A through Z Commands

shell environment load

To download Cisco IOS Shell (IOS.sh) environment from a specified file to the current TTY, use the **shell environment load**command in privileged EXEC mode.

shell environment load *filename*:*URL*{**merge**| **replace**}

Syntax Description

filename:URL	The URL of the shell environment file.
merge	Merge into the current shell environment.
replace	Replace the current shell environment

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.1(4)M	This command was introduced.
15.1(2)S	This command was integrated into Cisco IOS Release 15.1(2)S.
15.1(1)SG	This command was integrated into Cisco IOS Release 15.1(1)SG.
15.1(1)SY	This command was integrated into Cisco IOS Release 15.1(1)SY.

Usage Guidelines

Use the **shell environment load** command to download the Cisco IOS.sh environment from a specified file to the current terminal. Use the **shell environment save** command to save this specific file prior to loading the file using the **shell environment load** command.

Examples

This example shows how to save and then load a Cisco IOS.sh environment file and merge it into the current shell environment :

```
Router> enable
Router# configure terminal
Router(config)#
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# shell processing full
Router# exit
Router# shell environment save disk0:
URL
Router# shell environment load disk0:
URL
merge
```

Related Commands

Command	Description
shell environment save	Saves a Cisco IOS.sh environment functions to a specific file.
show shell environment	Displays a Cisco IOS.sh environment information.

shell environment save

To download Cisco IOS Shell (IOS.sh) environment from a specified file to the current TTY, use the **shell environment load**command in privileged EXEC mode.

shell environment load *filename*:*URL*

Syntax Description

filename:URL	The URL of the shell environment file.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.1(4)M	This command was introduced.
15.1(2)S	This command was integrated into Cisco IOS Release 15.1(2)S.
15.1(1)SG	This command was integrated into Cisco IOS Release 15.1(1)SG.
15.1(1)SY	This command was integrated into Cisco IOS Release 15.1(1)SY.

Usage Guidelines

Use the **shell environment save** command to save all current shell environment, except built-in shell functions, to a specified file. Then only you can use the**shell environment load** command to load the Cisco IOS.sh environment in the specific file on the current terminal.

Examples

This example shows how to save a Cisco IOS.sh environment file:

```
Router> enable
Router# configure terminal
Router(config)#
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# shell processing full
Router# exit
Router# shell environment save disk0:URL
```

Related Commands

Command	Description
shell environment save	Saves a Cisco IOS.sh environment functions to a specific file.
show shell environment	Displays a Cisco IOS.sh environment information.

shell init

To enable Cisco IOS Shell (IOS.sh) initialization options, use the **shell init** command in global configuration mode. To disable the Cisco IOS.sh initialization options, use the **no** form of this command.

shell init {*filename***: no-exec**}

no shell init

Syntax Description

filename:URL	The URL of the shell environment file.
no-exec	Stores the initialization filename and loads the saved environment from that file at the next rebooting of the router.

Command Default

Cisco IOS.sh initialization is disabled.

Command Modes

Global configuration (config)

Command History

Release	Modification
15.1(4)M	This command was introduced.
15.1(2)S	This command was integrated into Cisco IOS Release 15.1(2)S.
15.1(1)SG	This command was integrated into Cisco IOS Release 15.1(1)SG.
15.1(1)SY	This command was integrated into Cisco IOS Release 15.1(1)SY.

Usage Guidelines

Use the **shell init**command to provide Cisco IOS.sh environment access to everyone who logs on to the router. If this command is configured, during system boot up, the router automatically reads and loads the content of the shell configuration from the given file which includes the user-defined variables and functions created previously. It copies the saved files into the Cisco IOS.sh environment. When you use the **no-exec** keyword, it allows you to store the initialization filename and loads the saved environment from that file at the next rebooting of the router.

Note This command is similar to the **show environment load** command which allows you to download the Cisco IOS.sh environment operations.

Examples

This example shows how to enable Cisco IOS.sh processing in all IOS modes:

```
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# shell processing full
Router(config)# shell init disk0:URL
```

Related Commands

Command	Description
show environment load	Downloads a Cisco IOS.sh environment operations.

shell processing

To restore the default behavior of Cisco IOS Shell (IOS.sh) processing, use the **shell processing** command in global configuration mode. To disable the Cisco IOS.sh functions, use the **no** form of this command.

shell processing full

no shell processing

Syntax Description

full	Enables shell processing.

Command Default

Cisco IOS.sh processing is enabled for other applications to use the shell functions.

Command Modes

Global configuration (config)

Command History

Release	Modification
15.1(4)M	This command was introduced.
15.1(2)S	This command was integrated into Cisco IOS Release 15.1(2)S.
15.1(1)SG	This command was integrated into Cisco IOS Release 15.1(1)SG.
15.1(1)SY	This command was integrated into Cisco IOS Release 15.1(1)SY.

Usage Guidelines

To enable shell processing and access all its functions, it is recommended that you use the **shell processing full**command. This allows you the ability to use shell processing in the configuration level without entering the **terminal shell** command on your terminal at the EXEC level. Use the **no shell processing** command to disable Cisco IOS.sh processing on the router. To restore the default Cisco IOS.sh processing, use the **shell processing** command. To reenable shell processing and access all its functions, it is recommended that you use the **shell processing full**command.

You can turn shell processing on the terminal by using the **terminal shell**EXEC command. However shell processing feature is only on while the terminal is running. Once the terminal is turned off, shell processing is off. When the **terminal shell** command is used, shell processing is not visible in the running configuration because it is only on the terminal level and is not in the configuration level. It is convenient to use the **terminal shell** command at the terminal level to quickly access the Cisco IOS.sh man commands.

To enable shell processing and access all its functions in the configuration, it is recommended that you use the **shell processing full** command.

Examples

This example shows how to enable Cisco IOS.sh processing in all Cisco IOS configuration modes:

```
Router> enable
Router# configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)# shell processing full
```

Related Commands

Command	Description
terminal shell	Enables Cisco IOS.sh functions on the router.

show shell environment

To display shell environment information about user-defined functions, built-in functions, and user-created variables, use the **show shell environment**command in privileged EXEC mode.

show shell environment

Syntax Description

This command has no arguments or keywords.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.1(4)M	This command was introduced.
15.1(2)S	This command was integrated into Cisco IOS Release 15.1(2)S.
15.1(1)SG	This command was integrated into Cisco IOS Release 15.1(1)SG.
15.1(1)SY	This command was integrated into Cisco IOS Release 15.1(1)SY.

Usage Guidelines

Use this command to display the shell variables used on the router.

Examples

The following example displays both the shell environment variables and shell functions. Field names are self-explanatory

```
Router# show shell environment
# Environment Variables:
# User Environment Variables:
?=0
VAR1=value1
VAR2=value2
prc_change_mode=PRC_IMMEDIATE
prc_change_type=PRC_CONFIG_CHANGE
prc_error_code=PRC_SUCCESS
prc_failure_type=PRC_INVALID
prc_ha_sync=PRC_HA_SYNC
# Global Environment Variables:
# Builtin Environment Variables:
PATH=CLI%Userfunctions%Builtins%SYSTEM
# Environment Functions:
# User Environment Functions:
Function namespace: DEFAULT
function enable_archive()
{
 configure terminal
 archive
 path disk0:backup
 write-memory
 end
}
```

```
Function namespace: DEFAULT
function enable_logging()
{
 configure terminal
 archive
 logging enable
 record rc
 end
}
# Global Environment Functions:
# Builtin Environment Functions:
Function namespace: DEFAULT
((              evaluate a numeric test expression
Function namespace: DEFAULT
[[              evaluate a logical test expression
Function namespace: DEFAULT
cat             output data from a pipe or file to the terminal
Function namespace: DEFAULT
cut             edit piped output
Function namespace: DEFAULT
echo            echo arguments to the terminal
Function namespace: DEFAULT
false           return false in while or if expressions, and set the result
Function namespace: DEFAULT
fetch           return values from the configuration database
Function namespace: DEFAULT
grep            search for regular expressions in piped output or files
Function namespace: DEFAULT
head            print the first lines in the input
Function namespace: DEFAULT
interface       print interfaces that match the argument
Function namespace: DEFAULT
let             evaluate a numeric expression, and set the result
Function namespace: DEFAULT
man             print information for builtins
Function namespace: DEFAULT
more            page piped output to the terminal
Function namespace: DEFAULT
nl              number the lines in the input
Function namespace: DEFAULT
null            ignore the input
Function namespace: DEFAULT
printf          output formatted data to the terminal
Function namespace: DEFAULT
read            read input into variables
Function namespace: DEFAULT
set_oper        set operational values
Function namespace: DEFAULT
sleep           pause execution of the terminal
Function namespace: DEFAULT
sort            sort the input
Function namespace: DEFAULT
tail            print the tail of the input
Function namespace: DEFAULT
true            return true in while or if expressions, and set the result
Function namespace: DEFAULT
uname           print system information
Function namespace: DEFAULT
wc              count lines, words, and chars
```

Related Commands

Command	Description
shell environment load	Downloads Cisco IOS.sh environment from a specified file to the current TTY.

show shell functions

To display information about Cisco IOS Shell (IOS.sh) user-defined and built-in functions, use the **show shell functions** command in privileged EXEC mode.

show shell functions [*functionname*| **brief**]

Syntax Description

functionname	(Optional) Specifies a user-defined shell function.
brief	(Optional) Lists the names of the user-defined and built-in functions.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.1(4)M	This command was introduced.
15.1(2)S	This command was integrated into Cisco IOS Release 15.1(2)S.
15.1(1)SG	This command was integrated into Cisco IOS Release 15.1(1)SG.
15.1(1)SY	This command was integrated into Cisco IOS Release 15.1(1)SY.

Usage Guidelines

Use this command to display the shell information for the router.

Examples

The following example shows how to use the **show shell functions** privileged EXEC command to display the details of the user-defined and built-in functions. Field names are self-explanatory.

```
Router# show shell functions
#User defined functions:
Function namespace: DEFAULT
function enable_archive()
{
 configure terminal
 archive
 path disk0:backup
 write-memory
 end
}
Function namespace: DEFAULT
function enable_logging()
{
 configure terminal
 archive
 logging enable
```

```
 record rc
 end
}
#Global User functions:
#Built-in functions:
Function namespace: DEFAULT
((              evaluate a numeric test expression
Function namespace: DEFAULT
[[              evaluate a logical test expression
Function namespace: DEFAULT
cat             output data from a pipe or file to the terminal
Function namespace: DEFAULT
cut             edit piped output
Function namespace: DEFAULT
echo            echo arguments to the terminal
Function namespace: DEFAULT
false           return false in while or if expressions, and set the result
Function namespace: DEFAULT
fetch           return values from the configuration database
Function namespace: DEFAULT
grep            search for regular expressions in piped output or files
Function namespace: DEFAULT
head            print the first lines in the input
Function namespace: DEFAULT
interface       print interfaces that match the argument
Function namespace: DEFAULT
let             evaluate a numeric expression, and set the result
Function namespace: DEFAULT
man             print information for builtins
Function namespace: DEFAULT
more            page piped output to the terminal
Function namespace: DEFAULT
nl              number the lines in the input
Function namespace: DEFAULT
null            ignore the input
Function namespace: DEFAULT
printf          output formatted data to the terminal
Function namespace: DEFAULT
read            read input into variables
Function namespace: DEFAULT
set_oper        set operational values
Function namespace: DEFAULT
sleep           pause execution of the terminal
Function namespace: DEFAULT
sort            sort the input
Function namespace: DEFAULT
tail            print the tail of the input
Function namespace: DEFAULT
true            return true in while or if expressions, and set the result
Function namespace: DEFAULT
uname           print system information
Function namespace: DEFAULT
wc              count lines, words, and chars
```

The following example shows how to use the **show shell functions brief** privileged EXEC command to display a list of the names of user-defined and built-in functions:

```
Router# show shell functions brief
#User defined functions:
enable_archive
enable_logging
#Global User functions:
#Built-in functions:
((
[[
cat
cut
echo
false
fetch
grep
head
```

```
interface
let
man
more
nl
null
printf
read
set_oper
sleep
sort
tail
true
uname
wc
```

Table 1: Built-in Cisco IOS.sh Functions

Built-in Function	Description
cat	Output data from a pipe or file to the terminal.
cut	Edit piped output.
echo	Echo arguments to the terminal.
false	Return false in while or if expressions, and set the result.
fetch	Return values from the configuration database.
for	Cisco IOS.sh for loops.
grep	Search for regular expressions in piped output or files.
head	Print the first lines in the input.
interface	Print interfaces that match the argument.
let	Evaluate a numeric expression, and set the result.
man	Print information for built-ins.
more	Page piped output to the terminal.
nl	Number the lines in the input.
null	Ignore the input.
printf	Output formatted data to the terminal.
read	Read input into variables.
set_oper	Set operational values.

Built-in Function	Description
sleep	Pause execution of the terminal.
sort	Sort the input.
tail	Print the tail of the input.
true	Return true in while or if expressions, and set the result.
uname	Print system information.
wc	Count lines, words, and chars.

The following example shows how to use the **show shell functions enable_archive** privileged EXEC command to view the user-defined **enable_archive** function:

```
Router# show shell functions enable_archive
#User defined functions:
Function namespace: DEFAULT
function enable_archive()
{
 configure terminal
 archive
 path disk0:backup
 write-memory
 end
}
```

terminal shell

To enable Cisco IOS Shell (IOS.sh) functions on the router use the **terminal shell**command in privileged EXEC mode. To disable Cisco IOS.sh functions, use the **no** form of this command.

terminal shell [trace]

terminal no shell

Syntax Description

trace	Turns on the trace feature on the shell functions that prints the CLI as they are applied.

Add trace if it is available [If the trace option is given, it will turn on the trace on the shell functions that prints the CLI as they are applied.]

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.1(4)M	This command was introduced.
15.1(2)S	This command was integrated into Cisco IOS Release 15.1(2)S.
15.1(1)SG	This command was integrated into Cisco IOS Release 15.1(1)SG.
15.1(1)SY	This command was integrated into Cisco IOS Release 15.1(1)SY.

Usage Guidelines

You can turn shell processing on the terminal by using the **terminal shell**command. However shell processing feature is only on while the terminal is running. Once the terminal is turned off, shell processing is off. When the **terminal shell** command is used, shell processing is not visible in the running configuration because it is only on the terminal level and is not in the configuration level. It is convenient to use the **terminal shell** command at the terminal level to quickly access the Cisco IOS.sh man commands. To enable shell processing in the configuration, it is recommended that you use the **shell processing full**command.

Examples

The following examples shows how to enable Cisco IOS.sh functions on the router and enable the trace feature:

```
Router# terminal shell
Router# terminal shell
Router# terminal shell trace
```

The following examples shows how to disable Cisco IOS.sh functions on the router:

```
Router# terminal no shell
```

Related Commands

Command	Description
shell processing full	Enables Cisco IOS.sh processing on the router in global configuration mode.