CHAPTER 3

# Customizing Address Space Operations (IJTCFGxx)

This chapter will help you customize Cisco IOS for S/390 Address Space Operations. This is handled through the Infrastructure (IFS) environment. The IFS configuration member is IJTCFGxx.

IFS is a generic, multitasking, runtime environment for MVS system application address spaces. A system using IFS is an authorized, operator-started task or job that initializes as a subsystem (the primary JES must be initialized first to support SYSOUT requirements).

- IFSPARM Statement

  Describes how to set timing services, time zone, tracename, and GTF ID

- Controlling Message Logging

  Describes how to tailor message logging

- Authorization Key Information

  Describes how to set component authorization key

- Setting SMF Parameters

  Describes how to tailor parameters for SMF statements issued by Cisco IOS for S/390

- Setting Exit Points

  Describes how to configure exit points

- Security Settings

  Describes how to set Security

- POOLDEF Settings

  Describes the pools available for Cisco IOS for S/390

# IFSPARM Statement

The IFSPARM Statement in IJTCFGxx allows you to set GTF ID, timezone, and timing services.

## IFSPARM Syntax

**IFSPARM GTFID (***value***)**

**[DATASPACESIZE (***size***)]**

**[MAXSTGPCT (***below, above***)]**

**[PROMPT | NOPROMPT]**

**[SECONDARYNAME (***name***)]**

**[TIMER (***interval***)]**

**[TIMEZONE (ATLANTIC | EASTERN | CENTRAL |**
**MOUNTAIN | PACIFIC |** *char hours***)]**

**[TRACENAME (***name***)]**

**[VMCFNAME (***vmcf***) PROMPT]**

**[VSREPORT (***interval***) | NOVSREPORT]**

### Syntax Description

| | |
|---|---|
| **GTFID** (*value*) | Specifies the identification value for generalized trace facility (GTF) records produced by Cisco IOS for S/390. This can be specified by a decimal value or a hexadecimal string.<br>Default: X'4F' (79) Range: X'00' - X'4F' (0 to 79) |
| **DATASPACESIZE** (*size*) | Specifies the number of megabytes to allocate for a trace dataspace. This is valid only for a trace address space.<br>Default: 128 |
| **MAXSTGPCT** (*below, above*) | Specifies the upper limit of storage usage above and below the 16MB line when further connections will be stopped.<br>Ranges: below: 50 to 95; above: 50 to 95.<br>Default: (90, 75) |
| P**ROMPT | NOPROMPT** | Specifies whether to prompt to issue message T00IJ13R, which requires a reply, before shutting down the address space.<br>Default: PROMPT |
| **SECONDARYNAME** (*name*) | Specifies the VMCF subsystem name. Valid only for an IUCV address space. |
| **TIMEZONE (ATLANTIC | EASTERN | CENTRAL | MOUNTAIN | PACIFIC |** *char hours*) | Specifies the time zone of local time or a character (that you may specify) and the number of hours before (gmt).<br>Default: EASTERN |

| | |
|---|---|
| **TIMER** (*interval*) | Specifies the time interval for a timer interrupt. This specifies the resolution for timers used in this address space. *Interval* is in 0.01 seconds. Range is 1 to 99.<br><br>Default: 20 |
| **TRACENAME** (*name* | Specifies the subsystem name of the component trace subsystem.<br><br>Default: ACTR |
| **VMCFNAME** (*vmcf*) | Specifies the subsystem identifier used by the IUCV address space. *Vmcf* is a 1 to 4 character string, which, if not specified, will default to the string VMCF. The subsystem name of VMCF must be different from the subsystem name defined as the UNIQ parameter on the execute statement in the startup JCL (in other words, it must not be the same as the subsystem name for Cisco IOS for S/390).<br><br>Default: VMCF |
| **VSREPORT** (*interval*) \| NOVSREPORT | Specifies whether to run the virtual storage report program. It produces a message about virtual storage usage after the specified time to the T01LOG DD statement. *Interval* specifies how often, in minutes, message is to be sent to the account log. The default is 5 minutes, with a maximum of 1439 minutes and a minimum of 1 minute allowed.<br><br>Default: NOVSREPORT |

# IFSPARM Usage Notes

## Timing Services

The IFS timing service, ITIME, allows interval timing to occur in modes other than primary task mode. The heart of the timing service is a fixed interval timer DIE that schedules SRBs in response to timers expiring.

Each timer is capable of tracking eight separate intervals. When the interval time expires, either an exit can be driven or an ECB posted.

The time interval (TIMER) for the DIE is configurable and has a range of 0.1 to 1 second. The time interval is specified in IJTCFG00. Note that specifying a large interval will decrease timer resolution. Specifying a very small interval will increase timer overhead.

### Setting the Time Zone

To set the time zone for your system, specify your choice on the IFSPARM statement with the TIMEZONE parameter. For example, to set the time zone to Eastern Standard Time:

```
IFSPARM TIMEZONE (EASTERN)
```

### Setting the GTF ID

To set the Generalized Trace Facility (GTF) identification value for Cisco IOS for S/390, specify the value on the IFSPARM GTFID statement. For example, to set the value as 17:

```
IFSPARM GTFID 17
```

# Controlling Message Logging

The LOGGING statement in IJTCFGxx specifies the filtering for both print and console messages. Messages are filtered by component and level within component. For instance, specifying:

```
LOGGING PRINT((TC,RFEWISDT)) WTO((TC,FE))
```

prints all TCP messages, but only issues operator messages for fatal and error type TCP messages. Messages for other components are issued at their default levels for both print and operator messages.

The LOGGING statement also controls the spin attributes of the SYSOUT data set and the console routing codes of messages issued by Cisco IOS for S/390.

## LOGGING Statement

**LOGGING [CLASS (***class***)]**

**[DEST (***destination***)]**

**[FORMS (***name***)]**

**[PRINT ((***subparameter***) [, (***subparameter***) [, ... ]])]**

**[ROUTCDE (***list***)]**

**[SPIN (LINES (***lines***) | MINUTES (***minutes***) | SYNC) | NOSPIN]**

**[WRITER (***name***)]**

**[WTO (***subparameter* **[,** *subparameter* **[ , ... ]])]**

## Syntax Description

**CLASS (***class***)**

Specifies the SYSOUT class.

Default: Class specified as SOUT = keyword of PARM field

**DEST (***destination***)**

Specifies the SYSOUT destination.

Default: No destination

**FORMS** (*name*)

Specifies the print form. (Corresponds to FORMS = on the T01LOG DD statement.)

Default: None.

**PRINT** *(subparameter)*

Subparameters are processed left to right. Valid values:

ALL—PRINT all messages, all types

NONE—PRINT no messages

(ALL, types—PRINT given types for all components

component, ALL—PRINT all messages for given component

component, NONE—PRINT no messages for given component

component, types—PRINT given messages for given component

See Component and Message Type Definitions for component and type specifications and defaults.

**ROUTCDE (***list)*

Specifies the MVS routing codes for console messages. *List* can be one or more valid MVS routing codes, separated by commas. Routing code ranges can be specified by separating them with a hyphen.

```
IFSPARM LOGGING ROUTECD(2)
IFSPARM LOGGING ROUTECD(3,4,8-11)
IFSPARM LOGGING ROUTECD(9-11)
```

Default: No routing code. This means console messages are routed according to the defaults specified in the MVS SYSGEN. Range: 1 to 16

**SPIN** (LINES **(***lines***) |** MINUTES **(***minutes***) |** SYNC) | NOSPIN

Determines when the log file will be closed and reopened.

LINES—The number of lines in the log to trigger the spin. The maximum number of lines that can be specified is 2,000,000,000.

MINUTES—number of minutes. The maximum number of minutes that can be specified is 357,913. TIME is an alias for MINUTES.

SYNC—Specifying SYNC with a MINUTES parameter causes a SPIN on the next occurrence of the interval synchronized to the previous midnight. Specifying SYNC without a MINUTES parameter causes a SPIN on the hour, every hour.

Default: NOSPIN

**WRITER** (*name*)

Specifies the member name of an installation-written program in the system library that the external writer loads to write the output data set. (Corresponds to WRITER= on the T01LOG DD statement.)

Note: Do not code INTRDR or STDWTR (and for JES3, NJEWTR) as the writer name. These names are reserved for JES.

**WTO** (*subparameter* [, *subparameter* [,...]])

Subparameters are processed left to right. Valid values:

ALL—WTO all messages, all types

NONE —WTO no messages

ALL,types—WTO given types for all components

component, ALL—WTO all messages for given component

component, NONE—WTO no messages for given component

component, types—WTO given messages for given component

See Component and Message Type Definitions for component and type specifications and defaults.

## Component and Message Type Definitions

Table 3-1 displays the WTO and PRINT components and their message type defaults, according to message destination. Message type is listed in the table and is specified as:

- R—Response
- F—Fatal
- E—Error
- W—Warning
- I—Informational
- S—Statistics
- D—Debugging
- T—Trace

**Table 3-1 Component Names and Message Defaults**

| Message | Description | Print Default Message Level | Console Default Message Level |
|---|---|---|---|
| AP | API Support | RFEWI | RF |
| CF | Configuration | RFEWI | RFEW |
| CO | Commutator | RFEWI | RFE |
| DN | Domain Name Resolver | RFEWI | RFEW |
| EX | Exit Interface | RFEWI | RFEWI |
| GD | GateD | RFEWI | RFEWI |
| IF | IFS | RFEWI | RFEWI |
| IJ | IJT Job Step Task | RFEWI | RFEWI |
| IP | IP Internet Layer | RFEWI | RF |
| IU | IUCV Transport | RFEWI | RFEWI |

**Table 3-1 Component Names and Message Defaults (Continued)**

| Message | Description | Print Default Message Level | Console Default Message Level |
|---|---|---|---|
| LL | Link Layer | RFEWIS | RFEWI |
| NT | Netstat | RFEWI | RFEWI |
| OE | OpenEdition | RFEWI | RFEWI |
| PM | PortMapper Application | RFEWI | RFEWI |
| SF | Server FTP Application | RFEWI | RF |
| SM | SMTP Application | RFEWI | RF |
| SN | SNMP Application | RFEWI | RF |
| SO | Socket API Layer | RFEWI | RFEWI |
| ST | Server Telnet Application | RFEWI | RF |
| S4 | Spool#4 Application | RFEWI | RF |
| TC | Transport (TCP, UDP, RAW) Layer | RFEWI | RF |
| TE | Telnet Echo Application | RFEWI | RF |
| TG | Telnet Character Generator | RFEWI | RF |
| TR | Trace Support | RFEWI | RFEWI |
| UD | server UDP Application | RFEWI | RF |
| UM | User SMTP Application | RFEWI | RF |
| US | USPOOL Application | RFEWI | RF |
| VT | VTAMAPPL Application | RFEWI | RF |
| XL | XLI (Assembler TLI) Transport | RFEWI | RF |

# Examples of Logging

To Write to the Operator (WTO) only messages for the IF component and type Debug, Error, Warning, and Informational messages for the IJ component, use this syntax:

```
WTO(NONE,(IF,ALL),(IJ,DEWI))
```

To print all type Debug and Error messages, but not anything for the SO component:

```
PRINT((ALL,DE),(SO,NONE))
```

To send everything to Write to Operator:

```
WTO(ALL)
```

To print nothing:

```
PRINT(NONE)
```

Note that parameters are processed left to right, as an example:

```
PRINT(ALL,(IJ,I),NONE)
```

The final NONE subparameter overrides the ones before it.

# Authorization Key Information

AUTH information is required for Cisco IOS for S/390 operation. Both customer number and authorization key must be supplied. The initial component authorization key is provided at installation time and determines which Cisco IOS for S/390 components are authorized for your installation. The key has a built-in expiration date. Once the expiration date arrives, you will no longer be able to restart Cisco IOS for S/390 and you will need to contact Customer Support for a new key. After entering the component authorization key, you will need to stop and restart Cisco IOS for S/390 for the authorization key to take effect.

Information on setting the component authorization key for initial startup is included in the *Cisco IOS for S/390 Planning Guide*.

## AUTH Statement Syntax

**AUTH CUSTNUM (*cust_num*)**

**KEY (*auth_key*)**

### Syntax Description

| | |
|---|---|
| **CUSTNUM** (*cust_num*) | Specifies the Cisco IOS for S/390 customer number for component authorization use. |
| **KEY** (*auth_key*) | Specifies the component authorization key. This key must be obtained from Customer Support. The key is case-insensitive and may be entered with spaces to improve readability. |

# Setting SMF Parameters

The TCP/IP stack generates SMF records at various processing points, such as connection establishment, connection termination, connection rejection, and failed connection attempts.

The INTERVAL parameter of the SMF statement determines the frequency (in minutes) with which periodic SMF statistics records, such as virtual storage utilization and driver statistics, are written.

The available record subtypes are described in the *Cisco IOS for S/390 System Management Guide*.

You can set your SMF parameters using the SMF statement in IJTCFGxx.

## SMF Statement Syntax

### Syntax Description

**SMF [TYPE (***number***)]**

**[SUBTYPE (***subtypes***) | ALL | NONE]**

**[INTERVAL (***minutes***)]**

| | |
|---|---|
| **TYPE (***number***)** | Specifies the SMF record type to be assigned to records generated by Cisco IOS for S/390. Any number from 128 through 255 not in use by your installation is available.<br><br>Default: 130 |
| **SUBTYPE** (*subtypes*) **\| ALL \| NONE** | Specifies record subtypes.<br><br>*subtypes* is the list of subtype numbers. The available record subtypes are described in the *Cisco IOS for S/390 System Management Guide*.<br><br>ALL \| NONE specifies if SMF recording is performed.<br><br>ALL specifies that all Cisco IOS for S/390 SMF record subtypes are generated.<br><br>NONE specifies no SMF recording is performed.<br><br>Default: NONE |
| **INTERVAL** (*minutes*) | Specifies, in minutes, the time interval at which periodic SMF records are to be written. For more information, refer to the *Cisco IOS for S/390 System Management Guide*.<br><br>Records of subtype 80 and 100 are interval driven. All other types are event driven. If subtype 80 or 100 records are chosen, but no INTERVAL is configured, a default INTERVAL of 15 minutes will be used.<br><br>Range: 1 -1439 minutes<br><br>Default: 0 |

## SMF Usage Notes

### Record Subtypes

The absence of an SMF statement in the configuration indicates that no SMF recording is performed.

## SMF Examples

These examples show the usage of the SMF statement:

```
SMF TYPE(189) ALL
SMF TYPE(241) SUBTYPE(21)
SMF NONE
```

# Setting Exit Points

Use the EXIT statement to define global user exits to Cisco IOS for S/390. Each EXIT statement defines an exit program to Cisco IOS for S/390, and specifies at which exit point(s) the program will be invoked.

In addition to the exit points listed below, each exit program will receive control at the INIT and TERM exit points. For more information on the usage of the Exit facility, read the *Cisco IOS for S/390 Planning Guide*.

## EXIT Statement Syntax

**EXIT PROGRAM (***program_name***)**

**[PARM (***string***) ,]**

**[FTPLOGIN | NOFTPLOGIN | AUTOFTPLOGIN, ]**

**[FTPRSRCE | NOFTPRSRCE | AUTOFTPRSRCE, ]**

**[LOG | NOLOG | AUTOLOG, ]**

**[TCPBIND | NOTCPBIND | AUTOTCPBIND, ]**

**[SYNRCVD | NOSYSRCVD | AUTOSYSRCVD, ]**

**[SENDSYN | NOSENDSYN | AUTOSENDSYN, ]**

**[TCPESTAB | NOTCPESTAB | AUTOTCPESTAB, ]**

**[TCPCLOSE | NOTCPCLOSE | AUTOTCPCLOSE, ]**

**[UDPBIND | NOUDPBIND | AUTOUDPBIND, ]**

**[UDPSEND | NOUDPSEND | AUTOUDPSEND, ]**

**[UDPRECV | NOUDPRECV | AUTOUDPRECV, ]**

**[RAWSOCK | NORAWSOCK | AUTORAWSOCK, ]**

**[RAWSEND | NORAWSEND | AUTORAWSEND, ]**

**[RAWRECV | NORAWRECV | AUTORAWRECV]**

### Syntax Description

| | |
|---|---|
| **EXITPRGRAM** *program_name*) | Identifies the name of the exit program to be invoked. This program must be made available to Cisco IOS for S/390 at startup. |
| **PARM** (*string*) | Any string of data that is desired. This string will be passed, uninterpreted, to the program at the INIT exit point. |
| **FTPLOGIN \| NOFTPLOGIN \| AUTOFTPLOGIN** | Specifies whether the FTPLOGIN exit point will be called (FTPLOGIN) or not (NOFTPLOGIN), or whether it will be left to the INIT exit to specify (AUTOFTPLOGIN).<br>Default: AUTOFTPLOGIN |

| | |
|---|---|
| **FTPRSRCE \| NOFTPRSRCE \| AUTOFTPRSRCE** | Specifies whether the FTPRSRCE exit point will be called (FTPRSRCE) or not (NOFTPRSRCE), or whether it will be left to the INIT exit to specify (AUTOFTPRSRCE).<br><br>Default: AUTOFTPRSRCE |
| **LOG \| NOLOG \| AUTOLOG** | Specifies whether the LOG exit point will be called (LOG) or not (NOLOG), or whether it will be left to the INIT exit to specify (AUTOLOG).<br><br>Default: AUTOLOG |
| **TCPBIND \| NOTCPBIND \| AUTOTCPBIND** | Specifies whether the TCPBIND exit point will be called (TCPBIND) or not (NOTCPBIND), or whether it will be left to the INIT exit to specify (AUTOTCPBIND).<br><br>Default: AUTOTCPBIND |
| **SYNRCVD \| NOSYSRCVD \| AUTOSYSRCVD** | Specifies whether the SYNRCVD exit point will be called (SYNRCVD) or not (NOSYNRCVD), or whether it will be left to the INIT exit to specify (AUTSYNRCVD).<br><br>Default: AUTOSYNRCVD |
| **SENDSYN \| NOSENDSYN \| AUTOSENDSYN** | Specifies whether the SENDSYN exit point will be called (SENDSYN) or not (NOSENDSYN), or whether it will be left to the INIT exit to specify (AUTOSENDSYN).<br><br>Default: AUTOSENDSYN |
| **TCPESTAB \| NOTCPESTAB \| AUTOTCPESTAB** | Specifies whether the TCPESTAB exit point will be called (TCPESTAB) or not (NOTCPESTAB), or whether it will be left to the INIT exit to specify (AUTOTCPESTAB).<br><br>Default: AUTOTCPESTAB |
| **TCPCLOSE \| NOTCPCLOSE \| AUTOTCPCLOSE** | Specifies whether the TCPCLOSE exit point will be called (TCPCLOSE) or not (NOTCPCLOSE), or whether it will be left to the INIT exit to specify (AUTOTCPCLOSE).<br><br>Default: AUTOTCPCLOSE |
| **UDPBIND \| NOUDPBIND \| AUTOUDPBIND** | Specifies whether the UDPBIND exit point will be called (UDPBIND) or not (NOUDPBIND), or whether it will be left to the INIT exit to specify (AUTOUDPBIND).<br><br>Default: AUTOUDPBIND |
| **UDPSEND \| NOUDPSEND \| AUTOUDPSEND** | Specifies whether the UDPSEND exit point will be called (UDPUDPSEND) or not (NOUDPSEND), or whether it will be left to the INIT exit to specify (AUTOUDPSEND).<br><br>Default: AUTOUDPSEND |
| **UDPRECV \| NOUDPRECV \| AUTOUDPRECV** | Specifies whether the UDPRECV exit point will be called (UDPRECV) or not (NOUDPRECV), or whether it will be left to the INIT exit to specify (AUTOUDPRECV).<br><br>Default: AUTOUDPRECV |

| | |
|---|---|
| **RAWSOCK \| NORAWSOCK \| AUTORAWSOCK** | Specifies whether the RAWSOCK exit point will be called (RAWSOCK) or not (NORAWSOCK), or whether it will be left to the INIT exit to specify (AUTORAWSOCK).<br><br>Default: AUTORAWSOCK |
| **RAWSEND \| NORAWSEND \| AUTORAWSEND** | Specifies whether the RAWSEND exit point will be called (RAWSEND) or not (NORAWSEND), or whether it will be left to the INIT exit to specify (AUTORAWSEND).<br><br>Default: AUTORAWSEND |
| **RAWRECV \| NORAWRECV \| AUTORAWRECV** | Specifies whether the RAWRECV exit point will be called (RAWRECV) or not (NORAWRECV), or whether it will be left to the INIT exit to specify (AUTORAWRECV).<br><br>Default: AUTORAWRECV |

## Using the IEFUSI Sample Exit

When a single Cisco IOS for S/390 region is to service many application users, it may require virtual storage beyond the default provided in most installations. Specifying a private area REGION size greater than 16 MB however, can cause storage allocation problems for system resources below the 16 MB line. In these instances it may be necessary to implement a user exit, such as IEFUSI, to ensure that adequate region values are supplied for Cisco IOS for S/390 operation. Source for a sample IEFUSI exit is provided in the TCPSAMP distribution data set.

**Note** This is a sample only. The region values should be modified to fit your installation's requirements.

# Security Settings

This section describes how to configure the SECURITY Statement.

## Security Statement Syntax

This is the syntax for the Security statement.

**SECURITY CLASS (***class***)**

**PROFILE (***profile***)**

**REQID (***id***)**

**APPLNAME (***name***)**

**SUBSYS (***name***)**

**XSEC (***option1 option2*. .**)**

## Syntax Description

| | |
|---|---|
| **CLASS** (*class* | Specifies the class name to be used for command authorization.<br>Default for RACF: AC#CMD<br>Default for ACF2: AC#CMD<br>Default for RTSS: UR1<br>Default for ALRT: AC#CMD |
| **PROFILE** (*profile*) | Specifies the profile name configured in the security product.<br>Default: SYSTRAN |
| **REQID** (*id*) | Specifies the ID of the caller issuing the security call.<br>Default for RACF: blanks<br>Default for ACF2: ACSECPC<br>Default for RTSS: blanks<br>Default for ALRT: ACSECPC |
| **APPLNAME** (*name*) | Specifies the application name of the caller issuing the security call.<br>Default for RACF: blanks<br>Default for ACF2: blanks<br>Default for RTSS: blanks<br>Default for ALRT: taken from ICSLPID |
| **SUBSYS** (*name*) | Specifies the security subsystem name.<br>Default for RACF: blanks<br>Default for ACF2: taken from ICSLPID<br>Default for RTSS: blanks<br>Default for ALRT: taken from ICSLPID |

**XSEC**
(*option1 option2 ...*)

ACEE | NOACEE —Monitors ACEE activity.

ACSECPC | NOACSECPC—Monitors entry to the security call module that performs all security checking within Cisco IOS for S/390.

COMMAND | NOCOMMAND—Monitors command authorization calls. The COMMAND option aids in the debugging of COMMAND security problems within Cisco IOS for S/390.

DATASET | NODATASET—Monitors data set authorization calls. The DATASET option aids in the debugging of DATASET security problems within Cisco IOS for S/390.

GLBLACT | NOGLBLACT—Activates or suppresses All security calls. If this option is turned off, Cisco IOS for S/390 does no security checking. This could lead to serious problems in an ACF2 environment. Do not turn this option off except at the direction of support personnel.

GLBLCMD | NOGLBLCMD—Activates or suppresses security calls for command authorization. Turns on or off all calls to the external security interface module (ACSECPC) for COMMAND security.

LOGOFF | NOLOGOFF—Monitors departure from the system. The LOGOFF option aids in the debugging of sign-off security problems within Cisco IOS for S/390.

LOGON | NOLOGON—Monitors attempts to gain entry to the system. The LOGON option aids in the debugging of sign-on security problems within Cisco IOS for S/390.

TERMID | NOTERMID—TERMID tells the Cisco IOS for S/390 security interface to place a terminal ID into the Terminal field of the sign-on parameter list for any user attempting a sign-on to Cisco IOS for S/390. The terminal ID passed during sign-on attempts will be either the remote IP address of the originating host for the user, or a VTAM APPL LU name. NOTERMID tells the Cisco IOS for S/390 security interface to not use the Terminal field in the sign-on parameter list during sign-on attempts.

Default: (NOACEE NOACSECPC NOCOMMAND NODATASET NOLOGON NOLOGOFF NOTERMID)

# POOLDEF Settings

The POOLDEF statement is used to define pools of control blocks necessary to run the API and put limits on API usage. Because the API is used internally by the domain name resolver, it must be set up properly for Cisco IOS for S/390 to run. The pool definitions specify an initial amount, an expansion amount, and a minimum amount to limit contraction. You can adjust these numbers to minimize expansion and contraction and improve efficiency. Refer to Defining Control Block Pools (POOLDEF Statement) for more specific information.