Cisco IOS for S/390 Operation

This chapter describes the basic operation of Cisco IOS for S/390. It includes these sections:

- Operator Commands

  Describes the operator commands, including starting and stopping and related **VTAM** commands.

- JCL Requirements

  Describes the JCL issues for Cisco IOS for S/390 operation and how to execute Cisco IOS for S/390 as a started task.

- STARTxx Configuration

  Describes information for the STARTxx member.

- System Commands

  Describes functions, syntax, and arguments of Cisco IOS for S/390 system commands.

- General Task Group Commands

  Describes the **SNAP** and **STATUS** commands.

- Dynamic Configuration Commands

  Describes the dynamic configuration commands.

- APP Commands

  Describes the **REFRESH** command, which is processed by the APP task group.

- DNR Commands

  Describes the **DUMP** command processed by the DNR task group.

- GATED Commands

  Describes commands processed by the GateD (GTD) task group.

- IJT Commands

  Describes the commands processed by the jobstep (IJT) task group.

- TSO Commands

  Describes the TSO commands available for Cisco IOS for S/390 OpenEdition (UNIX System Services) socket users.

- Command Scripts

  Describes how to build command scripts and includes a sample command script and some notes.

Operator Commands

This section describes the operator commands available to Cisco IOS for S/390 users.

Starting and Stopping

Cisco IOS for S/390 can be run as a batch job or as a started task. Many installations route started JCL output to a purge output class. Running Cisco IOS for S/390 as a batch job (as opposed to a started task) may provide valuable information in the JES logs if problems arise. Due to cross-memory services restrictions in MVS, the JES initiator is terminated after Cisco IOS for S/390, running as a batch job, is stopped. For this reason it is probably better to run Cisco IOS for S/390 as a started task. Use the appropriate start command to start the Cisco IOS for S/390 JCL procedure.

Orderly shutdown of Cisco IOS for S/390 notifies the application layer facilities that a shutdown has been requested. These facilities can then terminate prior to the termination of the underlying protocols. Orderly shutdown also lets the IFS task groups that make up Cisco IOS for S/390 (TCP, APP, DNR, MAP, and SNM) terminate gracefully with regard to the various interdependencies and interfaces that exist between them.

If you are using the Inter-User Communications Vehicle (IUCV), examine the T01LOG output data set for message T01IU001I (Connection to IUCV Established). If Cisco IOS for S/390 cannot connect to IUCV, message T01IU004I displays.

If Cisco IOS for S/390 fails to connect to IUCV, it will continue to retry at 30 second intervals until successful connection.

Note The stepname given to the Cisco IOS for S/390 started task becomes the TCPIPJOBNAME that applications use to connect via IUCV to the Cisco IOS for S/390 started task. If no stepname is given, then the jobname (or task name) is used.

Startup

Cisco IOS for S/390 starts by executing IFS. IFS initializes the address space as a whole and then starts the other task groups. To do so, IFS reads and executes the start command script defined by CMND=START*xx* in the PARM field of the EXEC statement and contained in the PARM data set defined by the SYSPROC DD statement.

Each task group is started by an IFS **START** command in the command file. This command, in turn, specifies the configuration file member that contains the parameters to be used by the task group as it starts. The main configuration member for the task group is specified by CNFG(*xx*) on the IFS **START** command for the task group. Some task groups also use secondary configuration members from the PARM data set as specified in their main configuration file. Generally speaking, a task group dynamically loads various other programs and tables from the STEPLIB data set during initialization and later, as needed.

The IJT task group uses IJTCFG*xx*, as this task group controls all other task groups. Therefore, you cannot restart this task group without stopping and restarting the entire Cisco IOS for S/390 address space.

Shutdown

Shutdown can be initiated by these methods:

- By issuing the MVS **STOP** command (**P**) for the Cisco IOS for S/390 job name (for example, **P Cisco IOS for S/390**)
- By using the IFS subsystem **P** command (for example, subsystem recognition character preceding P).
- By using the MVS **MODIFY** command (for example, **F** *jobname*,**P)**

The MVS operator is prompted after the **STOP** command to confirm the shutdown operation.

Note Reference the **NOPROMPT** command in the *Cisco IOS for S/390 Customization Guide* to turn off the WTOR prompt.

The first stop request places Cisco IOS for S/390 into “drain” or “slow shutdown” mode. In this mode, Cisco IOS for S/390 lets existing protocol and API activity continue for an indefinite period of time and does not proceed with other termination functions until the completion of these activities. User Level Protocol Processes (ULPPs) (for example, FTP, SMTP) and API applications are notified that drain mode is in effect. As a result, API applications should begin their termination procedures.

The first P command causes the following for each task group:

- TCP: APEND and TPEND are driven with DRAIN and STOP for all applications and Gated is stopped. If all applications issue TClose and AClose, then TCP continues shutdown automatically, stopping each LNI and freeing resources.
- APP: All Ptasks are notified of termination. They should also see APEND and TPEND from TLI (for example TCP). If all Ptasks terminate, then the APP task group stops.
- MAP: Should exit immediately.
- SNM: Should exit immediately.
- DNR: Attempts to stop the receive, send and timer threads. When all threads have ended, shutdown continues.

Since no new API endpoints or protocol sessions can be established when drain mode is in effect, listening endpoints receive a return code indicating the drain state. The socket interface interprets this return code and terminates certain socket functions in drain mode. Idle server tasks should disconnect from the API in this situation, letting Cisco IOS for S/390 come down.

Fast Shutdown Mode

If a second operator stop request is issued, Cisco IOS for S/390 enters “stop” or “fast shutdown” mode. In fast shutdown mode, any remaining API endpoints and ULPPs are notified to terminate immediately. This results in the termination of all protocol connections and the purging of all pending or outstanding API requests.

The second P command causes the following for each task group:

- TCP: If any active TLI, OE socket, or IUCV socket applications are active, message T01SO010 is issued. This indicates TERM phase and displays the number of active address spaces using API services. APEND and TPEND with the TERM indication is issued to all remaining TLI users. If PROMPT was specified in IJTCFGxx, message T01SO012 is issued asking the operator if shutdown should continue. If the operator replies **Y**, shutdown of TCP continues immediately, otherwise shutdown is delayed.
- APP: Any active Ptasks are again notified to stop. All active TLI connections are immediately closed if still open (usually TPEND STOP from the first P command causes them to close immediately).
- DNR: Continues shutdown.
- MAP: Continues shutdown.
- SNM: Continues shutdown.

Cancel Shutdown Mode

Normally, only one or two operator stop requests are necessary to terminate Cisco IOS for S/390. However, it is possible for API applications to require a third stop request. The third stop request terminates the remaining Cisco IOS for S/390 components regardless of an API application's refusal to issue the required API termination functions. A third stop request also forces detaching of MVS tasks within the APP task group. More than three operator stop requests have no effect on the termination of Cisco IOS for S/390. If necessary, the **MVS CANCEL** command can be used.

The third P command causes the following for each task group:

TCP: Stop immediately regardless of active users. Note that if the operator replied **N** to message T01SO012, this is treated as a second stop and message T01SO012 is reissued.

APP: Stops immediately.

DNR: Stop immediately.

MAP: Stop immediately.

SNM: Stops immediately.

If the address space does not terminate normally after 3 stop commands, issue the **F TCPACCES,ILATCH** command to see if there are any hung latches. If any latches are hung, issue **F TCPACCES,ILATCH FREE LATCH(***nn***)**, where *nn* is the latch number for each hung latch. Shutdown should then proceed. If there are no hung latches after the third **P** command, issue **F TCPACCES,SVCDUMP**. When the SVCDUMP is complete, issue **C TCPACCES** to cancel the address space and contact technical support, supplying the SVCDUMP for problem resolution.

Starting and Stopping Task Groups

The individual IFS task groups within Cisco IOS for S/390 can be stopped and started independently with the IFS subsystem **STOP** and **START** commands.

Example

If the DNR task group terminates abnormally, you can restart it without terminating and restarting the remainder of Cisco IOS for S/390. Read START for more information on restarting task groups.

Note Any task group can be stopped and started to pick up new configuration information, with the exception of the IJT and TCP task groups.

Shutting Down with P versus P CLEAR

The normal method of shutting down Cisco IOS for S/390 is to issue the MVS **STOP (P)** command. If you plan to install maintenance on the Cisco IOS for S/390 base product before restarting Cisco IOS for S/390, use the following command to stop Cisco IOS for S/390:

f *runtcp*,**P CLEAR**

or

%P CLEAR

The **P CLEAR** command specifies to clear modules and control blocks from CSA and the subsystem hooks installed by this address space.

In addition, the **P CLEAR** command also performs the normal processing associated with an **MVS STOP (P)** command. Use the CLEAR option to stop Cisco IOS for S/390 before applying the updates. This command also lets Cisco IOS for S/390 use the most recent versions of the base modules when it is restarted.

The command must be issued using the subsystem recognition character, (in other words, **P CLEAR**) or by using the MVS modify command (in other words, **F Cisco IOS for S/390,P CLEAR**).

When a **P CLEAR** is issued, it is possible that Cisco IOS for S/390 takes an SOC1 or SOC4 during termination. This is normal due to the missing CSA control blocks and modules. Only one **P CLEAR** is needed to clear the CSA.

After a **P CLEAR** command is issued, it may be necessary to follow the additional stopping instructions for Cisco IOS for S/390. For more information, read the section on STOP.

Recycling Task Groups

Although in general it is not needed, individual task groups can be stopped and restarted while Cisco IOS for S/390 is running.

To stop a task group, issue the IFS **STOP** *tgi* command from the MVS operator console.

To restart a task group, issue the **IFS START** *tgi* command from the MVS operator console (*tgi* is the task group identifier). One or more IFS **START** commands can be issued from a command script as shown in the startup example in Startup JCL Customization.

Converting Translate Tables

The **TSO** command **CONVXL8** converts a table from editable text to binary. Read the description of the **CONVXL8** command, later in this chapter.

The TSO command **LOADXL8** works like the **CONVXL8** command, but reads the load module from compiled Cisco IOS for S/390 translate tables as input. Read the description of the **LOADXL8** command, later in this chapter.

Latch Command

The IJT command, **ILATCH**, is a latch lock utility that displays and frees latches used by OpenEdition (UNIX System Services) sockets. It can be used to serialize resources in the local address space. Read the description of the ILATCH command, later in this chapter.

Related VTAM Commands

To activate the major node to VTAM for Cisco IOS for S/390, issue the following command:

```
V NET,ACT,ID=A03ACCES
```

This command also activates the naval LUs required for Server Telnet and the client user commands (for example, **FTP**, **FTP2**, etc.).

JCL Requirements

The JCL provided on the distribution tape is functionally equivalent to that listed in the following topics, but may differ in the order of JCL statements or the contents of comment statements included in the JCL. (See Table 2-1). The sample startup JCL, RUNTCP, is in *TRGINDX*.CNTL(RUNTCP).

JCL Descriptions

Table 2-1 Statements and Descriptions of the Cisco IOS for S/390 JCL

DD Statement	Description
ABNLIGNR DD	Disable the Abend-Aid program product
ARPAHELP DD	Defines a PDS whose members contain the HELP text displayed by Server Telnet and Server FTP
DNRERR DD	DNR error log file
DNRLOG DD	DNR log file.
GTDERR DD	GateD error log file.
GTDLOG DD	GateD log file.
GTDTRC DD	GateD trace log file.
MAPERR DD	Port mapper error log file
MAPLOG DD	Port mapper logging file
SNMLOG DD	SNM task group logging file
STEPLIB DD	Defines the load library for Cisco IOS for S/390; these data sets must be APF-authorized
SYSHELP DD	Defines the help data set for the Cisco IOS for S/390 task groups.
SYSPARM DD	Defines the PARM data set that contains all the configuration file members which provide the parameters for the various task groups within Cisco IOS for S/390.
SYSPRINT DD	DNR and Port Mapper logging file.
SYSPROC DD	Defines the PARM data set from which command scripts (command file members) are obtained by the IJT task group. In particular, the PARM data set contains the command file member STARTxx, which contains the commands to start Cisco IOS for S/390
SYSSNAP DD	Defines the data set to receive dynamic formatted dumps created via the IFS **SNAP** operator command (if no **SNAP** parameter list is specified)
SYSUDUMP DD	Contains formatted system dumps
T01LOG	Defines the log file
SYSABEND	Contains system dumps

Sample JCL for Starting Cisco IOS for S/390

Use this Cisco IOS for S/390 JCL with all interfaces including LOOPBACK. Edit the appropriate member, supplying the correct start member and verifying other symbolic parameters for accuracy.

```
//RUNTCP JOB
//*
//*    SAMPLE JCL PROCEDURE TO RUN TCP/IP
//*    THIS JCL CAN BE USED WITH ANY INTERFACE
//*
//*    EDIT THE TRGINDX, SSN, SRC, SOUT, CMND SYMBOLIC PARAMETERS
//*
//*    VERIFY THAT THE JOB CARD AND NAMING CONVENTIONS MEET
//*    YOUR SITE’S JCL REQUIREMENTS, THEN SUBMIT THIS JOB.
//*
//TCPIP PROC TRGINDX=’TRGINDX’, TARGET LIBRARIES DSN INDEX
//            SSN=ACSS,        DFLT SUBSYSTEM NAME
//            SRC=’%’,         DFLT SUBSYSTEM RECOGNITION CHAR
//            SOUT=’*’,        CHOOSE A HOLD NONPURGE SYSOUT CLASS
//            CMND=START00     DFLT STARTUP COMMAND SCRIPT NAME
//            CNFG=00          IJTCFGxx SUFFIX
//*
//TCPIP EXEC PGM=IFSSTART,REGION=6144K,TIME=1440,
// PARM=’IFSINIT,U=&SSN,P=T01,SR=&SRC,SO=&SOUT,CM=&CMND,CF=&CNFG’
//*
//STEPLIB  DD  DISP=SHR,DSN=&TRGINDX..LOAD
//         DD  DISP=SHR,DSN=&TRGINDX..SASLINK
//*
//* WARNING:  THE LOAD DATA SET MUST NEVER BE ADDED TO THE LINK LIST.
//*           TCPACCESS’ ELEMENT NAMES ARE NOT UNIQUE AND COULD AFFECT
//*           THE OPERATIONS OF OTHER SOFTWARE. THE LOAD DATA SET SHOULD
//*           ALWAYS BE REFERENCED THROUGH A STEPLIB OR JOBLIB STATEMENT.
//*
//*        CONFIGURATION DATA SETS
//*
//SYSPARM  DD  DISP=SHR,DSN=&TRGINDX..PARM
//SYSPROC  DD  DISP=SHR,DSN=&TRGINDX..PARM
//*
//*        LOG DATA SETS
//*
//T01LOG   DD  SYSOUT=&SOUT
//SYSPRINT DD  SYSOUT=&SOUT
//DNRLOG   DD  SYSOUT=&SOUT
//DNRERR   DD  SYSOUT=&SOUT
//GTDLOG   DD  SYSOUT=&SOUT
//GTDERR   DD  SYSOUT=&SOUT
//GTDTRC   DD  SYSOUT=&SOUT
//MAPLOG   DD  SYSOUT=&SOUT
//MAPERR   DD  SYSOUT=&SOUT
//SNMLOG   DD  SYSOUT=&SOUT
//*
//*        DUMP DATA SETS
//*
//SYSUDUMP DD   SYSOUT=&SOUT
//*
//*        MISC DATA SETS
//*
//ARPAHELP DD  DISP=SHR,DSN=&TRGINDX..HELP
//SYSHELP  DD  DISP=SHR,DSN=&TRGINDX..HELP
//ABNLIGNR DD  DUMMY                       /* DISABLE ABEND-AID PROCESSING */
```

Startup JCL Customization

Make the following changes to the startup JCL on the PROC statement:

TRGINDX= Enter the high level qualifier of the Cisco IOS for S/390 data sets.

SSN= Enter the subsystem ID you are using.

Alias for SSN is U. Default is ACSS.

SRC= Enter the subsystem recognition character you are using. (Verify that it does not conflict with any other subsystems you are running.)

Alias for SRC is SUBS. Default is %.

SOUT= Choose an installation defined SYSOUT class that will be held and not automatically purged at job or started task termination. IFS uses this SYSOUT class for dynamically allocated SYSOUT data sets produced during task ABENDs, output from the operator **SNAP** command, and task initialization to print copies of the task startup PARM members.

Alias for SOUT is SYSO. Default is *.

CMND= Enter the name of the STARTxx member in the PARM library.

Alias for CMND is CMD. Default is START00.

CNFG= Enter the suffix of the IJTCFG*xx* configuration member.

Alias for CNFG is CFG. Default is 00.

PRFX= Application prefix. For Cisco IOS for S/390 this is T01.

Alias for PRFX if PFX.

Caution Many installations route started task JCL output to a purge class to be deleted at started task termination. If an installation's default SYSOUT class is purged at job or started task termination and is using SOUT='*', then all dumps and/or SNAPs produced will be lost.

Note If you have not link-listed the Cisco IOS for S/390 LINK library, you must not reference it in the startup JCL. However, if you use any of the client commands such as **FTP2** or **TCPEEP**, you will need to STEPLIB to it in the TSO logon procedure or JCL streams if you run the commands in batch.

These are the only changes required. Copy the JCL stream from *TRGINDX*.CNTL(RUNTCP) to your started task procedure library.

You are now ready to start Cisco IOS for S/390. Before doing so, it is a good idea to review the installation steps in the *Cisco IOS for S/390 Release Notes* to ensure that you have made all the necessary changes.

Starting and Stopping Cisco IOS for S/390

To start Cisco IOS for S/390, issue the MVS **START** command.

To shut down the address space you can use either the subsystem recognition character (**%**) or the MVS **STOP** command (**P Cisco IOS for S/390**). The system will request confirmation with the following message:

```
T01IF013R Confirm request to stop A/S -- Reply 'Y' or 'N'
```

As termination continues, messages will be issued indicating that various components are terminating.

T01LOG Logspin Utility

The T01LOG logspin utility lets you specify that the T01LOG SYSOUT file be closed and re-opened on a regular basis determined by either the number of lines, a period of time, or by a combination of the two. The logspin utility is implemented by including the necessary spin parameter on the LOGGING statement in the IJTCFGxx member. Read the *Cisco IOS for S/390 Customization Guide* for information about the IJTCFG member.

This feature is useful for customers who have long uptimes where spool usage can grow to the point that it impacts Cisco IOS for S/390 performance.

This feature, along with the addition of FREE=CLOSE on the T01LOG DD statements, lets you examine, print, and/or purge output at the convenience of the installation and prevent uncontrolled growth of the spool.

If you specify FREE=CLOSE on the T01LOG DD statement, it will cause the SYSOUT data set to be freed and made available on the output queue. Without this DD parameter, at the closing of the T01LOG file, the output will remain as part of the total job output with each iteration of output appended to the last.

Note The authorization key messages are written to T01LOG causing this file to be opened and closed early in Cisco IOS for S/390 initialization. When the T01LOG logspin feature is active, this will result in the spin-off of this set of messages due to the FREE=CLOSE on the DD statement. There is currently no ability to spin the logs at a specified time of day (in other words, at 8 A.M. each day). You can only specify hours since Cisco IOS for S/390 was started. Do not specify FREE=CLOSE on T01LOG if you are not planning to implement logspin. This will cause the DD to be freed after key authorization and never be reallocated.

STARTxx Configuration

The STARTxx member in the PARM library is an IFS command script that tells Cisco IOS for S/390 which task groups to start and which configuration members to use.

The STARTxx member is the highest level configuration file, as it points to the configuration members for each of the task groups. It is pointed to by the CMND= symbolic parameter in the startup JCL procedure.

The default file, START00, as distributed, contains the following:

```
DISPLAY IFS
DISPLAY SRC
START TCP CNFG(00)
START APP CNFG(00)
START DNR CNFG(00)
START MAP CNFG(00)
START SNM CNFG(00)
SET TEST ON TGB(IJT)
```

Initial STARTxx Customization

The individual commands are described later in this chapter. Do not eliminate any of them. For installation purposes, the ones of interest are the **START** commands for the TCP and DNR task groups. When initially installing Cisco IOS for S/390, there is no need to make changes to the any of the other groups listed.

- Make a copy of the START00 member, giving it a new name (such as START01).
- Change the CNFG parameters on the **START TCP** and **START DNR** commands to match the two character suffixes you used for the TCPCFG*xx* and DNRCFG*xx* parameter files.
- When this has been done, modify the JCL startup procedure and point to your updated STARTxx member.

System Commands

This section describes, in reference form, the functions, syntax, and arguments of all the Cisco IOS for S/390 system commands.

The most commonly used commands are the following:

TASK	Display active task groups.
STATUS	Display maintenance status of a task group.
P CLEAR	Remove system hooks at shutdown.
STOP *TGI*	Stop the task group (indicated by *TGI*).
START *TGI* **CNFG (** *xx* **)**	Start the task group *TGI* using **CNFG(** *xx* **)**, where *xx* is the task group identifier for the task group you are starting.
REFRESH LUPARM (APPLUP*xx* **)**	Refresh LUPOOL and LU information for Server Telnet usage.

Command Format

A command consists of a Subsystem Recognition Character (SRC), optionally followed by a Task Group Identifier (TGI), followed by a command verb, a command object, and usually, by one or more operands. Commands are referred to by the command object. Commands are limited to 126 characters.

Subsystem Recognition Character

The SRC provides the method for an MVS subsystem address space to have operator commands directed to it. The person responsible for installing the subsystem sets SRC and it is specified in the RUNTCP job to start Cisco IOS for S/390. It is one of the parameters used in the execute step for IFSSTART. The SRC can be any valid SRC supported by JES2 or JES3 (JES2 uses $, JES3 uses *). Ask your JES systems programmer what character to use and ensure that it is not a character used by an installation subsystem. The *JES2/JES3 Initialization and Tuning* documentation defines valid SRCs to use when passing commands from local consoles to subsystems (see CONDEF statement, CONCHAR argument, for JES2; and CONSTD statement, SYN argument, for JES3).

SRC is an optional parameter. If no SRC is specified, there will not be an SRC (and all commands issued to the Cisco IOS for S/390 address space must be done via the MVS **MODIFY** command).

Task Group Identifier

Most commands are processed by an implied task group depending on the command object. Some commands provided by Cisco IOS for S/390, such as **SNAP**, can be directed to a specific task group; in this case, the task group identifier must be placed between the SRC and the verb.

Use this command to direct the **SNAP** command to the API task group:

TCP SNAP ALL

The **SNAP** command, entered simply as **SNAP**, is directed to the jobstep task group (IJT) since the task group identifier is omitted. The keyword ALL indicates to **SNAP** the IFS trace table.

These are the valid task group identifiers:

TCP	TCP/IP stack
APP	TCP/IP applications
DNR	Domain Name Resolver
IJT	IFS Jobstep Task
MAP	Port Mapper
SNM	Simple Network Management Agent Task

Verbs

These are the command verbs. If none of these are specified, DISPLAY is assumed. The action taken for each verb is described below:

DISPLAY Displays the status of the command object.

MODIFY Modifies or changes the value of the command object. The keyword ADD, DELETE, or UPDATE usually appears as an operand in this command.

VARY Changes the status of the command object in an orderly manner. The keyword ON or OFF usually appears as an operand in this command.

Some command objects, such as **SNAP**, support only the DISPLAY form and ignore the another specified verb.

You can enter verbs spelled exactly as they are shown or you can use an acceptable abbreviation. You can abbreviate any verb by entering only the significant characters; that is, you must type as much of the verb as is necessary to distinguish it from other verbs. DISPLAY, MODIFY, and VARY can be abbreviated as D, M, and V, respectively.

Objects

Verb action is performed on command objects. **SNAP** is a command object. The **SNAP** command can be entered in these ways:

- **DISPLAY SNAP**
- **SNAP**

You can enter command objects spelled exactly as they are shown or you can use an acceptable abbreviation. You can abbreviate any object by entering only the significant characters. That is, you must type as much of the object as is necessary to distinguish it from other objects.

Operands

Operands provide the specific information required for the command to perform the requested operation.

These operands for the **SNAP** command specify either a JES SYSOUT class or, alternately, an **MVS SNAP**:

```
SNAP CLASS(A)            /* JES SYSOUT CLASS      */
SNAP MVS                 /* MVS SNAP              */
```

Note These examples issue the **SNAP** against the IFS Jobstep Task Group (IJT) by default.

The types of operands used with commands are positional and keyword.

Positional Operands

Positional operands follow the command object in a prescribed sequence.

You must replace the TGI with the actual three-character task group identifier when you enter the command.

When you want to enter a positional operand that is a list of several names or values, the list must be enclosed within parentheses. The names or values must not include unmatched parentheses.

Keyword Operands

Keywords are specific names or symbols that have a particular meaning to the system. You can include keywords in any order following the positional operands. In the command descriptions within this document, keywords are shown in uppercase characters. A typical keyword is ADD.

Some keywords let you specify values. Place the value inside parentheses following the keyword. The following is a typical keyword with a value:

TG (*TGI* **)**

You select the task group identifier desired and substitute that value for TGI when you enter the operand.

TG (IJT)

Note If conflicting keywords are entered, the last keyword entered overrides the previous ones.

Abbreviating Keyword Operands

You can enter keywords spelled exactly as they are shown or you can use an acceptable abbreviation. You can abbreviate any keyword by entering only the significant characters. That is, you must type as much of the keyword as is necessary to distinguish it from the other keywords of the command object.

The **SNAP** command includes the keywords CLASS and MVS. Abbreviations for CLASS are C, CL, CLA, and CLAS; abbreviations for MVS are M and MV.

In addition, some commands allow unique abbreviations or aliases for some of their keywords.

Comments

Comments can be added to a command anywhere a blank might appear. Enter them within the comment delimiters, **/*** and ***/**.

SNAP C   /* OVERRIDE DEFAULT SYSOUT CLASS */

Delimiters

When you type a command, you must separate the command task group identifier, verb, object, and operand(s) from each other by one or more blanks or a comma. Do not use a semicolon as a delimiter because the characters entered after a semicolon are ignored.

POOL (SRB XWA)

General Task Group Commands

These Cisco IOS for S/390 commands can be processed by any active task group in a Cisco IOS for S/390 address space. If the task group identifier is not specified, the command is processed by the jobstep task group (IJT). The commands are **SNAP** and **STATUS**.

SNAP

Use the **SNAP** command for debugging purposes to spin off (dynamically allocate and free) a SYSOUT data set containing a formatted snap dump of control blocks for a task group:

Note The **SNAP** command is not supported for the MAP task group.

[*TGI* **] SNAP [MVS] [ALL] [CLASS (** *SYSOUT_class* **)]**

Syntax Description

TGI — Specifies the three-character task group identifier of the task group to process the command. If not specified, IJT is assumed.

MVS — Specifies an MVS formatted snap dump instead of a Cisco IOS for S/390 formatted snap dump. The output goes to the SYSSNAP DD data set that is not dynamically freed (in other words, the address space must be stopped to make it available for printing if SYSSNAP is a SYSOUT data set).

ALL — Specifies to include the Cisco IOS for S/390 internal trace table in a Cisco IOS for S/390 formatted snap dump. Applications may also include extra information.

CLASS(*SYSOUT_class*) — Specifies an override value for the SYSOUT class to dynamically allocate. The default class is specified by the **SOUT=** symbolic parameter in the JCL procedure for this address space.

Examples

The following are examples of this command:

SNAP

SNAP MVS

SNAP ALL CLASS(A)

STATUS

Use the **STATUS** command to display the maintenance status of a task group. The version and release numbers are displayed.

[*TGI* **] STATUS**

Examples

The following are examples of this command:

STATUS

APP STATUS

Dynamic Configuration Commands

All the configuration variables contained in the TCPCFG*xx* member, which are processed during startup, can be dynamically reprocessed with a set of operator commands. The net effect can be compared to a limited startup with only selected items from the TCPCFG*xx* member, but without recycling the entire Cisco IOS for S/390 address space. For more details, read the description of the TCPCFG*xx* member in the *Cisco IOS for S/390 Customization Guide*.

The **LNI** and **DEVICE** commands activate and de-activate local network hardware interfaces.

The **DELETE** command removes specified items from the active configuration.

The **UPDATE** command is not a single action command, but instead reads a member containing a list of commands and/or configuration statements. The configuration statements are identical to those in the startup TCPCFG*xx* member, and the **LNI**, **DEVICE**, and **DELETE** commands can be included. The default member name is TCPCFGUP, but command syntax allows any name to be used.

The provision to include commands among configuration statements becomes significant where state conditions are enforced. For example, to replace an active LNI, it must first be stopped, then deleted, before the configuration statement is processed. Finally, since this is not in startup, the device must be started manually. The member to the **UPDATE** command can contain a list of these activities, and the entire sequence can be activated with one command.

The distinction between adding to a configuration, or replacing, is based on whether the item itself allows multiples. For example, multiple LNIs are permissible, so a replacement will require a prior deletion. Inversely, a statement like TCP is always a total replacement.

For example, you have a configuration with a MEDIA named CETIETH, and a CETI device with the address 884 and DEVICE name CETI0884. You want to replace the CETI controller with one with an address of E50. The MEDIA name came from the NAME keyword on the MEDIA statement, and the DEVICE name was internally generated. Both names can be displayed with a **NETSTAT CNFG** command.

The typical sequence of steps would be as follows:

1. Extract the CETI statement from the TCPCFG*xx* startup member into a new member (conveniently named TCPCFGUP). This CETI statement will contain the following:

   ```
   DEVADDR(E50),MEDIANAME(CETIETH)
   .
   .
   .
   ```

2. Since this is a replacement, the previous LNI must be stopped and deleted. Ahead of the CETI statement, add the following commands:

   ```
   DEVICE STOP NAME(CETI0884)
   DELETE LNI NAME(CETI0884)
   ```

3. Since this is NOT started, LNI activation is not automatic, so following the CETI statement, add the following:

   ```
   DEVICE START NAME(CETI0E50)
   ```

   The DEVICE name is predictable in this case, as the last 4 characters consist of the hardware channel/device address.

4. Execute the **UPDATE** command.

If this is an addition of a new LNI, rather than a replacement of an existing one, Step 2, above, is not required.

Commands Imbedded in the TCPCFGUP Member

The dynamic configuration commands can be executed as individual commands (entered from the console), or can be imbedded in the TCPCFGUP member to be activated via the **UPDATE** command where they will be executed as individual commands entered from the console. This allows combining a series of sequences, such as deletions and additions, into one member, then executing them all with one command.

All statements in TCPCFGUP are assumed to be additions to the existing configuration.

In addition to the startup statements in the TCPCFGUP member, TCPCFGUP can also contain any of the other three dynamic configuration operator commands. When placed in the TCPCFGUP member, these commands will be executed as individual commands entered from the console. This allows combining a series of sequences, such as deletions and additions, into one member, then executing them all with one command.

DELETE

The DELETE command deletes various configuration components. Specific rules of association and state are enforced. For example, a MEDIA block cannot be deleted until all its NETWORK blocks are also deleted. An LNI cannot be deleted unless it is stopped, and no ingredient belonging to the LOOPBACK configuration can be manipulated at all.

DELETE *target PARAMETER1* **...** *PARAMETERn*

Syntax Description

target Item to be deleted. Choices are:

- ARP – Invalidates an ARP table entry.
- DEVICE – Deletes a device and its associated LNIs. The LNI state must be stopped.
- LNI – Deletes an LNI. If the LNI is the only one associated with a device, the device will also be deleted. Driver configurations that support multiple LNIs tied to one device may require several LNI deletions. The current LNI state must be stopped.
- MEDIA – Deletes a MEDIA block. All associated NETWORKs and LNIs must have been deleted prior to issuing a MEDIA deletion.
- NETWORK – Deletes a NETWORK block.
- ROUTE – Deletes a ROUTE table entry.

PARAMETER The type of *target*. Choices are:

- DEST(*value*) – Dotted decimal IP address; must be specified when deleting a ROUTE entry. See ROUTER, below.
- IPADDR(*value*) – Dotted decimal IP address; must be specified whenever a protocol address is required. Use to identify a NETWORK block and when deleting ARP, NETWORK, or ROUTE entries.

Alias: PA

- MAC(*hex_value*) – 12-character hexadecimal string specifying the 6-byte hardware address used when deleting ARP entries.
- MEDIANAME(*name*) – Must be specified whenever there is an association with a specific MEDIA block. Use when deleting ARP or ROUTE entries. A MEDIA block deletion can be identified by either NAME(*name*) or MEDIANAME(*name*).
- NAME(*name*) – Must be included whenever a specific identification of a DELETE target is required. Use when deleting MEDIA, LNI, or DEVICE entries.
- ROUTER(*value*) – Dotted decimal IP address; must be specified when deleting a ROUTE entry. Both DEST and ROUTER parameters are functionally identical to those on the ROUTE statement in the TCPCFG*xx* startup member.

Alias: GATEWAY

DEVICE

The DEVICE command stops or starts a device. The distinction between an LNI and DEVICE is evident only when the LCS statement has multiple LINKs associated with it. The LCS statement is the DEVICE, and each LINK is an LNI. For all other drivers, there is a one-to-one correspondence between DEVICE and LNI. The internally generated names for both DEVICEs and LNIs are available via the NETSTAT CNFG command.

DEVICE START | STOP NAME (*dev_name* **)**

Syntax Description

START Initiates activity on a DEVICE. The current state must be stopped.

STOP Terminates device activity. The current state must be active.

NAME(*dev_name*) DEVICE name. If the name is not known, issue the **NETSTAT CNFG** command to identify it.

LNI

The LNI command stops or starts an LNI. In the case of a device with multiple LNIs, each of the LNIs must be stopped before the device is stopped; when the last LNI is stopped the device is also stopped.

LNI START | STOP NAME (*lni_name* **)**

Syntax Description

START	Initiates activity on an LNI driver. The current state must be stopped.
STOP	Terminates driver activity. The current state must be active.
NAME(*lni_name*)	Name of the LNI. If the name is not known, issue the **NETSTAT CNFG** command to identify it.

UPDATE

The **UPDATE** command behaves differently from the other dynamic configuration commands. It contains no information itself, but activates commands in a member.

The **UPDATE** command also allows for some items that are not in the startup file, in addition to the other three dynamic configuration commands.

The **UPDATE** command adds new configuration data to an operational gateway. The parameters for the **UPDATE** command specify a member name that contains configuration statements as they would appear in the startup member TCPCFG*xx*. The update member name can default to TCPCFGUP, and must be available in the same DD definitions as the TCPCFG*xx* member.

For more details, read the description of the TCPCFG*xx* member in the *Cisco IOS for S/390 Customization Guide*.

UPDATE [CNFG (*mem_name* **)] [MEMBER (** *mem_name* **)] [IGNORE | TERM]**

Syntax Description

CNFG(*mem_name_suf* **)**

Two character suffix for a member name starting with TCPCFG.

Default: UP

Alias: CFG, CONFIG

MEMBER(*mem_name* **)**

The entire member name. If the default name (TCPCFGUP) is not used, either CNFG or MEMBER should be specified. If both are specified, CNFG is ignored.

Default: TCPCFGUP

Alias: MBR

IGNORE | TERM

Dictates the reaction when encountering an invalid statement in the TCPCFGUP member.

IGNORE issues an error message and proceeds to the next statement

TERM issues an error message and terminates

Default: TERM

APP Commands

This section describes the **REFRESH** command, which is processed by the APP task group.

REFRESH

Use the **REFRESH** command to refresh certain configuration parameters of the APP task group. It can be used to refresh the LU pool or the greeting member used by Server Telnet.

[APP] REFRESH TASK(*n* **) [LUPARM(** *mem_name* **) | GREETING(** *mem_name* **)]**

Syntax Description

APP	Specifies the name of the task group to which the command is directed. This is optional; the REFRESH command is always directed to the APP task group.
TASK	Specifies the task number of the task within the APP task group to which the command is to be directed. If only one APP task is active, specify TASK(1).
LUPARM(*mem_name*)	Specifies the member of the SYSPARM configuration data set from which the refresh is to be performed. The LU pool will be refreshed from this member.
GREETING(*mem_name*)	Specifies the member of the ARPAHELP data set from which the new Server Telnet greeting is to be read. Subsequent Telnet sessions will be presented with the greeting found in this member.

Note LUPARM and GREETING are mutually exclusive.

Examples

```
REFRESH LUPARM( APPLUP00 ) TASK( 2 )

APP REFRESH TASK ( 1 ) GREETING( GREETING )
```

DNR Commands

This section describes **DUMP** and **PURGE** commands, which are processed by the DNR task group.

DUMP

Use the **DUMP** command to produce formatted dumps of the DNR cache, the configuration table, or both. These dumps are written to DNRLOG.

```
[ DNR ] DUMP [
   CACHE [ ( DATA ( xxxx ) | NAMES ) ] | NAMESERVER ( xxxx )
   STATIC [ (
      ALIAS | HOST | NAMESERVER | NETPREF | NETWORK | RPC |
         PROTOCOL | SEARCHLIST | SERVICES
      )
   ]
]
```

Syntax Description

CACHE Specifies that DNR cache data is to be dumped:

- DATA—Specifies the host name or Internet address for which a cache dump is to be performed.
- NAMES—Specifies that all cached domain names be dumped. This is the default if CACHE is specified with no operands.

NAMESERVER Specifies that, for the specified domain name, the name servers used to resolve names within that domain are to be dumped.

STATIC Specifies that static configuration data is to be dumped. If no argument is specified, all static configuration data is dumped. These are the available arguments:

- ALIAS—Dump DNRALC*xx* configuration data
- HOST—Dump DNRHST*xx* configuration data
- NAMESERVER—Dump DNRNSC*xx* configuration data
- NETPREF—Dump DNRNPC*xx* configuration data
- NETWORK—Dump DNRNET*xx* configuration data
- RPC—Dump DNRRPC*xx* configuration data
- PROTOCOL—Dump DNRPRT*xx* configuration data
- SEARCHLIST—Dump DNRSLC*xx* configuration data
- SERVICES—Dump DNRSVCxx

Default

If the **DUMP** command is issued with no arguments, then all cache and static configuration data is dumped.

Usage Guidelines

The DNR task group identifier is optional; the **DUMP** command is automatically directed to the DNR task group, even if it is omitted.

Examples

The following are examples of the **DUMP** command:

```
DNR DUMP CACHE
DNR DUMP CACHE( DATA( A.OUR.COM. ) )
DNR DUMP NAMESERVER( OUR.COM. )
DNR DUMP STATIC
DNR DUMP STATIC( RPC )
```

PURGE

The **PURGE** command removes all entries from the DNR cache. DNR must then access your name server to resolve addresses while the cache is rebuilt. This can be used if your DNR cache contains entries that are no longer valid and you need them to be refreshed immediately.

[DNR] PURGE

GATED Commands

This section describes commands processed by the GateD (GTD) task group.

Note GateD task will *not* automatically restart after a **STOP** command. The **GATED START** command can be used to restart GateD.

DUMP

The **GATED DUMP** command generates a formatted dump.

```
GATED DUMP
```

RELOAD

The **GATED RELOAD** command reinitializes GateD (in other words, it will reread the GTDCFG*xx* configuration file).

```
GATED RELOAD
```

SCAN

The **GATED SCAN** command performs an immediate rescan of the interface.

```
GATED SCAN
```

START

The **GATED START** command starts the GateD subtask, using the configuration member name supplied. If no name is supplied, it will use the one specified on the GATED parameter of the IP statement of TCPCFGxx. If there is no GATED parameter, an error is returned.

```
GATED START [ CNFG( config_mem_name ) ]
```

STOP

The **GATED STOP** command shuts down GateD.

```
GATED STOP
```

TRACE

The **GATED TRACE** command suspends or resumes tracing (in other words, this is a toggle).

```
GATED TRACE
```

IJT Commands

This section describes commands processed by the jobstep (IJT) task group.

GTF

Use the **GTF** command to display or modify the settings of GTF trace event flags. A trace event is recorded only when an event is turned on (with this command) and the task group executing a module that invokes a trace event is in GTF mode (for more information, read SET). If there is no jobname the trace applies only to the TCP address space.

[DISPLAY | MODIFY] GTF [ON | OFF]

[EI (*event_id* **[...]) | CB (** *cb_id* **[...]) | MOD (** *mod_name* **[...]) | ALL]**

Syntax Description

DISPLAY | MODIFY Specifies whether to display or modify the settings of GTF trace event flags.

ON | OFF Specifies to select only those events that are turned on or off for display, or to turn on or off a specified event(s) for modify.

If neither ON nor OFF is specified with DISPLAY, the on/off state of an event is not considered for inclusion in the display. Either ON or OFF is required with MODIFY.

EI (*event_id*) Select the event identifier(s) listed (1-8 alphanumerics).

CB (*cb_id*) Select the events for the control block identifier(s) listed (1-4 alphanumerics).

MOD (*mod_name*) Select the events generated by the module(s) listed (1-4 alphanumerics). A name stem is permitted, for example, if IFSP is specified, all modules with names beginning with IFSP are included.

ALL Select all events.

Examples

The following are examples of this command:

```
GTF OFF
GTF EI( MESSAGE CALLPC SRBDISP )
GTF CB( SSOB ISRB )
MODIFY GTF OFF ALL
MODIFY GTF ON CB( MODI SDWA )
DISPLAY GTF
```

HELP

Use the **HELP** command to obtain online information about the function, syntax, and operands of commands. This reference information is contained in the SYSHELP DD data set(s) and is displayed at your console in response to your request for help. Enter **HELP** without operands to obtain an introduction to using the help facility. Enter the **HELP** command with the operand COMMANDS to obtain a list of all the Cisco IOS for S/390 commands for which help is available.

HELP [*cmnd_name* **| COMMANDS | GENERAL POOL]**

Syntax Description

cmnd_name	Specifies the full name of a command for which help information is requested.
COMMANDS	Requests a list of all commands for which help information is available.
GENERAL	Requests a display of the general format and syntax of Cisco IOS for S/390 commands.
POOL	Requests a display of Cisco IOS for S/390 pools.

Example

The following are examples of this command:

HELP

HELP COMMANDS

HELP POOL

IFS

Use the **IFS** command to display environmental settings for the address space and to display key subsystem-related control block addresses.

IFS

Syntax Description

This command has no arguments or keywords.

ILATCH

Latches are locking mechanisms that can be used to serialize resources more granular than an address space. For example, you would use a latch to serialize resources in the local address space.

Note Cisco IOS for S/390 latches do not use the IBM latch facilities.

If a latch is allocated by a program but not freed, it may cause other programs requesting that same latch to hang.

The **IJT** command **ILATCH** displays and frees latches used by OpenEdition (UNIX System Services) sockets.

The alias for the **ILATCH** command is **ILA**.

There are three different versions of the ILATCH command:

ILATCH [DISPLAY | FREE] [TIME (*seconds* **)] [LATCH (** *latch_num* **)]**

[MSG | NOMSG]

Syntax Description

DISPLAY	Display all latches held more than the time specified by TIME. It can be restricted to display a specific latch number (*latch_num*) if it has been held more than the specified time. Default: DISPLAY
FREE	Free all latches or the latch specified by *latch_num* if held for more than the time specified by TIME(*seconds*).
TIME(*seconds*)	Specifies the time, in seconds, that the latch has been held. Default: 60 seconds
LATCH(*latch_num*)	Specifies the latch number (*latch_num*). Default: All latches
MSG | NOMSG	MSG specifies that the message `T00IF041 END OF ILATCH COMMAND` is displayed. NOMSG suppresses that message. Default: DISPLAY TIME(60) MSG

ILATCH CONTENTION [MSG | NOMSG]

Syntax Description

CONTENTION Lists the history of prior latch contentions.

MSG | NOMSG MSG specifies that the `T00IF041 END OF ILATCH COMMAND` is displayed. NOMSG suppresses that message.

Default: DISPLAY TIME(60) MSG

ILATCH TRON | TROFF | TRACE | TRESET [ENTRIES (***nn*** **)] [KEEP | NOKEEP] [MSG | NOMSG]**

Syntax Description

TRON Activates ILATCH tracing.

TROFF Stops adding to existing table entries. The table itself is not released, and is still available for display.

TRACE Displays table entries. The RC is the return code leaving the GET request. All other displayed information is based on entry to the GET routine.

TRESET Release the storage. This command is not required between a TRON/TROFF cycle and another TRON activation with new parameters, but should be used as a clean-up after all tracing is done.

ENTRIES(*nn* **)** Dictates the table size (the number of entries you want to monitor).

Default: 31; however, this may not be sufficient if KEEP is specified.

KEEP | NOKEEP KEEP controls whether tracing should provide a historic record, or just keep track of unreleased latches. NOKEEP specifies that entries that are released are not retained. The NOKEEP table can be relatively small.

Default: NOKEEP

MSG | NOMSG MSG specifies that the message `T00IF041 END OF ILATCH COMMAND` is displayed; NOMSG suppresses that message.

Default: DISPLAY TIME(60) MSG

LOGGING

Use the **LOGGING** command to reparse and update the entire LOGGING statement, as contained in the IJTCFG*xx* startup member. Any parameter can be changed, within valid limits, and the entire statement can be reprocessed on an active gateway.

LOGGING [CLASS (*class* **)]**
[DEST (*destination* **)]**
[NOW]
[PRINT (*subparm* **[,** *subparm* **[, ...]])]**
[ROUTCDE (*list* **)]**
[SPIN (LINES (*lines* **) | MINUTES (** *minutes* **) | SYNC) | NOSPIN]**
[WTO (*subparm* **[,** *subparm* **[, ...]])]**

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.

NOW

When this parameter is issued through the console, an immediate logspin is done.

PRINT (*subparm*)

Subparameters are processed left to right. Valid values:

ALL - WTO/PRINT all messages, all types

NONE - WTO/PRINT no messages

(ALL,types - WTO/PRINT given types for all components

(component,ALL - WTO/PRINT all messages for given component

(component,NONE - WTO/PRINT no messages for given component

(component,types - WTO/PRINT given messages for given component

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: 1-16

SPIN (	Determines when the log file will be closed and reopened.
LINES (*lines*) | MINUTES (*minutes*) | SYNC)	LINES - Number of lines to be written to each log file before it is closed and reopened. MINUTES -Duration of time before logout is done. Alias for MINUTES is TIME.
| NOSPIN	SYNC - Synchronizes to the hour. Default: NOSPIN
WTO (*subparameter*)	Subparameters are processed left to right. Valid values: ALL - WTO/PRINT all messages, all types NONE - WTO/PRINT no messages (ALL,types - WTO/PRINT given types for all components (component,ALL - WTO/PRINT all messages for given component (component,NONE - WTO/PRINT no messages for given component (component,types - WTO/PRINT given messages for given component

You can also change logging dynamically with the MODIFY command. For example:

MODIFY *job_name* **LOGGING** *parameter(s)*

MODULE

Use the **MODULE** command to display information about a resident module such as call count and assembly date and time.

MODULE [(*mod_name* **[...]) | ALL | *]**

Syntax Description

mod_name **[...]**	Module name(s) to display (1-8 alphanumerics).
ALL or *	Display all resident modules.

Examples

The following are examples of this command:

MODULE *

MODULE IFSSCALL IFSXPOST

MEM

Use the **MEM** command to display up to 1024 bytes of virtual storage.

MEM *addr* **| * [DECLEN (** *nnn* **) | HEXLEN (** *xxx* **) MOD (** *mod_name* **)]**

Syntax Description

addr or *****

Starting display address. It can be entered as an explicit address or as an asterisk (*) with a module name (MOD(*mod_name*)) parameter.

MOD(*mod_name*)

Module name.

DECLEN |
HEXLEN

Length. It can be specified as decimal (DECLEN(*nnn*)) or hexadecimal (HEXLEN(*nnn*)), with a maximum of 1024.

If not specified, default value = 16.

Alias for DECLEN is LEN; it can be abbreviated to DECL.

HEXLEN has no alias but can be abbreviated to HEXL.

MVS

Use the **MVS** command to display the MVS environment and, optionally, the contents of selected control blocks.

MVS [IFS | JESCT | LNKLST | SCVT | SMCA | SSCT]

Syntax Description

IFS Displays the subsystem communication vector table address and name of each defined IFS-based subsystem.

JESCT Displays the address of the JES control table and the name of the primary JES.

LNKLST Displays the names of the data sets in the MVS Link Library List.

SCVT Displays, in dump format, the MVS secondary communications vector table.

SMCA Displays, in dump format, the SMF Control Area.

SSCT Displays the subsystem communication vector table address and name of each defined subsystem.

Examples

The following are examples of this command:

MVS

MVS SSCT

P

The **P** command terminates all task groups and the address spaces. Optionally, it removes the subsystem hooks installed by this address space at initialization.

P [CLEAR]

Syntax Description

CLEAR Specifies to clear the subsystem hooks installed by this address space before returning to MVS. To update Cisco IOS for S/390 after applying maintenance, use the **CLEAR** option to stop Cisco IOS for S/390 before applying updates. In addition to providing normal stop processing, this also clears control blocks and certain modules in the CSA and lets Cisco IOS for S/390 use the most up-to-date versions of the base modules when it is restarted. S0C1/S0C4 messages during termination after a **P CLEAR** are normal and can be ignored. Only one **P CLEAR** is needed to clear the CSA.

Examples

The following are examples of this command:

```
P
P CLEAR
```

POOL

Use the **POOL** command to display the statistics or attributes of data area pools. A pool is a collection of fixed-length data areas residing in a single MVS storage subpool managed by Cisco IOS for S/390 without the overhead of GETMAIN/FREEMAIN.

POOL [(*pool_name* **[...]) | *] [ATTR]**

Syntax Description

pool_name Specifies the name of pool(s) to be displayed. Options are:

- ATCB — Address space task block.
- DSRB — Domain Name Resolution Request Block.
- FRR — IFS Recovery Element.
- IPTH — IUCV only, path to TCP.
- MB1 — Buffer pool for moving data - 128 bytes.
- MB2 — Buffer pool for moving data - 384 bytes.
- MB3 — Buffer pool for moving data - 640 bytes.
- MB4 — Buffer pool for moving data - 1152 bytes.
- MB5 — Buffer pool for moving data - 1536 bytes.
- MB6 — Buffer pool for moving data - 2048 bytes.
- MB7 — Buffer pool for moving data - 5120 bytes.
- MB8 — Buffer pool for moving data - 10240 bytes.
- MB9 — Buffer pool for moving data - 16384 bytes.
- MBA — Buffer pool for moving data - 32768 bytes.
- MSRB — Message Service Request Block.
- MWA — Module Work Area.
- QCB — Queue Control Element for pools.
- SAW — Socket API function.
- SEPM — Socket endpoint.
- SNMP — SNMP data.
- SPCB — Transport Provider (only 3 required total).
- SRB — IFS Service Request Block.
- STAK — Module Stack Block for workareas.
- XAE — SNMP Request/Response header.
- XWA — Cross Memory Workarea.

ATTR Specifies to display the pool attributes instead of statistics. Pool attributes are the values used to control expansion and minimum or maximum values. (Set by the POOLDEF configuration parameter statement or the default.)

Default values for the **POOL** options are described in the following table:

Table 2-2 POOL Command Option Default Values

Value	Initial	Minimum	Expand	Contract
ATCB	32	32	16	32
FRR	100	200	50	0
IPTH	64	64	32	64
MB1	32	128	16	128
MB2	32	256	16	256
MB3	32	192	16	128
MB4	32	160	16	128
MB5	16	128	8	64
MB6	16	96	4	32
MB7	8	48	8	16
MB8	4	32	4	16
MB9	4	16	4	16
MBA	2	8	2	8
MSRB	200	400	100	0
MWA	100	200	50	75
QCB	100	200	50	0
SAW	64	512	32	256
SEPM	16	256	32	128
SPCE	3	3	1	3
SRB	100	200	50	0
STAK	40	20	20	0
XWA	112	160	24	0

The DSRB, SNMP, and XAE pools have no defaults; you must give explicit values for them.

Note It is best to use the defaults at first, and issue the **POOL** command every so often to display pool usage. If you find pools being expanded and staying at the higher value, you can override the default and specify a higher minimum value.

Examples

The following are examples of this command:

POOL

POOL SRB ATTR

POOL * ATTR

SET

Use the **SET** command to set execution options for a task group.

SET [DEMO | TEST | GTF]

[ON | OFF]

[TG(*TGI* **[...]) | ALL]**

Syntax Description

DEMO | TEST | GTF	Specifies which mode to set: • DEMO—Some messages and/or processing is performed only in DEMO mode. • TEST—Some messages and/or processing is performed only in TEST mode. Some API and APP trace events are recorded only in TEST mode. • GTF—GTF events are recorded only when GTF mode is on.
ON | OFF	Specifies to turn the specified mode on or off.
TG(*TGI* [...]) | ALL	Specifies the three-character task group identifier affected by the mode change or ALL. If not specified, all active task groups are affected.

Examples

The following are examples of this command:

SET DEMO ON

SET GTF OFF TG(APP DNR)

SET TEST OFF TG(IJT)

SET GTF ON

SRC

Use the **SRC** command to display or modify the subsystem recognition character for the address space.

[DISPLAY] SRC

MODIFY SRC [*char* **]**

Syntax Description

char	Specifies the new subsystem recognition character (1 alphanumeric) for the address space. The change is effective immediately. This must be a character acceptable by MVS console services. If no character is entered, there will not be any subsystem recognition character.

Examples

The following are examples of this command:

SRC

MODIFY SRC #

DISPLAY SRC

START

Use the **START** command to start a task group and to specify initialization parameter overrides.

```
START     [ TGI CNFG( xx ) MEMBER( mem_name ) ]
            [ CNFG( xx ) ]
            [ MEMBER( mem_name ) ]
```

Syntax Description

TGI

Specifies the three-character task group identifier of the task group to be started.

CNFG(*xx* **)**

Specifies the two character suffix to be used in constructing the name of the configuration parameter file. The name takes the form TGICFG*xx*, where TGI is the task group identifier specified with the **START***xx* command. If CNFG() is not specified, a value of 00 is assumed for *xx*.

The configuration parameter member resides in the SYSPARM DD data set(s) and provides site-specific configuration parameter values for the task group being started.

MEMBER(*mem_name* **)**

Specifies the name of a command script to be invoked when task group initialization has completed (1-8 alphanumerics). The command script resides in the SYSPROC DD data set(s).

Examples

The following are examples of this command:

```
START DNR CNFG( 01 )
START APP MEMBER( APPCFG05 )
```

STCK

Use the **STCK** command to convert the binary 8-byte clock value to a useful date and time.

```
STCK X'hex_string'
```

Syntax Description

hex_string	Specifies the 8-byte binary clock value as stored by the **STCK** instruction. This value is expressed as a 16-character hex string within quotes.

Example

The following is an example of this command:

```
STCK X'AF82198271FB4401'
```

STOP

Use the **STOP** command to terminate a task group. The task group performs an orderly shutdown before terminating. Static control blocks are left in a state to be reused if the task group is started again. Depending on the specific task group, the stop request can be delayed to let work in progress complete.

STOP [*TGI* **]**

TASK(*n* **)**

Note You cannot STOP a TCP task group. The TASK parameter is valid only when you stop an APP task group. There can be up to 4 APP task groups active concurrently.

Syntax Description

TGI Specifies the three-character task group identifier of the task group to be terminated.

Usage Guidelines

- A second **STOP** command performs fast shutdown.
- A third **STOP** performs CANCEL shutdown.

Example

STOP DNR

STOP APP TASK(1)

SVCDUMP

Use the **SVCDUMP** command to generate a system formatted dump. There are no parameters with this command.

TASK

Use the **TASK** command to display the active task groups. Information displayed includes execution-related flags, dispatch count, and the date/time of task group initialization.

TASK [(*TGI* **[...])]**

Syntax Description

TGI Specifies the three-character task group identifier of the task group to be displayed.

Default

If *TGI* is omitted, all task groups are displayed.

Examples

The following are examples of this command:

TASK

TASK (TCP DNR)

TIME

Use the **TIME** command to display the current date and time in all useful forms. This command uses no parameters.

TRACE

Use the **TRACE** command to display the current trace table status, to turn internal tracing on or off, and to set the internal trace table size. The trace table is formatted and included in an IFS-formatted snap dump if an ABEND occurs.

[DISPLAY] TRACE

MODIFY TRACE [ON | OFF] [SIZE(*number* **| 16)] [FIXED]**

Syntax Description

ON | OFF Specifies to enable or disable internal tracing

SIZE(*number*) Specifies the size (number) of the trace table as a number of 4K (4096) pages (1 or 2 decimal digits). Each entry in the trace table is 64 bytes long. The maximum number of 4K pages is 2048.

Use the SIZE parameter only when ON is also specified.

Default: 16 4K pages(64K)

FIXED Specifies that the trace table be in fixed storage, allowing trace capture of I/O-related events. Normally this facility is not required, but may sometimes be requested for diagnostic purposes.

Examples

The following are examples of this command:

MODIFY TRACE OFF

TRACE

MODIFY TRACE ON

MODIFY TRACE ON SIZE(8)

VSM

Use the **VSM** command to display virtual storage usage statistics. This command uses no parameters.

TSO Commands

This section describes the TSO commands available for Cisco IOS for S/390 OpenEdition (UNIX System Services) socket users.

CONVXL8 Command

The TSO command CONVXL8 converts a table from editable text to binary.

CONVXL8 creates a data set with three records. Each record is 256 bytes in length. The first record has “*TCP/IP translate tables” (in EBCDIC) starting in column one, with the remainder of the record padded with EBCDIC blanks (X’40’). The second record will have 256 EBCDIC values representing the ASCII-to-EBCDIC translation. The third record will have 256 ASCII values representing the EBCDIC-to-ASCII translation.

File names use TSO prefix as defined by TSO rules. A fully qualified data set name needs to be enclosed in quotes. A data set name without quotes may have a user specified prefix placed before the name. This prefix is defined by the user’s TSO profile. Refer to TSO documentation for more information about prefixes.

```
CONVXL8 INPUT OUTPUT
```

Command Syntax

INPUT

Specifies the source data set to be converted. The data set must be in standard IBM format for SBCS translation tables. If input is a PDS member, *INPUT* should be specified as dsname(*member*).

This parameter is required.

Note: Cisco IOS for S/390 translate tables in the SAMP data set are not in this format. Use the TSO command LOADXL8 to prepare them for use with OpenEdition (UNIX System Services). Read the section LOADXL8 Command, following, for more information.

Default: None

OUTPUT

Specifies the output data set created by the conversion. If output is a PDS member, *OUTPUT* should be specified as dsname(*member*).

This parameter is required.

Default: None

Examples

This command will read USER.LIB.SOURCE(TRANS) and create a translate table in USER.LIB.TRANTAB:

```
CONVXL8 LIB.SOURCE(TRANS) LIB.TRANTAB
```

This command will read SYSTEM.TCP.DATA(TRAN) and create a translate table in SYSTEM.BIN.TRANS:

```
CONVXL8 ‘SYSTEM.TCP.DATA(TRAN)’  ‘SYSTEM.BIN.TRANS’
```

LOADXL8 Command

LOADXL8 has the same functionality as the **CONVXL8** command, but it reads the load module from compiled translate tables as input. Note that it does not read the source. It loads the module from STEPLIB or TSO **TASKLIB**. The load module referred to is the Cisco IOS for S/390 translate table used by the rest of Cisco IOS for S/390. It can be converted for OpenEdition (UNIX System Services) use with this command.

LOADXL8 *MODULE OUTPUT*

MODULE Specifies the load module name in the STEPLIB/TASKLIB data sets.

Note: You do not explicitly specify the data set name in the command line; just the member name.

Default: None

OUTPUT Specifies the destination of the output data set created by the conversion. If output is a PDS member, OUTPUT should be specified as dsname(*member*).

This parameter is required.

Default: None

These output files are used by OpenEdition (UNIX System Services) DNR services only. Cisco IOS for S/390 continues to use its own translation load modules within the product. You can convert several members and place them in a PDS but OpenEdition (UNIX System Services) will use only the one placed in the sequential data set ‘PREFIX.STANDARD.TCPXLBIN’.

LOADXL8 ENGLISH ‘PREFIX.STANDARD.TCPXLBIN’

Command Scripts

This section describes how to build command scripts that are a prearranged executable sequence of IFS task group commands and command script statements that can be invoked by specifying the command script name prefixed with a percent sign (%). The START*xx* PARM member is an example of a command script.

Command scripts are read from the SYSPROC DD data set(s) and a sample **START00** script is provided in the PARM data set. A command script can be invoked at address space initialization by specifying the command script name with the CMND= parameter in the JCL procedure for an IFS address space. Command scripts can also be invoked using the **IJT START** command with the MEMBER(*xxx*) option.

The command script to start Cisco IOS for S/390 is member STARTxx in the PARM data set. The command data set is specified by DDNAME SYSPROC in the run-time JCL. Member STARTxx is specified by CMND=STARTxx in the PARM field on the EXEC statement. Member STARTxx was established during installation and customization as described in the *Cisco IOS for S/390 Customization Guide*. Alternate STARTxx members can be created and overridden on the PROC CMND field, in other words, S RUNTCP,CMND=START99.

Sample Command Script

The following is a sample command script to start GTF and turn tracing on for some events in a task group named *TGI*, and then to invoke another command script named *TGI*CMNDS:

```
SET GTF OFF                        /*  TURN GTF MODE OFF FOR ALL TASK GROUPS       */
SET GTF ON TG(TGI)                 /*  TURN GTF MODE ON FOR TGI                    */
MOD GTF OFF ALL                    /*  TURN OFF ALL GTF EVENTS                     */
MOD GTF ON CB(SDWA PARM MODI)      /*  TURN ON DESIRED EVENTS                      */
%TGICMNDS                          /*  INVOKE COMMAND SCRIPT FOR TGI TASK GROUP    */
```

Special command statements valid only within a command script are provided to do these tasks:

- Control the display of command statements before being executed
- Control command statement processing if a command fails

Syntax Description

The special commands are as follows:

FLUSH | NOFLUSH FLUSH specifies that the command input stack be purged (flushed) when execution of a command statement fails. This is useful to suppress further command execution if a critical command in a sequence fails.

NOFLUSH specifies that the command input stack continue to be processed even if execution of a command statement fails.

Default is NOFLUSH.

LIST | NOLIST LIST specifies that command statements should be displayed before execution.

NOLIST specifies that command statements not display.

Default is LIST.

Notes

- The SRC should not be specified.
- A complete command statement must be contained in one input source record.
- Comment statements can be included anywhere; begin a comment statement with an * in column one or by enclosing the comment in /* and */ (for example, /* *comment* */).
- Sequence numbers, if included in source input, are assumed to be in the last eight columns for fixed-length records and in the first eight columns for variable-length records. Otherwise, the entire record is assumed to contain text.