM Health API:

- If the *interval* value is 0, the initial z/OS WLM health indicator of the CICS region is set to 0 during CICS initialization, and the z/OS WLM health incrementing process must be initiated at a later time by issuing a **SET WLMHEALTH OPEN** command.
- If the *interval* value is greater than 0, the CICS server health indicator is incremented at the specified intervals. The first interval starts when the following message is issued:

```
DFHSI1517 applid Control is being given to CICS
```

For example, `WLMHEALTH=(5,10)` specifies that CICS will update the z/OS WLM health indicator in increments of 10 every 5 seconds until the health indicator reaches 100.

For a normal CICS shutdown, the z/OS WLM health indicator is set to 0 during the shutdown process, regardless of the current z/OS WLM health indicator.

OFF
Specifies that CICS does not use the z/OS Workload Manager Health API (IWM4HLTH) to inform z/OS WLM about the health state of a CICS address space.

To use the health settings provided by CICS, Sysplex Distributor needs to be configured to use a value of `SERVERWLM` for **VIPADISTRIBUTE DISTMETHOD**. In this case, if the health value for a CICS region is less than 100, z/OS WLM reduces the recommendation that is provided to Sysplex Distributor for that region. For port sharing, if the shared port is a sysplex-distributed port and `SERVERWLM` is the distribution method that is in use, `SHAREPORTWLM` should be specified on the PORT statement of each target to take advantage of the z/OS WLM server-specific recommendations. For more information, see *z/OS Communications Server: IP Configuration Reference and z/OS Communications Server: IP Configuration Guide*.

WRKAREA

The **WRKAREA** system initialization parameter specifies the number of bytes to be allocated to the common work area (CWA).

WRKAREA={512|number}
This area, for use by your installation, is initially set to binary zeros, and is available to all programs. It is not used by CICS. The maximum size for the work area is 3584 bytes.

XAPPC

The **XAPPC** system initialization parameter specifies whether RACF session security can be used when establishing APPC sessions.

XAPPC={NO|YES}
Valid values are as follows:

NO RACF session security cannot be used.

YES RACF session security can be used.

If you specify BINDSECURITY=YES for a specific APPC connection, a request to RACF is issued to extract the security profile. If the profile exists, it is used to bind the session.

Note: If you specify XAPPC=YES, the external security manager that you use must support the APPCLU general resource class, otherwise CICS fails to initialize.

Restriction: You can specify the XAPPC parameter in the SIT, PARM, or SYSIN only.

For information on how resource security can provide a further level of security to transaction security, see Resource security for transactions.

XCFGROUP

The **XCFGROUP** system initialization parameter specifies the name of the cross-system coupling facility (XCF) group to be joined by this region.

XCFGROUP={DFHIR000|*name*}
The group name must be eight characters long, padded at the end with blanks if necessary. The valid characters are A-Z 0-9 and the national characters $ # and @. To avoid using the names IBM uses for its XCF groups, do not begin group names with the letters A through C, E through I, or the character string "SYS". Also, do not use the name "UNDESIG", which is reserved for use by the system programmer in your installation.

It is recommended that you use a group name beginning with the letters "DFHIR".

You can specify **XCFGROUP** on the SIT macro or as a SYSIN override. You cannot specify it as a console override.

Each CICS region can join only one XCF group, which happens when it signs on to CICS interregion communication (IRC). The default XCF group is DFHIR000.

XCF groups allow CICS regions in different MVS images within the same sysplex to communicate with each other across multi-region operation (MRO) connections.

Note: Regions in the same MVS image too, can communicate with each other using MRO, but this does not require a coupling facility. The only situation in

which CICS regions in the same MVS image cannot communicate via MRO is when they are members of different XCF groups.

For introductory information about XCF/MRO, and instructions on how to set up XCF groups, see Cross-system multiregion operation (XCF/MRO).

For information on how resource security can provide a further level of security to transaction security, see Resource security for transactions.

XCMD

The **XCMD** system initialization parameter specifies whether you want CICS to perform command security checking, and optionally the RACF resource class name in which you have defined the command security profiles.

XCMD={YES|name|NO}

If you specify YES, or a RACF resource class name, CICS calls RACF to verify that the userid associated with a transaction is authorized to use a CICS command for the specified resource. Such checking is performed every time a transaction tries to use a COLLECT, DISABLE, DISCARD, ENABLE, EXTRACT, INQUIRE, PERFORM, RESYNC, or SET command, or any of the FEPI commands, for a resource.

Note: The checking is performed only if you have specified YES for the SEC system initialization parameter and specified the CMDSEC(YES) option on the transaction resource definition.

YES CICS calls RACF, using the default class name of CICSCMD prefixed by C or V, to check whether the userid associated with a transaction is authorized to use a CICS command for the specified resource. The resource class name is CCICSCMD and the grouping class name is VCICSCMD.

name CICS calls RACF, using the specified resource class name prefixed by C or V, to verify that the userid associated with a transaction is authorized to use a CICS command for the specified resource. The resource class name is C*name* and the grouping class name is V*name*.

The resource class name specified must be 1 through 7 characters.

NO CICS does not perform any command security checks, allowing any user to use commands that would be subject to those checks.

Restrictions: You can specify the **XCMD** parameter in the SIT, PARM, or SYSIN only.

For information on how resource security can provide a further level of security to transaction security, see Resource security for transactions.

XDB2

The **XDB2** system initialization parameter specifies whether you want CICS to perform DB2ENTRY security checking.

XDB2={NO|name}

Valid values are as follows:

NO CICS does not perform any DB2 resource security checks.

name CICS calls RACF, using the specified general resource class name, to

check whether the userid associated with the CICS DB2 transaction is authorized to access the DB2ENTRY referenced by the transaction.

Unlike the other X*aaa* system initialization parameters, this DB2 security parameter does not provide a YES option that implies a default CICS resource class name for DB2ENTRY resources. You have to specify your own DB2 resource class name.

For information on how resource security can provide a further level of security to transaction security, see Resource security for transactions.

XDCT

The **XDCT** system initialization parameter specifies whether you want CICS to perform resource security checking for transient data queues.

XDCT={YES|name|NO}

If you specify YES or a RACF resource class name, CICS calls RACF to verify that the userid associated with a transaction is authorized to access the transient data queue. Such checking is performed every time a transaction tries to access a transient data queue.

Note: The checking is performed only if you have specified YES for the SEC system initialization parameter and specified the RESSEC(YES) option on the transaction resource definition. For further information on how resource security can provide a further level of security to transaction security, see Resource security for transactions.

YES CICS calls RACF, with the default CICS resource class name of CICSDCT prefixed by D or E, to verify whether the userid associated with the transaction is authorized to access the specified transient data queue.

The resource class name is DCICSDCT and the grouping class name is ECICSDCT.

name CICS calls RACF, using the specified resource class name, to check whether the userid associated with the transaction is authorized to access the specified transient data queue. The resource class name is D*name* and the grouping class name is E*name*.

The resource class name specified must be 1 through 7 characters.

NO CICS does not perform any transient data security checks, allowing any user to access any transient data queue.

Restrictions: You can specify the XDCT parameter in the SIT, PARM, or SYSIN only.

XFCT

The **XFCT** system initialization parameter specifies whether you want CICS to perform file resource security checking, and optionally specifies the RACF resource class name in which you have defined the file resource security profiles.

XFCT={YES|name|NO}

If you specify YES, or a RACF resource class name, CICS calls RACF to verify that the userid associated with a transaction is authorized to access File Control-managed files. Such checking is performed every time a transaction tries to access a file managed by CICS file control. The checking is performed

only if you have specified YES for the SEC system initialization parameter and specified the RESSEC(YES) option on the resource definitions. For further information on how resource security can provide a further level of security to transaction security, see Resource security for transactions.

Note: You can specify the **XFCT** parameter in the SIT, PARM, or SYSIN only.

YES CICS calls RACF, using the default CICS resource class name of CICSFCT prefixed by F or H, to verify that the userid associated with a transaction is authorized to access files reference by the transaction. The resource class name is FCICSFCT and the grouping class name is HCICSFCT.

name CICS calls RACF, using the specified resource class name, to verify that the userid associated with a transaction is authorized to access files referenced by the transaction. The resource class name is F*name* and the grouping class name is H*name*.

The resource class name specified must be 1 through 7 characters.

NO CICS does not perform any file resource security checks, allowing any user to access any file.

XHFS

The **XHFS** system initialization parameter specifies whether CICS is to check the transaction user's ability to access files in the z/OS UNIX System Services file system.

XHFS={YES|NO}

At present, this checking applies only to the user ID of the Web client when CICS Web support is returning z/OS UNIX file data as the static content identified by a URIMAP definition. The checking is performed only if you have specified YES for the SEC system initialization parameter. However, the RESSEC option on the transaction resource definition does not affect this security checking. For further information on how resource security can provide a further level of security to transaction security, see Resource security for transactions.

Note: You can specify the **XHFS** parameter in the SIT, PARM, or SYSIN only.

YES CICS is to check whether the user identified as the Web client is authorized to access the file identified by the URIMAP that matches the incoming URL. This check is in addition to the check performed by z/OS UNIX System Services against the CICS region user ID. If access to the file is denied for either of these user IDs, the HTTP request is rejected with a 403 (Forbidden) response.

NO CICS is not to check the client user's access to z/OS UNIX files. Note that the CICS region user ID's access to these files is still checked by z/OS UNIX System Services.

XJCT

The **XJCT** system initialization parameter specifies whether you want CICS to perform journal resource security checking.

XJCT={YES|name|NO}

If you specify YES, or a RACF resource class name, CICS calls RACF to verify that the userid associated with a transaction is authorized to access the

referenced journal. Such checking is performed every time a transaction tries to access a CICS journal. The checking is performed only if you have specified YES for the **SEC** system initialization parameter and specified the RESSEC is active for the resource definitions. For further information on how resource security can provide a further level of security to transaction security, see Resource security for transactions.

Note: You can specify the **XJCT** parameter in the SIT, PARM, or SYSIN only.

YES CICS calls RACF using the default CICS resource class name of CICSJCT prefixed by a J or K, to check whether the userid associated with a transaction is authorized to access CICS journals referenced by the transaction. The resource class name is JCICSJCT and the grouping class name is KCICSJCT.

name CICS calls RACF, using the specified resource class name prefixed by J or K, to verify that the userid associated with a transaction is authorized to access CICS journals. The resource class name is J*name* and the grouping class name is K*name*.

The resource class name specified must be 1 through 7 characters.

NO CICS does not perform any journal resource security checks, allowing any user to access any CICS journal.

XLT

The **XLT** system initialization parameter specifies a suffix for the transaction list table.

XLT={NO|xx|YES}

The table contains a list of transactions that can be attached during the first quiesce stage of system termination. See Defining CICS resource table and module keywords.

YES The default transaction list table, DFHXLT, is used.

xx The transaction list table DFHXLTxx is used.

NO A transaction list table is not used.

For guidance information about coding the macros for this table, see Transaction list table (XLT).

For information on how resource security can provide a further level of security to transaction security, see Resource security for transactions.

XPCT

The **XPCT** system initialization parameter specifies whether you want CICS to perform started transaction resource security checking, and optionally specifies the name of the RACF resource class name in which you have defined the started task security profiles.

XPCT={YES|name|NO}

If you specify YES, or a RACF resource class name, CICS calls RACF to verify that the userid associated with a transaction is authorized to use started transactions and related EXEC CICS commands. Such checking is performed every time a transaction tries to use a started transaction or one of the EXEC CICS commands: COLLECT STATISTICS TRANSACTION, DISCARD TRANSACTION, INQUIRE TRANSACTION, or SET TRANSACTION. The

checking is performed only if you have specified YES for the **SEC** system initialization parameter and specified the RESSEC(YES) option on the resource definitions. For further information on how resource security can provide a further level of security to transaction security, see Resource security for transactions.

Note: You can specify the **XPCT** parameter in the SIT, PARM, or SYSIN only.

YES CICS calls RACF using the default CICS resource class name CICSPCT prefixed with A or B, to verify that the userid associated with a transaction is authorized to use started transactions or related EXEC CICS commands.

The resource class name is ACICSPCT and the grouping class name is BCICSPCT.

name CICS calls RACF, using the specified resource class name, to verify that the userid associated with a transaction is authorized to use started transactions or related EXEC CICS commands. The resource class name is A*name* and the grouping class name is B*name*.

The resource class name specified must be 1 through 7 characters.

NO CICS does not perform any started task resource security checks, allowing any user to use started transactions or related EXEC CICS commands.

XPPT

The **XPPT** system initialization parameter specifies that CICS is to perform application program resource security checks and optionally specifies the RACF resource class name in which you have defined the program resource security profiles.

XPPT={YES|name|NO}

You can specify the **XPPT** parameter in the SIT, PARM, or SYSIN only. Checking is performed every time a transaction tries to invoke another program by using one of the CICS commands: LINK, LOAD, or XCTL.

Note: The checking is performed only if you have specified YES for the **SEC** system initialization parameter and specified the RESSEC(YES) option on the resource definitions. For further information on how resource security can provide a further level of security to transaction security, see Resource security for transactions.

YES CICS calls RACF, using the default resource class name prefixed by M or N, to verify that the userid associated with a transaction is authorized to use LINK, LOAD, or XCTL commands to invoke other programs. The resource class name is MCICSPPT and the grouping class name is NCICSPPT.

name CICS calls RACF, with the specified resource class name prefixed by M or N, to verify that the userid associated with a transaction is authorized to use LINK, LOAD, or XCTL commands to invoke other programs. The resource class name is M*name* and the grouping class name is N*name*.

The resource class name specified must be 1 through 7 characters.

NO CICS does not perform any application program authority checks, allowing any user to use LINK, LOAD, or XCTL commands to invoke other programs.

XPSB

The **XPSB** system initialization parameter specifies whether you want CICS to perform program specification block (PSB) security checking and optionally specifies the RACF resource class name in which you have defined the PSB security profiles.

XPSB={YES|name|NO}

You can specify the **XPSB** parameter in the SIT, PARM, or SYSIN only. If you specify YES, or a RACF resource class name, CICS calls RACF to check that the userid associated with a transaction is authorized to access PSBs (which describe databases and logical message destinations used by application programs). Such checking is performed every time a transaction tries to access a PSB.

Note:

1. The checking is performed only if you have specified YES for the **SEC** system initialization parameter and specified the RESSEC(YES) option on the resource definitions. For further information on how resource security can provide a further level of security to transaction security, see Resource security for transactions.
2. If you require security checking for PSBs to apply to remote users who access this region by means of transaction routing, the system initialization parameter **PSBCHK**=YES must be specified. For more information about this parameter, see "PSBCHK" on page 83.

YES CICS calls RACF, using the default resource class name CICSPSB prefixed by P or Q, to verify that the userid associated with a transaction is authorized to access PSBs. The resource class name is PCICSPSB and the grouping class name is QCICSPSB.

name CICS calls RACF, using the specified resource class name prefixed by P or Q, to verify that the userid associated with a transaction is authorized to access PSBs. The resource class name is P*name* and the grouping class name is Q*name*.

The resource class name specified must be 1 through 7 characters.

NO CICS does not perform any PSB resource security checks, allowing any user to access any PSB.

XPTKT

The **XPTKT** system initialization parameter specifies whether CICS checks if a user can generate a PassTicket for the user's userid using the **EXEC CICS REQUEST PASSTICKET** command or the **EXEC FEPI REQUEST PASSTICKET** command.

For more information about generating PassTickets, see Generating and using PassTickets for secure sign-on.

XPTKT={YES|NO}

Valid values for this parameter are as follows:

YES
A check is made that the `userid` has update authority for the profile `IRRPTAUTH.applid.userid` in the class PTKTDATA.

NO No check is done.

For information on how resource security can provide a further level of security to transaction security, see Resource security for transactions.

XRES

The **XRES** system initialization parameter specifies whether you want CICS to perform resource security checking for particular CICS resources and optionally specifies the general resource class name in which you have defined the resource security profiles.

XRES={YES|*name*|NO}

You can specify the **XRES** parameter in the SIT, PARM, or SYSIN only. If you specify YES, or a general resource class name, CICS calls the external security manager to verify that the user ID associated with a transaction is authorized to use the resource. This checking is performed every time a transaction tries to access a resource.

The actual profile name passed to the external security manager is the name of the resource to be checked, prefixed by its resource type; for example, for a document template whose resource definition is named "WELCOME", the profile name passed to the external security manager is DOCTEMPLATE.WELCOME. Even if a command references the document template using its 48-character template name, the shorter name (up to 8 characters) of the DOCTEMPLATE resource definition is always used for security checking.

The checking is performed only if you have specified YES for the **SEC** system initialization parameter and specified the RESSEC(YES) option on the TRANSACTION resource definition. For further information on how resource security can provide a further level of security to transaction security, see Resource security for transactions.

YES CICS calls the external security manager, using the default CICS resource class name of RCICSRES, to check whether the user ID associated with a transaction is authorized to use the resource it is trying to access. The resource class name is RCICSRES and the grouping class name is WCICSRES.

name CICS calls the external security manager, using the specified resource class name prefixed by the letter R, to check whether the user ID associated with a transaction is authorized to use the resource it is trying to access. The resource class name is R*name* and the grouping class name is W*name*. The resource class name specified must be 1 through 7 characters.

NO CICS does not perform any security checks for resources, allowing access to any user.

For a list of commands subject to XRES resource class checks, together with their respective profiles, see Resource and command check cross reference.

XRF

The **XRF** system initialization parameter specifies whether XRF support is to be included in the CICS region.

XRF={NO|YES} (active and alternate)
If the CICS region is started with the START=STANDBY system initialization parameter specified, the CICS region is the alternate CICS region. If the CICS region is started with the START=AUTO, START=INITIAL or START=COLD system initialization parameter specified, the CICS region is the active CICS region. The active CICS region signs on as such to the CICS availability manager.

If you specify XRF=YES, do not specify a value for the **GRNAME** system initialization parameter. Any value specified is set to blanks.

For information on how resource security can provide a further level of security to transaction security, see Resource security for transactions.

XTRAN

The **XTRAN** system initialization parameter specifies whether you want CICS to perform transaction-attach security checking and optionally specifies the RACF resource class name in which you have defined the transaction security profiles.

XTRAN={YES|*name*|NO}
You can specify the **XTRAN** parameter in the SIT, PARM, or SYSIN only. If you specify YES, or a RACF resource class name, CICS calls RACF to verify that the userid associated with the transaction is permitted to run the transaction.

Note: The checking is performed only if you have specified YES for the **SEC** system initialization parameter.

YES CICS calls RACF, using the default CICS resource class name of CICSTRN prefixed by T or G, to verify that the userid associated with the transaction is authorized to run the transaction. The resource class name is TCICSTRN and the grouping class name is GCICSTRN.

name CICS calls RACF, using the specified resource class name prefixed by T or G, to verify that the userid associated with the transaction is authorized to run the transaction. The resource class name is T*name* and the corresponding grouping class name is G*name*.

The name specified must be 1 through 7 characters.

NO CICS does not perform any transaction-attach security checks, allowing any user to run any transaction.

For information on how resource security can provide a further level of security to transaction security, see Resource security for transactions.

XTST

The **XTST** system initialization parameter specifies whether you want CICS to perform security checking for temporary storage queues and optionally specifies the RACF resource class name in which you have defined the temporary storage security profiles.

XTST={YES|name|NO}
You can specify the XTST parameter in the SIT, PARM, or SYSIN only. If you

specify YES, or a RACF resource class name, CICS calls RACF to verify that the userid associated with a temporary storage request is authorized to access the referenced temporary storage queue.

Security checking for temporary storage queues is performed only if you have specified all of the following options in addition to the XTST parameter:

- YES for the **SEC** system initialization parameter
- RESSEC(YES) in the relevant TRANSACTION resource definitions
- SECURITY(YES) in your TSMODEL resource definitions
- If you use a temporary storage table (TST), the DFHTST TYPE=SECURITY macro

For further information on how resource security can provide a further level of security to transaction security, see Resource security for transactions.

YES CICS calls RACF, using the default CICS resource class name of CICSTST prefixed by S or U, to verify that the userid associated with the transaction is authorized to access temporary storage queues referenced by the transaction. The resource class name is SCICSTST and the corresponding grouping class name is UCICSTST.

name CICS calls RACF, using the specified resource class name prefixed by S or U, to verify that the userid associated with a transaction is authorized to access temporary storage queues. The resource name class name is S*name* and the grouping class name is U*name*.

The name specified must be 1 through 7 characters.

NO CICS does not perform any temporary storage security checks, allowing any user to access any temporary storage queue.

XUSER

The **XUSER** system initialization parameter specifies whether CICS is to perform surrogate user checks.

XUSER={YES|NO}

You can specify the **XUSER** parameter in the SIT, PARM, or SYSIN only. Valid values are as follows:

YES CICS is to perform surrogate user checking in all those situations that permit such checks to be made; for example, on EXEC CICS START commands without an associated terminal. Surrogate user security checking is also performed by CICS against user IDs installing or modifying DB2 resource definitions that specify AUTHID or COMAUTHID.

Note: The **XUSER** parameter is also used by CICS to control access to the AUTHTYPE and COMAUTHTYPE attributes on DB2 resource definitions, although not through surrogate user checks. For more information about AUTHTYPE and COMAUTHTYPE attributes, see DB2CONN resources.

For information about the various circumstances in which CICS performs surrogate user checks, see Surrogate user security.

NO CICS is not to perform any surrogate user checking.

For information on how resource security can provide a further level of security to transaction security, see Resource security for transactions.

Notices

This information was developed for products and services offered in the U.S.A. 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 (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.

© Copyright IBM Corp. 1989, 2017

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
US

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Programming License Agreement, or any equivalent agreement between us.

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

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 4 are included in the following sections of the online product documentation:

- Developing applications
- Developing system programs
- RACF security overview
- Developing for external interfaces
- Reference: application developmenth
- Reference: system programming
- Reference: connectivity

Information that is NOT intended to be used as a Programming Interface of CICS Transaction Server for z/OS, Version 5 Release 4 , but that might be misconstrued as Programming Interfaces, is included in the following sections of the online product documentation:

- Troubleshooting and support
- Reference: diagnostics

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 Transaction Server for z/OS, Version 5 Release 4 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
- Supplied Transactions
- 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 Transaction Server for z/OS, Version 5 Release 4 , 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.

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.

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.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

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 is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, 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 personally identifiable information 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 personally identifiable information 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 collect no personally identifiable information. 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 personally identifiable information 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

(z/OS Communications Server)
  high performance option (HPO) 55

A

active delay interval for XRF 77
activity keypoint frequency (AKPFREQ) 15
ADI, system initialization parameter 12
AIBRIDGE, system initialization parameter 12
AICONS, system initialization parameter 12
AIEXIT, system initialization parameter 13
AILDELAY, system initialization parameter 13
AIQMAX, system initialization parameter 14
AIRDELAY, system initialization parameter 14
AKPFREQ, system initialization parameter 15
alternate delay interval for XRF 12
APPL statement, VBUILD application identifier 15
APPLID, system initialization parameter 15
AUTCONN, system initialization parameter 17
AUTHTYPE DB2 parameter 136
AUTODST, system initialization parameter 17
automatic dynamic storage tuning
  AUTODST 17
automatic start 102
AUTORESETTIME, system initialization parameter 17
auxiliary storage trace 18
AUXTR, system initialization parameter 18
AUXTRSW, system initialization parameter 18

B

backout exit programs 113
backout of resources at emergency restart 113
batching requests 69
BMS (basic mapping support)
  BMS system initialization parameter 18
  page-chaining command character string 78
  page-copying command character string 78
  page-purging command character string 79
BMS (basic mapping support) *(continued)*
  page-retrieval command character string 79
  PGCHAIN, BMS CHAIN command 78
  PGCOPY, BMS COPY command 78
  PGPURGE, BMS PURGE command 79
  PGRET, BMS RETRIEVAL command 79
  PRGDLAY, BMS PURGE DELAY command 81
  purge delay time interval 81
  versions of BMS 19
BMS, system initialization parameter 18
BRMAXKEEPTIME, system initialization parameter 20
buffers and strings, VSAM 116, 120

C

CDSASZE, system initialization parameter 20
CHKSTRM, system initialization parameter 21
CHKSTSK, system initialization parameter 21
CICS Web interface
  WEBDELAY system initialization parameter 125
CICSSVC, system initialization parameter 21
CILOCK, system initialization parameter 22
class, monitoring 65
CLINTCP, system initialization parameter 22
close destination request limit 76
CLSDST destination request limit 76
CLSDSTP, system initialization parameter 22
CLT (command list table) 23
CLT, system initialization parameter 23
CMDPROT, system initialization parameter 23
CMDSEC, system initialization parameter 24
COLD option
  system initialization parameter BMS 19
  system initialization parameter ICP 56
  system initialization parameter START 103
  system initialization parameter TS 120
COMAUTHTYPE DB2 parameter 136
command list table (CLT) 23
common work area (CWA) 127
common work area storage key
  system initialization parameter 34
CONFDATA, system initialization parameter 24
CONFTXT, system initialization parameter 26
CPSMCONN, system initialization parameter 26
cross-system coupling facility (XCF)
  coding the XCFGROUP system initialization parameter 127
CSD (CICS system definition file)
  CSDSTRNO 34
CSDACC, system initialization parameter 28
CSDBKUP, system initialization parameter 29
CSDBUFND, system initialization parameter 29
CSDBUFNI, system initialization parameter 29
CSDDISP, system initialization parameter 30
CSDDSN, system initialization parameter 30
CSDFRLOG, system initialization parameter 30
CSDINTEG, system initialization parameter 31
CSDJID, system initialization parameter 32
CSDLSRNO, system initialization parameter 32
CSDRECOV, system initialization parameter 32
CSDSTRNO, system initialization parameter 34
CWA (common work area) 127
CWAKEY, system initialization parameter 34

D

DAE, system initialization parameter 35
data interchange program (DIP) 37
data sets
  dump 41
database recovery control (DBRC)
  system initialization parameter, DLDBRC 16
  use of generic applid 16
date format 35
DATFORM, system initialization parameter 35
DB2 resource security
  XUSER system initialization parameter
    AUTHTYPE 136
    COMAUTHTYPE 136
DB2CONN, system initialization parameter 35
DBCTLCON, system initialization parameter 36

© Copyright IBM Corp. 1989, 2017

DBRC (database recovery control)
system initialization parameter, DLDBRC 16
use of generic applid 16
DDS option of system initialization parameter BMS 19
DEBUGTOOL, system initialization parameter 36
delay interval, primary, for XRF 77
delay intervals
active delay for XRF 77
alternate delay for XRF 12
JES for XRF 59
reconnection for XRF 17
delay, persistent verification 85
destination request limit, open and close 76
DFHCSVC, the CICS type 3 SVC 21
DFHDYP, dynamic transaction routing program
coding the DTRPGM system initialization parameter 40
DFHMQCOD 68
DFLTUSER, system initialization parameter 37
DIP (data interchange program) 37
DIP, system initialization parameter 37
DISMACP, system initialization parameter 37
DL/I
PDIR, system initialization parameter 77
specifying security checking of PSB entries 133
DOCCODEPAGE, system initialization parameter 37
DSA (dynamic storage areas)
RENTPGM, system initialization parameter 87
STGPROT, system initialization parameter 106
DSALIM, system initialization parameter 38
DSHIPIDL, system initialization parameter 38
DSHIPINT, system initialization parameter 39
DSRTPGM, system initialization parameter 39
DTRPGM, system initialization parameter 40
DTRTRAN, system initialization parameter 40
dump analysis and elimination
system initialization parameter 35
dump data sets 41
DUMP, system initialization parameter 41
DUMPDS, system initialization parameter 41
DUMPSW, system initialization parameter 41
DURETRY, system initialization parameter 42
dynamic transaction routing program, DFHDYP
coding the DTRPGM system initialization parameter 40

E

ECDSASZE, system initialization parameter 42
EDSALIM, system initialization parameter 43
emergency restart
resource backout 113
START system initialization parameter 102
ENCRYPTION, system initialization parameter 44
EODI, system initialization parameter 45
ERDSASZE, system initialization parameter 45
ESDSASZE, system initialization parameter 45
ESMEXITS, system initialization parameter 46
EUDSASZE, system initialization parameter 46
exception class monitoring 66
exit interval, region 56
exits
FE, global trap exit 118
external security interface 94

F

facilities
auxiliary trace autoswitch facility 18
FCQRONLY, system initialization parameter 47
FCT (file control table)
specifying the FCT suffix 47
FCT, system initialization parameter 47
FE global trap exit 118
FEPI (front end programming interface)
FEPI, system initialization parameter 48
field name start character 48
field separator characters 48
file control table (FCT)
specifying the FCT suffix 47
FLDSEP, system initialization parameter 48
FLDSTRT, system initialization parameter 48
FORCEQR, system initialization parameter 49
frequency, activity keypoint 15
front end programming interface (FEPI)
FEPI, system initialization parameter 48
FSSTAFF, system initialization parameter 49
FULL option of system initialization parameter BMS 19

G

global trap exit, FE 118
globalization 71
GMTEXT, system initialization parameter 51
GMTRAN, system initialization parameter 51
GNTRAN, system initialization parameter 52
good morning message 51
good morning transaction 51
GOOD MORNING transaction 90
group list, RDO 54
GRPLIST, system initialization parameter 54
GTFTR, system initialization parameter 55

H

high-performance option (HPO) 55
HPO (high-performance option) 55
HPO, system initialization parameter 55
HTTPSERVERHDR, system initialization parameter 56
HTTPUSRAGENTHDR, system initialization parameter 56

I

ICP (interval control program) 56
ICP, system initialization parameter 56
ICV, system initialization parameter 56
ICVR, system initialization parameter 57
ICVTSD, system initialization parameter 58
identity class monitoring 67
INFOCENTER, system initialization parameter 58
INITIAL
system initialization parameter
START 103
initial start
START system initialization parameter 102
INITPARM, system initialization parameter 58
internal trace, main storage 59
interregion communication (IRC) 59
intersystem communication (ISC) 59
interval control program (ICP) 56
intervals, activity keypoint 15
INTTR, system initialization parameter 59
IRC (interregion communication) 59
IRCSTRT, system initialization parameter 59
ISC (intersystem communication) 59
ISC, system initialization parameter 59

J

JES delay interval for XRF 59
JESDI, system initialization parameter 59

journaling
specifying security checking for journal entries 130
XJCT, system initialization parameter 130
JVMPROFILEDIR, system initialization parameter 60

K

keypoint frequency 15
KEYRING, system initialization parameter 60
keys for page-retrieval 95

L

LGDFINT system initialization parameter 60
LGNMSG, system initialization parameter 61
LLACOPY macro 61
LLACOPY, system initialization parameter 61
LOCALCCSID, system initialization parameter 62
locating modules in the relocatable program library 61
log defer time interval 60
logon data, z/OS Communications Server 61
LPA (link pack area)
LPA system initialization parameter 62
PRVMOD system initialization parameter 83
LPA, system initialization parameter 62

M

MAXOPENTCBS, system initialization parameter 63
MAXSOCKETS, system initialization parameter 63, 64
MAXSSLTCBS, system initialization parameter 64
MCT (monitoring control table) 64
MCT, system initialization parameter 64
message case 70
message level 70
message, good morning 51
MINIMUM option of system initialization parameter BMS 18
MINTLSLEVEL, system initialization parameter 65
MN, system initialization parameter 65
MNCONV, system initialization parameter 66
MNEXC, system initialization parameter 66
MNFREQ, system initialization parameter 67
MNIDN, system initialization parameter 67
MNPER, system initialization parameter 67
MNRES, system initialization parameter 67
MNSYNC, system initialization parameter 68
MNTIME, system initialization parameter 68
monitoring 65
exception class 66
identity class monitoring 67
performance class 67
transaction resource monitoring 67
monitoring control table (MCT) 64
MQCONN, system initialization parameter 68
MRO (multiregion operation)
batching 69
batching requests 69
extend lifetime of long-running mirror 69
long-running mirror 70
MROBTCH, system initialization parameter 69
MROFSE, system initialization parameter 69
MROLRM, system initialization parameter 70
MSGCASE, system initialization parameter 70
MSGLVL, system initialization parameter 70
multiregion operation (MRO)
batching 69
batching requests 69
extend lifetime of long-running mirror 69
long-running mirror 70
MXT, system initialization parameter 71

N

NATLANG, system initialization parameter 71
NCPLDFT, system initialization parameter 72
NEWSIT, system initialization parameter 72
NISTSP800131A, system initialization parameter 73
NODDS option of system initialization parameter BMS 19
NONRLSRECOV, system initialization parameter 74
NQRNL, system initialization parameter 75

O

open destination request limit 76
OPERTIM, system initialization parameter 76
OPNDLIM, system initialization parameter 76
OPNDST write-to-operator timeout limit 76

P

PA keys for page-retrieval 95
PA keys for screen copying 81
page-chaining command character string 78
page-copying command character string 78
page-purging command character string 79
page-retrieval command character string 79
page-retrieval keys 95
PARMERR, system initialization parameter 76
PDI, system initialization parameter 77
PDIR, system initialization parameter 77
performance class monitoring 67
persistent sessions 83, 84
persistent verification delay 85
PF keys for page-retrieval 95
PGCHAIN, system initialization parameter 78
PGCOPY, system initialization parameter 78
PGPURGE, system initialization parameter 79
PGRET, system initialization parameter 79
PLT (program list table)
system initialization programs 79
system termination programs 80
PLTPI, system initialization parameter 79
PLTPIUSR, system initialization parameter 80
PLTSD, system initialization parameter 80
PRGDLAY, system initialization parameter 81
primary delay interval for XRF 77
PRINT, system initialization parameter 81
program list table (PLT)
system initialization programs 79
system termination programs 80
program specification block (PSB)
PDIR, system initialization parameter 77
specifying security checking of PSB entries 133
PRTYAGE, system initialization parameter 82
PRVMOD, system initialization parameter 83
PSB (program specification block)
PDIR, system initialization parameter 77
specifying security checking of PSB entries 133
PSBCHK, system initialization parameter 83
PSDINT, system initialization parameter 83
PSTYPE, system initialization parameter 84
purge delay time interval, BMS 81

PVDELAY, system initialization
parameter 85

R

RACF (resource access control facility)
checking program entries with RACF 132
DFLTUSER, system initialization parameter 37
establishing APPC sessions 94
MRO bind-time security 94
resource level checking 94
SEC, system initialization parameter 94
SECPRFX, system initialization parameter 95
specifying a prefix to resource name 95
XAPPC, system initialization parameter 127
RACFSYNC, system initialization parameter 85
RAMAX, system initialization parameter 86
RAPOOL, system initialization parameter 86
RDO (resource definition online) group list (GRPLIST) 54
RDSASZE, system initialization parameter 87
read-only storage
system initialization parameter 87
RECEIVE ANY (RA) maximum 86
RECEIVE ANY (RA) pool size 86
reconnection delay interval (XRF) 17
reconnection transaction for XRF 90
recovery for files 74
region exit interval (ICV) 56
RENTPGM, system initialization parameter 87
request parameter list (RPL) 86
resource access control facility (RACF)
checking program entries with RACF 132
DFLTUSER, system initialization parameter 37
establishing APPC sessions 94
MRO bind-time security 94
resource level checking 94
SEC, system initialization parameter 94
SECPRFX, system initialization parameter 95
specifying a prefix to resource name 95
XAPPC, system initialization parameter 127
resource backout at emergency restart 113
resource definition online (RDO) group list (GRPLIST) 54
RESP, system initialization parameter 88
RESSEC, system initialization parameter 88
RMTRAN, system initialization parameter 90
RNL processing 75
RPL (request parameter list) 86
RRMS, system initialization parameter 90
RST, system initialization parameter 91
RSTSIGNOFF, system initialization parameter 91
RSTSIGNTIME, system initialization parameter 91
RUWAPOOL, system initialization parameter 92

S

screen copying 81
SDSASZE, system initialization parameter 93
SDTRAN, system initialization parameter 93
SDUMP macro
CICS retry interval 42
DURETRY option 42
SEC, system initialization parameter 94
SECPRFX, system initialization parameter 95
security
for transactions 135
health indicator 126
MRO bind-time security 94
of attached entries 135
resource class names 128
SEC, system initialization parameter 94
SECPRFX, system initialization parameter 95
security checking
for EXEC CICS system commands 128
for program entries 132
for temporary storage queues 135
of DB2 resources 128
of EXEC-started transaction entries 131
of file control entries 129
of HFS files 130
of journal entries 130
of PSB entries 133
of resources 133, 134
of transient data queues 129
specifying a prefix to resource name 95
using RACF to establish APPC sessions 94
WLMHEALTH, system initialization parameter 126
XAPPC, system initialization parameter 127
XCMD, system initialization parameter 128
XDB2, system initialization parameter 128
XDCT, system initialization parameter 129
XFCT, system initialization parameter 129
XHFS, system initialization parameter 130
security *(continued)*
XJCT, system initialization parameter 130
XPCT, system initialization parameter 131
XPPT, system initialization parameter 132
XPSB, system initialization parameter 133
XPTKT, system initialization parameter 133
XRES, system initialization parameter 134
XTRAN, system initialization parameter 135
XTST, system initialization parameter 135
single keystroke retrieval (SKR) 95
SIT, system initialization parameter 95
SKR (single keystroke retrieval) 95
SKRxxxx, system initialization parameter 95
SNPRESET, system initialization parameter 96
SNSCOPE, system initialization parameter 96
SOTUNING, system initialization parameter 97
SPCTR, system initialization parameter 98
SPCTRxx, system initialization parameter 98
SPOOL, system initialization parameter 101
SRBSVC, system initialization parameter 101
SRT (system recovery table) 101, 102
SRT, system initialization parameter 101
SRVERCPsystem initialization parameter 101
SSLCACHE, system initialization parameter 102
SSLDELAY, system initialization parameter 102
STANDARD option of system initialization parameter BMS 19
STANDBY start option 104
standby start-up for XRF 104
START, system initialization parameter 102
(option,ALL) 104
STARTER, system initialization parameter 104
STATEOD, system initialization parameter 104
STATINT, system initialization parameter 105
STATRCD, system initialization parameter 105
STGPROT, system initialization parameter 106
STGRCVY, system initialization parameter 106
STNTR, system initialization parameter 107
STNTRxx, system initialization parameter 107

storage protection system initialization parameter, STGPROT 106
storage trace
auxiliary 18
main 59
trace option in transaction dump 120
trace table size in main storage 119
trace table size in transaction dump 119
strings and buffers, VSAM 116, 120
SUBTSKS, system initialization parameter 110
SUFFIX, system initialization parameter 111
supervisor call (SVC)
DFHSI1600 55
DFHSI1601 55
type 3, DFHCSVC 21
type 6, DFHHPSVC 55, 101
surveillance signal for XRF 12
SVC (supervisor call)
type 3, DFHCSVC 21
type 6, DFHHPSVC 55, 101
SYDUMAX, system initialization parameter 111
SYSIDNT, system initialization parameter 111
system identifier, system initialization parameter SYDUMAX 111
system identifier, system initialization parameter SYSIDNT 111
system initialization parameter
ENCRYPTION 44
MINTLSLEVEL 65
system initialization parameters
ADI 12
AIBRIDGE 12
AICONS 12
AIEXIT 13
AILDELAY 13
AIQMAX 14
AIRDELAY 14
AKPFREQ 15
APPLID 15
AUTCONN 17
AUTODST 17
AUTORESETTIME 17
AUXTR 18
AUXTRSW 18
BMS 18
BRMAXKEEPTIME 20
CDSASZE 20
CHKSTRM 21
CHKSTSK 21
CICSSVC 21
CILOCK 22
CLINTCP 22
CLSDSTP 22
CLT 23
CMDPROT 23
CMDSEC 24
CONFDATA 24
CONFTXT 26
CPSMCONN 26
CSDACC 28
CSDBKUP 29
CSDBUFND 29
system initialization parameters *(continued)*
CSDBUFNI 29
CSDDISP 30
CSDDSN 30
CSDFRLOG 30
CSDINTEG 31
CSDJID 32
CSDLSRNO 32
CSDRECOV 32
CSDSTRNO 34
CWAKEY 34
DAE 35
DATFORM 35
DB2CONN 35
DBCTLCON 36
DEBUGTOOL 36
DFLTUSER 37
DIP 37
DISMACP 37
DOCCODEPAGE 37
DSALIM (DSA storage limit) 38
DSHIPIDL 38
DSHIPINT 39
DSRTPGM 39
DTRPGM 40
DTRTRAN 40
DUMP 41
DUMPDS 41
DUMPSW 41
DURETRY 42
ECDSASZE 42
EDSALIM (EDSA storage limit) 43
EODI 45
ERDSASZE 45
ESDSASZE 45
ESMEXITS 46
EUDSASZE 46
FCQRONLY 47
FCT 47
FEPI 48
FLDSEP 48
FLDSTRT 48
FORCEQR 49
FSSTAFF 49
GMTEXT 51
GMTRAN 51
GNTRAN 52
GRPLIST 54
GTFTR 55
HPO 55
HTTPSERVERHDR 56
HTTPUSRAGENTHDR 56
ICP 56
ICV 56
ICVTSD 58
INFOCENTER 58
INITPARM 58
INTTR 59
IRCSTRT 59
ISC 59
JESDI 59
JVMPROFILEDIR 60
KEYRING 60
LGDFINT 60
LGNMSG 61
LLACOPY 61
system initialization parameters *(continued)*
LOCALCCSID 62
LPA 62
MAXOPENTCBS 63
MAXSOCKETS 63, 64
MAXSSLTCBS 64
MCT 64
MN 65
MNCONV 66
MNEXC 66
MNFREQ 67
MNIDN 67
MNPER 67
MNRES 67
MNSYNC 68
MNTIME 68
MQCONN 68
MROBTCH 69
MROFSE 69
MROLRM 70
MSGCASE 70
MSGLVL 70
NATLANG 71
NCPLDFT 72
NEWSIT 72
NISTSP800131A 73
NONRLSRECOV 74
NQRNL 75
OPERTIM 76
OPNDLIM 76
PARMERR 76
PDI 77
PDIR 77
PGAICTLG 77
PGAIEXIT 78
PGAIPGM 78
PGCHAIN 78
PGCOPY 78
PGPURGE 79
PGRET 79
PLTPI 79
PLTPIUSR 80
PLTSD 80
PRGDLAY 81
PRINT 81
PRTYAGE 82
PRVMOD 83
PSBCHK 83
PSDINT 83
PSTYPE 84
PVDELAY 85
RACFSYNC 85
RAMAX 86
RAPOOL 86
RDSASZE 87
RENTPGM 87
RESP 88
RESSEC 88
RMTRAN 90
RRMS 90
RST 91
RSTSIGNOFF 91
RSTSIGNTIME 91
RUWAPOOL 92
SDSASZE 93
SDTRAN 93

system initialization parameters
(continued)
SEC 94
SECPRFX 95
SIT 95
SKRxxxx 95
SNPRESET 96
SNSCOPE 96
SOTUNING 97
SPCTR 98
SPCTRxx 98
SPOOL 101
SRBSVC 101
SRT 101
SRVERCP 101
SSLCACHE 102
SSLDELAY 102
START 102
STARTER 104
STATEOD 104
STATINT 105
STATRCD 105
STGPROT 106
STGRCVY 106
STNTR 107
STNTRxx 107
SUBTSKS 110
SUFFIX 111
SYDUMAX 111
SYSIDNT 111
SYSTR 112
TAKEOVR 112
TBEXITS 113
TCP 113
TCPIP 114
TCSACTN 114
TCSWAIT 115
TCT 115
TCTUAKEY 116
TCTUALOC 116
TD 116
TDINTRA 117
TRANISO (transaction isolation) 117
TRAP 118
TRTABSZ 119
TRTRANSZ 119
TRTRANTY 120
TS 120
TSMAINLIMIT 120
TST 121
UDSASZE 121
UOWNETQL 122
USERTR 122
USRDELAY 123
USSCONFIG 124
USSHOME 124
VTPREFIX 124
WEBDELAY 125
WLMHEALTH 126
WRKAREA 127
XAPPC 127
XCFGROUP 127
XCMD 128
XDB2 128
XDCT 129
XFCT 129
XHFS 130
system initialization parameters
(continued)
XJCT 130
XLT 131
XPCT 131
XPPT 132
XPSB 133
XPTKT 133
XRES 134
XRF 135
XTRAN 135
XTST 135
XUSER 136
z/OS Communications Server 124
System initialization parameters
ICVR 57
MXT 71
system recovery table (SRT) 101, 102
system spooling interface 101
SYSTR, system initialization
parameter 112

T

takeover action for XRF 112
TAKEOVR, system initialization
parameter 112
TBEXITS, system initialization
parameter 113
TCP, system initialization parameter 113
TCPIP, system initialization
parameter 114
TCSACTN, system initialization
parameter 114
TCSWAIT, system initialization
parameter 115
TCT (terminal control table) 115
TCT, system initialization parameter 115
TCTUAKEY, system initialization
parameter 116
TCTUALOC, system initialization
parameter 116
TD, system initialization parameter 116
TDINTRA, system initialization
parameter 117
temporary storage
TSMAINLIMIT 120
VSAM buffers and strings 120
temporary storage table (TST) 121
terminal control table (TCT) 115
terminal control table user area storage
key
system initialization parameter 116
terminal scan delay, ICVTSD 58
time interval, region exit 56
timeout limit, user ID 123
trace
auxiliary storage trace 18
auxiliary trace autoswitch facility 18
AUXTR, system initialization
parameter 18
AUXTRSW, system initialization
parameter 18
CICS standard tracing, setting levels
of 107
GTFTR, system initialization
parameter 55
trace (continued)
INTTR, system initialization
parameter 59
option in transaction dump 120
SM component, warning when setting
trace level 107
SPCTR, system initialization
parameter 98
SPCTRxx, system initialization
parameter 98
special tracing, setting levels of 98
STNTR, system initialization
parameter 107
STNTRxx, system initialization
parameter 107
SYSTR, system initialization
parameter 112
table size in main storage 119
table size in transaction dump 119
TRTABSZ, system initialization
parameter 119
TRTRANSZ, system initialization
parameter 119
TRTRANTY, system initialization
parameter 120
USERTR, system initialization
parameter 122
TRANISO, system initialization
parameter 117
transaction list table, XLT 131
transaction resource monitoring 67
transient data queues
security checking 129
TRAP, system initialization
parameter 118
TRTABSZ, system initialization
parameter 119
TRTRANSZ, system initialization
parameter 119
TRTRANTY, system initialization
parameter 120
TS, system initialization parameter 120
TSMAINLIMIT, system initialization
parameter 120
TST (temporary storage table) 121
TST, system initialization parameter 121

U

UDSASZE, system initialization
parameter 121
UOWNETQL, system initialization
parameter 122
user ID timeout limit 123
USERTR, system initialization
parameter 122
USRDELAY, system initialization
parameter 123
USSCONFIG, system initialization
parameter 124
USSHOME, system initialization
parameter 124

V

virtual telecommunications access
  method (VTAM)
    logon data 60
virtual telecommunications access
  method (z/OS Communications Server)
    logon data 61
virtual terminals
    VTPREFIX 124
VSAM buffers and strings 116, 120
VTAM, system initialization
  parameter 124
VTPREFIX, system initialization
  parameter 124

W

warm start 102
WEBDELAY, system initialization
  parameter 125
welcome (good morning) message 51
WLMHEALTH, system initialization
  parameter 126
write-to-operator timeout limit 76
WRKAREA, system initialization
  parameter 127

X

XAPPC, system initialization
  parameter 127
XCFGROUP, system initialization
  parameter 127
XCMD, system initialization
  parameter 128
XDB2, system initialization
  parameter 128
XDCT, system initialization
  parameter 129
XFCT, system initialization
  parameter 129
XHFS, system initialization
  parameter 130
XJCT, system initialization
  parameter 130
XLT, system initialization parameter 131
XLT, transaction list table 131
XPCT, system initialization
  parameter 131
XPPT, system initialization
  parameter 132
XPSB, system initialization
  parameter 133
XPTKT, system initialization
  parameter 133
XRES, system initialization
  parameter 134
XRF (extended recovery facility)
    ADI (alternate) 12
    AIRDELAY parameter (active and
      alternate CICS) 14
    alternate delay 12
    alternate delay interval 12
    AUTCONN, system initialization
      parameter 17
XRF (extended recovery facility)
  (continued)
    CLT system initialization
      parameter 23
    command list table (CLT) 23
    DUMP system initialization
      parameter 41
    GOOD MORNING transaction 90
    JES delay interval 59
    JESDI system initialization
      parameter 59
    PDI system initialization
      parameter 77, 90
    primary delay interval (PDI) 77
    reconnection delay 17
    reconnection transaction 90
    START=STANDBY (alternate) 104
    surveillance signal 12
    TAKEOVR system initialization
      parameter 112
    XRF system initialization
      parameter 135
XRF, system initialization parameter 135
XTRAN, system initialization
  parameter 135
XTST, system initialization
  parameter 135
XUSER, system initialization
  parameter 136

Z

z/OS Communications Server
    VBUILD TYPE=APPL statement 15

