action string compare

To compare two unequal strings when an Embedded Event Manager (EEM) applet is triggered, use the **action string compare** command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string compare** [**nocase**] [**length** *integer*] *string1 string2*

no action *label* **string compare**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
nocase	(Optional) Specifies case insensitive comparison.
length	(Optional) Limits the comparison to the first integer character.
integer	(Optional) Valid values for the length argument range from 1 to 4294967295.
string1	Sequence of characters.
string2	Sequence of characters.

Command Default

Unequal strings are not compared.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(22)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

String comparisons are performed on a byte-by-byte basis from left to right. If the strings are of unequal length, the longer string is compared greater than the shorter string. The **action string compare** command forces a comparison between two unequal strings, which is followed by an integer comparison of the result of the string comparison.

When two equal strings are compared, the result is 0 and when one string sorts before the other, the result is -1. For all other comparisons the result is 1. If the strings being compared are converted to integers, the comparison is performed between the results using the **strcmp** command.

Table 7 shows the built-in variable in which the results of the **action string compare** command are stored.

Table 7 EEM Built-in Variables for action string compare Command

Built-in Variable	Description
$_string_result	The result of the **action string compare** command is stored in this variable.

Examples

The following example shows how to compare two unequal strings:

```
Router(config-applet)# event manager applet compare
Router(config-applet)# event none
Router(config-applet)# action 1 set str "this contains some $str"
Router(config-applet)# action 2 string compare nocase length 3 “contains” “$str”
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action string equal

To verify whether or not two strings are equal when an Embedded Event Manager (EEM) applet is triggered, use the **action string equal** command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string equal** [**nocase**] [**length** *integer*] *string1 string2*

no action *label* **string equal**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
nocase	(Optional) Specifies case insensitive comparison.
length	(Optional) Specifies the length of the value to limit the comparison.
integer	(Optional) Valid values for the length argument range from 1 to 4294967295.
string1	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
string2	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.

Command Default

Strings are not verified as equal.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(22)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

The **action string equal** command compares two strings and returns 1 if the strings are equal. Use **nocase** for case insensitive comparison.

Table 8 shows the built-in variable in which the results of the **action string equal** command are stored.

Table 8 EEM Built-in Variables for action string equal Command

Built-in Variable	Description
$_string_result	The result of the **action string equal** command is stored in this variable.

Examples

The following example shows how to verify whether or not two strings are equal:

```
Router(config-applet)# event manager applet equal
Router(config-applet)# event none
Router(config-applet)# action 1 set str "this contains some data"
Router(config-applet)# action 2 string equal "contains" "data"
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action string first

To return the index on the first occurrence of *string1* within *string2* when an Embedded Event Manager (EEM) applet is triggered, use the **action string first** command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string first** *string1 string2* [*index-value*]

no action *label* **string first**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string1	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
string2	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
index-value	(Optional) The index value to start the first test. Number in the range from 0 to 4294967295.

Command Default

The index is not returned on the first occurrence of *string1* within *string2*.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(22)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

On the first occurrence of *string1*, the index is placed in *string2*. If *string1* is not found, it returns -1.

Table 9 shows the built-in variable in which the results of the **action string first** command are stored.

Table 9 EEM Built-in Variables for action string first Command

Built-in Variable	Description
$_string_result	The result of the **action string first** command is stored in this variable.

Examples

The following example shows how to return the index on the first occurrence of *string1* within *string2*:

```
Router(config-applet)# event manager applet first
Router(config-applet)# event none
Router(config-applet)# action 1 set str "this contains some data"
```

```
Router(config-applet)# action 2 string first "contains" "$str"
Router(config-applet)# action 3 puts "$_string_result"
Router# event manager run first
5
```

Related Commands

Command	Description
action string last	Returns the index on the last occurrence of *string1* within *string2*.
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action string index

To return the characters specified at a given index value when an Embedded Event Manager (EEM) applet is triggered, use the **action string index** command in applet configuration mode. To disable this function, use the **no** form of the command.

action *label* **string index** *string* [*value* | *end*]

no action *label* **string index**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
value	(Optional) The index value. Number in the range from 0 to 4294967295. The count starts from 0.
end	(Optional) Last character of the string.

Command Default

The characters specified at a given index value are not returned.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(22)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

The index count starts from zero. Use the *end* argument for the last character of the string.

Table 10 shows the built-in variable in which the **action string index** command stores the characters.

Table 10 EEM Built-in Variables for action string index Command

Built-in Variable	Description
$_string_result	The **action string index** command stores the characters in this variable.

Examples

The following example shows how to return the character specified at a given index value:

```
Router(config-applet)# event manager applet index
Router(config-applet)# event none
Router(config-applet)# action 1 set str "this is text"
Router(config-applet)# action 2 string index "$str" 8
Router(config-applet)# action 3 puts "$_string_result"
```

```
Router# event manager run index
t
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action string last

To return the index on the last occurrence of *string1* within *string* 2 when an Embedded Event Manager (EEM) applet is triggered, use the **action string last** command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string last** *string1 string2* [*index-value*]

no action *label* **string last**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string1	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
string2	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
index-value	(Optional) The index value to start the last test. Number in the range from 0 to 4294967295.

Command Default

The index is not returned on the last occurrence of *string1* within *string2*.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(22)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

On the first occurrence of *string1*, the index is placed in *string2*. If *string1* is not found, it returns -1.

Table 11 shows the built-in variable in which the results of the **action string last** command are stored.

Table 11 EEM Built-in Variables for action string last Command

Built-in Variable	Description
$_string_result	The result of the **action string last** command is stored in this variable.

Examples

The following example shows how to return the index on the last occurrence of *string1* within *string2*:

```
Router(config-applet)# event manager applet last
Router(config-applet)# event none
Router(config-applet)# action 1 set str "this contains some data"
```

```
Router(config-applet)# action 2 string last "contains" "$str"
Router(config-applet)# action 3 puts "$_string_result"
Router# event manager run last
5
```

Related Commands

Command	Description
action string first	Returns the index on the first occurrence of *string1* within *string2*.
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action string length

To return the number of characters in a string when the Embedded Event Manager (EEM) applet is triggered, use the **action string length** command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string length** *string*

no action *label* **string length**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.

Command Default

The number of characters in a string are not returned.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(22)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

Use the **action string length** command to specify the action of returning the number of characters in a string when an EEM applet is triggered.

Table 12 shows the built-in variable in which the results of the **action string length** command are stored.

Table 12 EEM Built-in Variables for action string length Command

Built-in Variable	Description
$_string_result	The result of the **action string length** command is stored in this variable.

Examples

The following example shows how to return the number of characters in a string:

```
Router(config-applet)# event manager applet length
Router(config-applet)# event none
Router(config-applet)# action 1 set str "this contains some data"
Router(config-applet)# action 2 string length "contains"
Router(config-applet)# action 3 puts "$_string_result"
Router# event manager run length
8
Router#
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action string match

To return 1 to the $_string_result, if the string matches the pattern when an Embedded Event Manager (EEM) applet is triggered, use the **action string match** command in applet configuration mode. To disable this action, use the **no** form of this command.

action *label* **string match** [**nocase**] *string-pattern string*

no action *label* **string match**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
nocase	(Optional) Specifies case insensitive comparison.
string-pattern	The pattern for case insensitive comparison.
string	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.

Command Default

Results of the pattern matching of strings are not returned to the $_string_result.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(22)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

When the string matches the specified pattern, the result is 1; when the pattern does not match, the result is 0.

Table 13 shows the built-in variable in which the results of the **action string match** command is stored.

Table 13 EEM Built-in Variables for action string match Command

Built-in Variable	Description
$_string_result	The result of the **action string match** command is stored in this variable.

Examples

The following example shows how to return 1 to the $_string_result if the string matches the pattern:

```
Router(config-applet)# event manager applet match
Router(config-applet)# event none
Router(config-applet)# action 1 set str "this is some text"
```

```
Router(config-applet)# action 2 string match "$str" "this is"
Router(config-applet)# action 3 puts "$_string_result"
Router# event manager run match
1
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action string range

To store a range of characters in a string when an Embedded Event Manager (EEM) applet is triggered, use the **action string range** command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string range** *string start-index end-index*

no action *label* **string range**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string	Sequence of characters which can be up to 4294967295. If the string contains embedded blanks, enclose it in double quotation marks.
start-index	The starting index string value. The range is from 0 to 4294967295.
end-index	The ending index string value. The range is from 0 to 4294967295.

Command Default

A string is not stored.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(22)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

Use the **action string range** command to specify the action of storing a range of characters in a string when an EEM applet is triggered. The *start-index* and *end-index* arguments specify the range of the string on which to operate.

Table 14 shows the built-in variable in which the result of the **action string range** command is stored.

Table 14 EEM Built-in Variables for action string range Command

Built-in Variable	Description
$_string_result	The result of the **action string range** command is stored in this variable.

Examples

The following example shows how to store a range of characters in a specified string:

```
Router(config)# event manager applet store
Router(config-applet)# action 1.0 set string "This is some text"
Router(config-applet)# action 2.0 string range "$string" 0 6
```

```
Router(config-applet)# action 3.0 puts "$_string_result"
Router(config-applet)# end
Router# event manager run store
this is
Router#
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action string replace

To store a new string by replacing the range of characters in the specified string when an Embedded Event Manager (EEM) applet is triggered, use the **action string replace** command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string replace** *string start-index end-index* [*new-string*]

no action *label* **string replace**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string	Sequence of characters, which can be up to 4294967295. If the string contains embedded blanks, enclose it in double quotation marks.
start-index	The starting index string value. The range is from 0 to 4294967295.
end-index	The ending index string value. The range is from 0 to 4294967295.
new-string	(Optional) The sequence of characters that will replace the range of characters in the string.

Command Default

A string is not stored.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(22)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

Use the **action string replace** command to get a new string by replacing specific characters in a particular string. If the value for *new-string* argument is not specified, the characters are replaced with white space.

Table 15 shows the built-in variable in which the result of the **action string replace** command is stored.

Table 15 EEM Built-in Variables for action string replace Command

Built-in Variable	Description
$_string_result	The result of the **action string replace** command is stored in this variable.

Examples

The following example shows how to store the new string made by replacing the specific characters in a string:

```
Router(config)# event manager applet replace
Router(config-applet)# event none
Router(config-applet)# action 1.0 set string "This is some text"
Router(config-applet)# action 2.0 string replace "$string" 0 6 "that was"
Router(config-applet)# action 3.0 puts "$_string_result"
Router (config-applet)# end
Router# event manager run replace
that was some text
Router#
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action string tolower

To store a specific range of characters of a string in lowercase when an Embedded Event Manager (EEM) applet is triggered, use the **action string tolower** command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string tolower** *string* [*start-index*] [*end-index*]

no action *label* **string tolower**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string	The sequence of characters that needs to be replaced. If the string contains embedded blanks, enclose it in double quotation marks.
start-index	(Optional) The starting index string value. The range is from 0 to 4294967295.
end-index	(Optional) The ending index string value. The range is from 0 to 4294967295.

Command Default

A string is not stored.

Command Modes

Applet configuration (applet-config)

Command History

Release	Modification
12.4(22)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

Use the **action string tolower** command to store a specific range of characters of a string in lowercase. The *start-index* and *end-index* arguments specify the range of the string on which to operate.

Table 16 shows the built-in variable in which the result of the **action string tolower** command is stored.

Table 16 EEM Built-in Variables for action string tolower Command

Built-in Variable	Description
$_string_result	The result of the **action string tolower** command is stored in this variable.

Examples

The following example shows how to store a range of characters in a specific string in lowercase:

```
Router(config)# event manager applet lowercase
Router(config-applet)# action 1.0 set string "This is a STRING"
```

```
Router(config-applet)# action 2.0 string tolower "$string" 11 16
Router(config-applet)# action 3.0 puts "$_string_result"
Router(config-applet)# end
Router# event manager run lowercase
string
Router#
```

Related Commands

Command	Description
action string toupper	Stores a specific range of characters of a string in uppercase.
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action string toupper

To store a specific range of characters of a string in uppercase when an Embedded Event Manager (EEM) applet is triggered, use the **action string toupper** command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string toupper** *string* [*start-index*] [*end-index*]

no action *label* **string toupper**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string	Specifies the sequence of characters, that needs to be replaced. If the string contains embedded blanks, enclose it in double quotation marks.
start-index	(Optional) The starting index string value. The range is from 0 to 4294967295.
end-index	(Optional) The ending index string value. The range is from 0 to 4294967295.

Command Default

A string is not stored.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(22)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

Use the **action string toupper** command to store a specific range of characters of a string in uppercase. The *start-index* and *end-index* arguments specify the range of the string on which to operate.

Table 17 shows the built-in variable in which the result of the **action string toupper** command is stored.

Table 17 EEM Built-in Variables for action string toupper Command

Built-in Variable	Description
$_string_result	The result of the **action string toupper** command is stored in this variable.

Examples

The following example shows how to store a range of characters in a specific string in uppercase:

```
Router(config)# event manager applet uppercase
```

```
Router(config-applet)# action 1.0 set string "This is a string"
Router(config-applet)# action 2.0 string toupper "$string" 11 16
Router(config-applet)# action 3.0 puts "$_string_result"
Router(config-applet)# end
Router# event manager run uppercase
STRING
Router#
```

Related Commands

Command	Description
action string tolower	Stores a specific range of characters of a string in lowercase.
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action string trim

To trim a string when an Embedded Event Manager (EEM) applet is triggered, use the **action string trim** command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string trim** *string1* [*string2*]

no action *label* **string trim**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string1	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
string2	(Optional) Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.

Command Default

By default, there is no action to trim a string.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(22)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

Use the **action string trim** command to trim the characters in a string. This command trims the characters in *string2* from both ends of *string1*. By default, *string2* corresponds to white space.

Table 18 shows the built-in variable in which the result of the **action string trim** command is stored.

Table 18 EEM Built-in Variables for action string trim Command

Built-in Variable	Description
$_string_result	The result of the **action string trim** command is stored in this variable.

Examples

The following example shows how to trim a string:

```
Router(config)# event manager applet trim
Router(config-applet)# action 1.0 set string "Hello How are you?Hello"
Router(config-applet)# action 2.0 string trim "$string" "Hello "
Router(config-applet)# action 3.0 puts "$_string_result"
Router(config-applet)# end
Router# event manager run trim
```

```
How are you?
```

Related Commands

Command	Description
action string trimleft	Trims the characters by one string from the left end of another string.
action string trimright	Trims the characters by one string from the right end of another string.
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action string trimleft

To trim the characters of one string from the left end of another string when an Embedded Event Manager (EEM) applet is triggered, use the **action string trimleft** command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string trimleft** *string1* [*string2*]

no action *label* **string trimleft**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string1	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
string2	(Optional) Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.

Command Default

By default, there is no action to trim a string.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(22)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

Use the **action string trimleft** command to trim a string from the left end of another string. This command trims the characters specified by *string2* from the left end of *string1*. By default, *string2* corresponds to white space.

Table 19 shows the built-in variable in which the result of the **action string trimleft** command is stored.

Table 19 EEM Built-in Variables for action string trimleft Command

Built-in Variable	Description
$_string_result	The result of the **action string trimleft** command is stored in this variable.

Examples

The following example shows how to trim a string from the left side of another string:

```
Router(config)# event manager applet trimleft
Router(config-applet)# action 1.0 set string "Hello How are you?"
Router(config-applet)# action 2.0 string trimleft "$string" "Hello "
Router(config-applet)# action 3.0 puts "$_string_result"
```

```
Router(config-applet)# end
Router# event manager run trimleft
How are you?
Router#
```

Related Commands

Command	Description
action string trim	Trims a string.
action string trimright	Trims the characters by one string from the right end of another string.
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action string trimright

To trim the characters one string from the right end of another string when an Embedded Event Manager (EEM) applet is triggered, use the **action string trimright** command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **string trimright** *string1* [*string2*]

no action *label* **string trimright**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string1	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
string2	(Optional) Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.

Command Default

By default, there is no action to trim a string.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(22)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

Use the **action string trimright** command to trim a string from the right end of another string. This command trims the characters specified by *string2* from the right end of *string1*. By default, *string2* corresponds to white space.

Table 20 shows the built-in variable in which the result of the **action string trimright** command is stored.

Table 20 EEM Built-in Variables for action string trimright Command

Built-in Variable	Description
$_string_result	The result of the **action string trimright** command is stored in this variable.

Examples

The following example shows how to trim a string from the right side of another string:

```
Router(config)# event manager applet trimright
Router(config-applet)# action 1.0 set string "How are you? Hello"
```

```
Router(config-applet)# action 2.0 string trim "$string" " Hello"
Router(config-applet)# action 3.0 puts "$_string_result"
Router(config-applet)# end
Router# event manager run trimright
How are you?
Router#
```

Related Commands

Command	Description
action string trim	Trims a string.
action string trimleft	Trims the characters by one string from the left end of another string.
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action subtract

To subtract the value of a variable from another value when an Embedded Event Manager (EEM) applet is triggered, use the **action subtract** command in applet configuration mode. To undo the subtract action, use the **no** form of this command.

action *label* **subtract** {*variable-name* | *long-integer*} {*variable-name* | *long-integer*}

no action *label* **subtract**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
variable-name	String value that identifies the variable name.
long-integer	Long integer value by which another value gets subtracted.

Command Default

By default, there is no change in the value of variables configured within an EEM applet.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(22)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

You can use this action to subtract the value of a variable from the value of another variable. The result is stored in the variable named $_result. The value of the variable must be a long integer or else the action will fail.

Examples

The following example shows how to configure an EEM applet to subtract the value of a variable from another value:

```
Router(config)#event manager applet one
Router(config-applet)#action 1.0 set $var1 20
Router(config-applet)#action 1.0 set $var2 10
Router(config-applet)#action 1.0 subtract $var1 $var2
Router(config-applet)#
```

Related Commands

Command	Description
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action syslog

To specify the action of writing a message to syslog when an Embedded Event Manager (EEM) applet is triggered, use the **action syslog** command in applet configuration mode. To remove the syslog message event criteria, use the **no** form of this command.

action *label* **syslog** [**priority** *priority-level*] **msg** *msg-text* **facility** *string*

no action *label* **syslog**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
priority	(Optional) Specifies the priority level of the syslog messages. If this keyword is selected, the *priority-level* argument must be defined. If this keyword is not selected, all syslog messages are set at the informational priority level.
priority-level	(Optional) Number or name of the desired priority level at which syslog messages are set. Priority levels are as follows (enter the number or the keyword): • {**0** | **emergencies**}—System is unusable. • {**1** | **alerts**}—Immediate action is needed. • {**2** | **critical**}—Critical conditions. • {**3** | **errors**}—Error conditions. • {**4** | **warnings**}—Warning conditions. • {**5** | **notifications**}—Normal but significant conditions. • {**6** | **informational**}—Informational messages. This is the default. • {**7** | **debugging**}—Debugging messages.
msg	Specifies the message to be logged.
msg-text	Character text, an environment variable, or a combination of the two. If the string contains embedded blanks, enclose it in double quotation marks. **Note** Messages written to syslog from an EEM applet are not screened for EEM syslog events, which may lead to recursive EEM syslog events. Messages sent from an EEM applet include the applet name for identification.
facility *string*	Specifies the facility.

Command Default

No messages are written to syslog.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.0(26)S	This command was introduced.
12.3(4)T	This command was integrated into Cisco IOS Release 12.3(4)T.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(28)SB	This command was integrated into Cisco IOS Release 12.2(28)SB.
12.2(18)SXF4	This command was integrated into Cisco IOS Release 12.2(18)SXF4 to support Software Modularity images only.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(18)SXF5	This command was integrated into Cisco IOS Release 12.2(18)SXF5.
15.0(1)M	This command was modified. The **facility** keyword and the *string* argument were added.

Examples

The following example shows how to specify a message to be sent to syslog when the memory-fail applet is triggered:

```
Router(config)# event manager applet memory-fail
Router(config-applet)# event snmp oid 1.3.6.1.4.1.9.9.48.1.1.1.6.1 get-type exact entry-op
lt entry-val 5120000 poll-interval 10
Router(config-applet)# action 4.0 syslog msg "Memory exhausted; current available memory
is $_snmp_oid_val bytes"
```

The following example shows how to generate a syslog message when it detects a syslog message pattern "console", using priority level 3 (errors) and facility EEM-FAC:

```
Router(config)# event manager applet test
Router(config-applet)# event syslog pattern "console"
Router(config-applet)# action 4.0 syslog priority errors facility EEM-FAC msg "TEST MSG"
```

Command	Description
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

action track read

To specify the action of reading the state of a tracked object when an Embedded Event Manager (EEM) applet is triggered, use the **action track read** command in applet configuration mode. To remove the **action track read** command from the configuration, use the **no** form of this command.

action *label* **track read** *object-number*

no action *label* **track read** *object-number*

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
object-number	Tracked object number in the range from 1 to 500, inclusive. The number is defined using the **track stub** command.

Command Default

The state of a tracked object is not read.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(2)T	This command was introduced.
12.2(31)SB3	This command was integrated into Cisco IOS Release 12.2(31)SB3.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2(33)SXI	This command was integrated into Cisco IOS Release 12.2(33)SXI.

Usage Guidelines

This command generates the following result variable:

- _track_state—State of the specified tracked object. The text string returned is either up or down. If the state is up, it means that the object exists and is in an up state. If the state is down, it means that the object either does not exist or is in a down state.

This command is used to help track objects using EEM. Each tracked object is identified by a unique number that is specified on the tracking command-line interface (CLI). Client processes such as EEM use this number to track a specific object. The tracking process periodically polls the tracked objects and notes any change of value. The changes in the tracked object are communicated to interested client processes, either immediately or after a specified delay. The object values are reported as either up or down. The enhanced object tracking event detector publishes an EEM event when the tracked object changes.

Examples

The following example shows how to specify event criteria based on a tracked object:

```
event manager applet track-ten
 event track 10 state any
 action 1.0 track set 10 state up
 action 2.0 track read 10
```

Related Commands

Command	Description
action track set	Specifies the action of setting the state of a tracked object when an EEM applet is triggered.
event manager applet	Registers an event applet with the Embedded Event Manager and enters applet configuration mode.
show track	Displays tracking information.
track stub	Creates a stub object to be tracked.

action track set

To specify the action of setting the state of a tracked object when an Embedded Event Manager (EEM) applet is triggered, use the **action track set** command in applet configuration mode. To remove the **action track set** command from the configuration, use the **no** form of this command.

action *label* **track set** *object-number* **state** {**up** | **down**}

no action *label* **track set** *object-number* **state** {**up** | **down**}

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
object-number	Tracked object number in the range from 1 to 500, inclusive. The number is defined using the **track stub** command.
state	Specifies the state to which the tracked object will be set.
up	Specifies that the state of the tracked object will be set to up.
down	Specifies that the state of the tracked object will be set to down.

Command Default

The state of a tracked object is not set.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(2)T	This command was introduced.
12.2(31)SB3	This command was integrated into Cisco IOS Release 12.2(31)SB3.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.
12.2(33)SXI	This command was integrated into Cisco IOS Release 12.2(33)SXI.

Usage Guidelines

This command generates the following result variable:

- _track_state—State of the specified tracked object. The text string returned is either up or down. If the state is up, it means that the object exists and is in an up state. If the state is down, it means that the object either does not exist or is in a down state.

This command is used to help track objects using EEM. Each tracked object is identified by a unique number that is specified on the tracking command-line interface (CLI). Client processes such as EEM use this number to track a specific object. The tracking process periodically polls the tracked objects and notes any change of value. The changes in the tracked object are communicated to interested client processes, either immediately or after a specified delay. The object values are reported as either up or down. The enhanced object tracking event detector publishes an EEM event when the tracked object changes.

Examples

The following example shows how to specify event criteria based on a tracked object:

```
event manager applet track-ten
 event track 10 state any
 action 1.0 track set 10 state up
 action 2.0 track read 10
```

Related Commands

Command	Description
action track read	Specifies the action of reading the state of a tracked object when an EEM applet is triggered.
event manager applet	Registers an event applet with the Embedded Event Manager and enters applet configuration mode.
show track	Displays tracking information.
track stub	Creates a stub object to be tracked.

action while

To identify the beginning of a loop of a conditional block when an Embedded Event Manager (EEM) applet is triggered, use the **action while** command in applet configuration mode. To disable this function, use the **no** form of this command.

action *label* **while** *string-op1 operator string-op2*

no action *label* **while**

Syntax Description

label	Unique identifier that can be any string value. Actions are sorted and run in ascending alphanumeric key sequence using the label as the sort key. If the string contains embedded blanks, enclose it in double quotation marks.
string-op1	Specifies the first operand.
operator	Value used with the *string-op1* and s*tring-op2* operands that determines how the current counter value is compared to the entry value or the exit value. Valid values are: • **gt**—Greater than. • **ge**—Greater than or equal to. • **eq**—Equal to. • **ne**—Not equal to. • **lt**—Less than. • **le**—Less than or equal to.
string-op2	The second operand.

Command Default

No conditional block is specified.

Command Modes

Applet configuration (config-applet)

Command History

Release	Modification
12.4(22)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

Use the **action while** command to identify the beginning of a loop conditional block. If $_variable is found within a string, it will be substituted before the expression is tested.

Examples

The following example shows how to identify the beginning of a loop of a conditional block when an EEM applet is triggered:

```
Router(config-applet)# action 1 set _i 2
```

```
Router(config-applet)# action 2 while $_i lt 10
Router(config-applet)# action 3 action syslog msg "i is $_i"
Router(config-applet)# action 4 end
```

Related Commands

Command	Description
action else	Identifies the beginning of an else block in the if/else conditional block.
action elseif	Identifies the beginning of the if/else conditional block.
action if	Identifies the beginning of an if conditional block.
event manager applet	Registers an event applet with the EEM and enters applet configuration mode.

add (bulkstat object)

To add a MIB object to a bulk statistics object list, use the **add** command in Bulk Statistics Object List configuration mode. To remove a MIB object from an SNMP bulk statistics object list, use the **no** form of this command.

add {*object-name* | *oid*}

no add {*object-name* | *oid*}

Syntax Description

object-name	Name of the MIB object to add to the list. Only object names from the Interfaces MIB (IF-MIB.my), Cisco Committed Access Rate MIB (CISCO-CAR-MIB.my) and the MPLS Traffic Engineering MIB (MPLS-TE-MIB.my) may be used.
oid	Object ID (OID) of the MIB object to add to the list.Only OIDs from the Interfaces MIB (IF-MIB.my), Cisco Committed Access Rate MIB (CISCO-CAR-MIB.my) and the MPLS Traffic Engineering MIB (MPLS-TE-MIB.my) may be used.

Command Default

No MIB objects are listed in the bulk statistics object list.

Command Modes

Bulk Statistics Object List configuration (config-bulk-objects)

Command History

Release	Modification
12.0(24)S	This command was introduced.
12.3(2)T	This command was integrated into Cisco IOS Release 12.3(2)T.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.
Cisco IOS XE Release 2.1	This command was integrated into Cisco IOS Release XE 2.1.

Usage Guidelines

All the objects in an object list have to be indexed by the same MIB index, but the objects need not belong to the same MIB table. For example, it is possible to group ifInoctets and an Ether MIB object in the same schema because the containing tables are indexed by the ifIndex (in the IF-MIB).

Object names are available in the relevant MIB modules. For example, the input byte count of an interface is defined in the Interfaces Group MIB (IF-MIB.my) as ifInoctets. Complete MIB modules can be downloaded from Cisco.com at http://www.cisco.com/public/sw-center/netmgmt/cmtk/mibs.shtml.

Examples

In the following example, two bulk statistics object lists are configured: one for IF-MIB objects and one for CISCO-CAR-MIB objects. Because the IF-MIB objects and the CISCO-CAR-MIB objects do not have the same index, they must be defined in separate object lists.

```
Router(config)# snmp mib bulkstat object-list if-Objects
Router(config-bulk-objects)# add ifInoctets
Router(config-bulk-objects)# add ifOutoctets
Router(config-bulk-objects)# add ifInUcastPkts
Router(config-bulk-objects)# add ifInDiscards
Router(config-bulk-objects)# exit
Router(config)# snmp mib bulkstat object-list CAR-Objects
Router(config-bulk-objects)# add CcarStatSwitchedPkts
Router(config-bulk-objects)# add ccarStatSwitchedBytes
Router(config-bulk-objects)# add CcarStatFilteredBytes
Router(config-bulk-objects)# exit
Router(config)#
```

Related Commands

Command	Description
snmp mib bulkstat object-list	Names a bulk statistics object list and enters Bulk Statistics Object List configuration mode.

alias (boomerang)

To configure an alias name for a specified domain, use the **alias** command in boomerang configuration mode. To remove this command from the configuration file and restore the system to its default condition with respect to this command, use the **no** form of this command.

alias *alias-name*

no alias *alias-name*

Syntax Description

alias-name	Alias name for a specified domain.

Command Default

No domain name alias is configured.

Command Modes

Boomerang configuration

Command History

Release	Modification
12.2(8)T	This command was introduced.

Usage Guidelines

The **alias** command can be used only on a Director Response Protocol (DRP) agent. The boomerang client is the DRP agent.

Use the **alias** command to specify one or more alias names for an existing domain. Because the boomerang client maintains separate counters for requests received for each domain name (alias or otherwise), use the **show ip drp boomerang** command to view these counters for a specified domain name and each of its aliases.

Examples

In the following example, the domain name alias is configured for www.boom1.com. The new alias for www.boom1.com is www.boom2.com:

```
Router(config)# ip drp domain www.boom1.com
Router(config-boomerang)# alias www.boom2.com

Router# show running-config
.
.
.
ip drp domain www.boom1.com
alias www.boom2.com
```

Related Commands

Command	Description
ip drp domain	Adds a new domain to the DistributedDirector client or configures an existing domain and puts the client in boomerang configuration mode.
server (boomerang)	Configures the server address for a specified boomerang domain.
show ip drp	Displays DRP statistics on DistributedDirector or a DRP server agent.
show ip drp boomerang	Displays boomerang information on the DRP agent.
ttl dns	Configures the number of seconds for which an answer received from the boomerang client will be cached by the DNS client.
ttl ip	Configures the IP TTL value for the boomerang response packets sent from the boomerang client to the DNS client in number of hops.

announce config

To specify that an unsolicited configuration inventory is sent out by the CNS inventory agent at bootup, use the **announce config** command in CNS inventory configuration mode. To disable the sending of the configuration inventory, use the **no** form of this command.

announce config

no announce config

Syntax Description

This command has no arguments or keywords.

Defaults

Disabled

Command Modes

CNS inventory configuration

Command History

Release	Modification
12.3(1)	This command was introduced.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use this command to limit inventory requests by the CNS inventory agent. When configured, the routing device details will be announced on the CNS event bus, but the routing device will not respond to any queries from the CNS event bus.

Examples

The following example shows how to configure the CNS inventory agent to send out an unsolicited configuration inventory one time only at bootup:

```
Router(config)# cns inventory
Router(cns_inv)# announce config
```

Related Commands

Command	Description
cns inventory	Enables the CNS inventory agent and enters CNS inventory configuration mode.

attribute (EEM)

To specify a complex event for an Embedded Event Manager (EEM) applet, use the **attribute** command in trigger applet configuration mode. To remove the attributes, use the **no** form of this command.

attribute tag *event-tag* [**occurs** *occurs-value*]

no attribute tag *event-tag* [**occurs** *occurs-value*]

Syntax Description

tag	Specifies a tag using the *event-tag* argument that can be used with the **attribute** command to associate an event.
event-tag	String that identifies the tag.
occurs	(Optional) Specifies the number of occurrences before an EEM event is triggered. If not specified, an EEM event is triggered on the first occurrence.
occurs-value	(Optional) Number in the range from 1 to 4294967295.

Command Default

No complex events are specified for an EEM applet.

Command Modes

Trigger applet configuration (config-applet-trigger)

Command History

Release	Modification
12.4(20)T	This command was introduced.
12.2(33)SRE	This command was integrated into Cisco IOS Release 12.2(33)SRE.

Usage Guidelines

In the trigger applet configuration mode, up to eight attribute statements can be specified to build a complex event. If no attribute statements are specified, the options in the trigger statement apply to the first event defined in the applet.

Examples

The following example shows how to use the **attribute** command to specify a complex events for an EEM applet. In this example, the applet is run when the **show bgp all** command and any syslog message that contains the string “COUNT” occurs within a period of 60 seconds.

```
Router(config)# event manager applet delay_50
Router(config-applet)# event tag 1.0 cli pattern "show bgp all" sync yes occurs 32 period
60 maxrun 60
Router(config-applet)# event tag 2.0 syslog pattern "COUNT"
Router(config-applet)# trigger occurs 1 delay 50
Router(config-applet-trigger)# correlate event 1.0 or event 2.0
Router(config-applet-trigger)# attribute tag 1.0 occurs 1
Router(config-applet-trigger)# attribute tag 2.0 occurs 1
Router(config-applet-trigger)# action 1.0 cli command "show memory"
```

```
Router(config-applet)# action 2.0 cli command "enable"
Router(config-applet)# action 3.0 cli command "config terminal"
Router(config-applet)# action 4.0 cli command " ip route 192.0.2.0 255.255.255.224
192.0.2.12"
Router(config-applet)# action 91.0 cli command "exit"
Router(config-applet)# action 99.0 cli command "show ip route | incl 192.0.2.5"
```

Related Commands

Command	Description
correlate	Builds a single complex event.
trigger (EEM)	Enters trigger applet configuration mode and specifies the multiple event configuration statements for an EEM applet.

backup excluded

To set the time that the Web Services Management Agent (WSMA) profile must wait after a connection is lost before attempting to connect to the backup transport configuration, use the **backup excluded** command in WSMA initiator configuration mode. To disable the configured backup excluded time, use the **no** form of this command.

backup excluded *time*

no backup excluded

Syntax Description

time	The time, in seconds, that the WSMA profile waits before attempting to connect to the backup transport configuration. The backup excluded value must be within the range 1 to 2,000,000 seconds. The default value is 0 seconds.

Command Default

The time is set to 0 seconds.

Command Modes

WSMA initiator configuration (config-wsma-init)

Command History

Release	Modification
15.1(1)T	This command was introduced.

Usage Guidelines

If the primary transport connection is lost and a backup transport configuration has been set up, the WSMA profile will connect to the backup transport connection.

This setting is ignored if the WSMA profile has no primary transport configuration.

Examples

The following example shows how to configure the backup excluded time for a WSMA initiator profile:

```
Router(config)# wsma profile initiator prof1
Router(config-wsma-init)# backup excluded 60
Router(config-wsma-init)#
```

Related Commands

Command	Description
backup hold	Sets the time that the WSMA profile remains connected to the backup transport configuration.
encap	Configures an encapsulation for a WSMA profile.
idle-timeout	Sets a time for the WSMA profile to keep the session alive in the absence of any data traffic.
keepalive	Enables keepalive messages and configures interval and retry values for a WSMA profile.

Command	Description
max-message	Sets the maximum size limit for incoming messages.
reconnect	Specifies the time for the WSMA initiator profile to wait before attempting to reconnect a session.
stealth	Disables WSMA from sending SOAP faults.
transport	Defines a transport configuration for a WSMA profile.
wsma profile initiator	Configures and enables a WSMA initiator profile.
wsse	Enables the WSSE for a WSMA profile.

backup hold

To set the time that the Web Services Management Agent (WSMA) profile remains connected to the backup transport configuration, use the **backup hold** command in WSMA initiator configuration mode. To disable the backup hold time, use the **no** form of this command.

backup hold *time*

no backup hold

Syntax Description

time	The time, in minutes, to remain connected to the backup transport connection. By default, the connection is set to never disconnect. The backup hold value must be within the range 1 to 35,000 minutes.

Command Default

The backup hold time is set to infinity.

Command Modes

WSMA initiator configuration (config-wsma-init)

Command History

Release	Modification
15.1(1)T	This command was introduced.

Usage Guidelines

If both primary and backup transport connections are configured, the hold time indicates how long the WSMA profile remains connected to the backup transport before the connection to the backup is closed and a new connection to the primary transport is attempted.

If the primary transport connection is lost and a backup transport configuration has been set up, the WSMA profile will connect to the backup transport connection.

This command can be used when you need to disconnect from the primary transport for a specific time. For example, if you want to perform maintenance on the primary transport and want to automatically switch back from the backup to the primary transport after a known period.

This setting is ignored if the WSMA profile has no primary transport configuration.

Examples

The following example shows how to configure the backup hold time for a WSMA initiator profile:

```
Router(config)# wsma profile initiator prof1
Router(config-wsma-init)# backup hold 120
Router(config-wsma-init)#
```

Related Commands

Command	Description
backup excluded	Sets the time that the WSMA profile must wait after a connection is lost before attempting to connect to the backup transport configuration.
encap	Configures an encapsulation for a WSMA profile.
idle-timeout	Sets a time for the WSMA profile to keep the session alive in the absence of any data traffic.
keepalive	Enables keepalive messages and configures interval and retry values for a WSMA profile.
max-message	Sets the maximum size limit for incoming messages.
reconnect	Specifies the time for the WSMA initiator profile to wait before attempting to reconnect a session.
stealth	Disables WSMA from sending SOAP faults.
transport	Defines a transport configuration for a WSMA profile.
wsma profile initiator	Configures and enables a WSMA initiator profile.
wsse	Enables the WSSE for a WSMA profile.

bingd device

To reply to a Blocks Extensible Exchange Protocol (BEEP) ping daemon equivalents on the listening port with the help of a device, use the **bingd device** command in privileged EXEC mode.

bingd device {*listening-port-number* | **ipv6** *listening-port-number* [**privacy cipher** *sum* **trustpoint** *trustpoint-name* **sasl profile** *profile-name* | **sasl profile** *profile-name* | **spawn** *concurrent-number* | **syslog** [**privacy cipher** *sum* **trustpoint** *trustpoint-name* **sasl profile** *profile-name* | **sasl profile** *profile-name* | **spawn** *concurrent-number*]]}

Syntax Description

listening-port-number	Listening port number. The range is from 1 to 65535.
ipv6 *listening-port-number*	Specifies an IPv6 address with a listening port number. The range is from 1 to 65535.
privacy	(Optional) Uses Transport Layer Security (TLS) for encryption.
cipher *sum*	(Optional) The encryption algorithm to be used. The range is from 32 to 224 and is the result of the sum of the following numbers: • 32 (TLS_RSA_WITH_NULL_SHA) • 64 (TLS_RSA_WITH_RC4_128_MD5) • 128 (TLS_RSA_WITH_AES_128_CBC_SHA)
trustpoint *trustpoint-name*	(Optional) Specifies the trustpoint name to be used.
sasl	(Optional) Specifies Simple Authentication and Security Layer (SASL) details.
profile *profile-name*	(Optional) Specifies the SASL profile to be used.
spawn *concurrent-number*	(Optional) Allows multiple BEEP ping daemon equivalents to run concurrently. The range is from 2 to 100.
syslog	(Optional) Loads the syslog raw profile.

Command Default

There is no device that is activated by default to reply to a BEEP ping daemon equivalents on the listening port.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.0(1)M	This command was introduced in a release earlier than Cisco IOS Release 15.0(1)M.

Examples

The following example demonstrates how you can reply to a BEEP ping daemon equivalents with the help of a device with an IPV6 address and a listening port of 5, a privacy encryption of 32, trustpoint XYZ, and SASL profile DEF:

```
Router# bingd device ipv6 5 syslog privacy cipher 32 trustpoint XYZ sasl profile DEF
```

Related Commands

Command	Description
bingd template	Listens to a BEEP ping daemon equivalents with the help of a template.

bingd template

To listen to Blocks Extensible Exchange Protocol (BEEP) ping daemon equivalents with the help of a template, use the **bingd template** command in privileged EXEC mode.

bingd template *template-name* {**privacy cipher** *sum* **trustpoint** *trustpoint-name* **sasl profile** *profile-name* | **sasl** *profile-name* | **spawn** *concurrent-number*}

Syntax Description

privacy	Uses Transport Layer Security (TLS) for encryption.
cipher *sum*	The encryption algorithm to be used. The range is from 32 to 224 and is the result of the sum of the following numbers: • 32 (TLS_RSA_WITH_NULL_SHA), • 64 (TLS_RSA_WITH_RC4_128_MD5) • 128 (TLS_RSA_WITH_AES_128_CBC_SHA)
trustpoint *trustpoint-name*	Specifies the trustpoint name to be used.
sasl	Specifies Simple Authentication and Security Layer (SASL) details.
profile *profile-name*	Specifies the SASL profile to be used.
spawn *concurrent-number*	Allows multiple BEEP ping daemon equivalents to run concurrently. The range is from 2 to 100.

Command Default

There is no template that is activated by default.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.0(1)M	This command was introduced in a release earlier than Cisco IOS Release 15.0(1)M.

Examples

The following example demonstrates how you can listen to BEEP ping daemon equivalent with the help of a template ABC with a privacy encryption of 32, trustpoint XYZ, and SASL profile DEF:

```
Router# bingd template ABC privacy cipher 32 trustpoint XYZ sasl profile DEF
```

Related Commands

Command	Description
bingd stop	Stops all the running or active BEEP ping daemon equivalent.

bingd stop

To stop all the running or active Blocks Extensible Exchange Protocol (BEEP) ping daemon equivalent, use the **bingd stop** command in privileged EXEC mode.

bingd stop [*port-number*]

Syntax Description

port-number	(Optional) Listening port number. The range is from 1 to 65535.

Command Default

The BEEP ping daemon equivalent on the listening port is active.

Command Modes

Privileged EXEC (#)

Command History

Release	Modification
15.0(1)M	This command was introduced in a release earlier than Cisco IOS Release 15.0(1)M.

Examples

In the following example, the **bingd stop** command stops all the active or running BEEP ping daemon equivalents:

```
Router# bingd stop
```

Related Commands

Command	Description
bingd template	Listens to BEEP ping daemon equivalents with the help of a template.

buffer-length

To specify the maximum length of the data stream to be forwarded, use the **buffer-length** command in line configuration mode. To restore the default setting, use the **no** form of this command.

buffer-length *bytes*

no buffer-length

Syntax Description

bytes	The length of the buffer in bytes. Valid values range from 1 to 1536. The default buffer length is 1536 bytes.

Defaults

1536 bytes

Command Modes

Line configuration (config-line)

Command History

Release	Modification
12.1	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.
15.1(1)T	This command was modified. The minimum allowed length was changed to 1 byte.

Usage Guidelines

The **buffer-length** command configures the length of the forwarded data stream. The higher the value used for the *byte* argument is, the longer the delay between data transmissions will be. Configuring a smaller buffer-length can prevent connections from timing out inappropriately.

A connection timeout with a high buffer-length value is a very rare occurrence and it depends on the CPU load. Configuring a lower buffer-length value can prevent connection timeouts. A lower buffer-length value is needed only when data transmission is time critical.

Caution A lower buffer-length value should be used with caution. If all the Network Management (NM) and WAN interface card (WIC) slots in the router are filled with async cards, and each of the tty async lines is configured with a buffer length of 1 byte, then the load on the CPU can be increased and the CPU can stall.

Examples

The following example configures a buffer length of 1 byte:

```
Router(config)# line 1
Router(config-line)# buffer-length 1
```

buffer public

To enter buffer owner configuration mode to set thresholds for buffer usage, use the **buffer public** command in resource policy node configuration mode.

buffer public

Syntax Description

This command has no arguments or keywords.

Command Modes

Resource policy node configuration (config-policy-node)

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.

Usage Guidelines

This command allows you to enter buffer owner configuration mode to set rising and falling values for critical, major, and minor thresholds for buffer usage.

Examples

The following example shows how to enter buffer owner configuration mode to set thresholds for buffer usage:

```
Router(config)# resource policy
Router(config-erm)# policy policy1 type iosprocess
Router(config-erm-policy)# system
Router(config-policy-node)# buffer public
Router(config-owner-buffer)#
```

Related Commands

Command	Description
critical rising	Sets the critical level threshold values for the buffer, CPU, and memory ROs.
major rising	Sets the major level threshold values for the buffer, CPU, and memory ROs.
minor rising	Sets the minor level threshold values for the buffer, CPU, and memory ROs.
policy (ERM)	Configures an ERM resource policy.
resource policy	Enters ERM configuration mode.
show buffer leak	Displays the buffer details.
show resource all	Displays all the resource details.
slot (ERM policy)	Configures line cards.
system (ERM policy)	Configures system level ROs.

buffer-size (bulkstat)

To configure a maximum buffer size for the transfer of bulk statistics files, use the **buffer-size** command in Bulk Statistics Transfer configuration mode. To remove a previously configured buffer size from the configuration, use the **no** form of this command.

buffer-size *bytes*

no buffer-size

Syntax Description

bytes	Size of the bulk statistics transfer buffer, in bytes. The valid range is from 1024 to 2147483647. The default is 2048.

Command Default

The default bulk statistics transfer buffer is 2048 bytes.

Command Modes

Bulk Statistics Transfer configuration (config-bulk-tr)

Command History

Release	Modification
12.0(24)S	This command was introduced.
12.3(2)T	This command was integrated into Cisco IOS Release 12.3(2)T.
12.2(25)S	This command was integrated into Cisco IOS Release 12.2(25)S.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(33)SXH	This command was integrated into Cisco IOS Release 12.2(33)SXH.
12.2(33)SB	This command was integrated into Cisco IOS Release 12.2(33)SB.
Cisco IOS XE Release 2.1	This command was integrated into Cisco IOS Release XE 2.1.

Usage Guidelines

A configured buffer size limit is available primarily as a safety feature. Normal bulk statistics files should not generally meet or exceed the default value while being transferred.

Examples

In the following example, the bulk statistics transfer buffer size is set to 3072 bytes:

```
Router(config)# snmp mib bulkstat transfer bulkstat1
Router(config-bulk-tr)# schema ATM2/0-IFMIB
Router(config-bulk-tr)# url primary ftp://user:pswrd@host/folder/bulkstat1
Router(config-bulk-tr)# buffer-size 3072
Router(config-bulk-tr)# enable
Router(config-bulk-tr)# exit
Router(config)#
```

Related Commands

Command	Description
snmp mib bulkstat transfer	Identifies the transfer configuration with a name and enters Bulk Statistics Transfer configuration mode.

buffers

To make adjustments to initial public buffer pool settings and to the limits at which temporary buffers are created and destroyed, use the **buffers** command in global configuration mode. To return the buffer pool settings to their default sizes, use the **no** form of this command.

buffers {{**header** | **fastswitching** | *interface number* | **small** | **middle** | **big** | **verybig** | **large** | **huge** {**initial** | **max-free** | **min-free** | **permanent**} *buffers*} | **particle-clone** *particle-clones* | **element** {**minimum** | **permanent**} *elements*}

no buffers {{**header** | **fastswitching** | *interface number* | **small** | **middle** | **big** | **verybig** | **large** | **huge** {**initial** | **max-free** | **min-free** | **permanent**} *buffers*} | **particle-clone** *particle-clones* | **element** {**minimum** | **permanent**} *elements*}

Syntax Description

header	Number of particles in the header particle pool. The range is from 256 to 65535. The defaults are min:256, max:1024, and cache:256.
fastswitching	Number of particles in the fastswitching particle pool. The range is from 512 to 65535. The defaults are min:0, max:512, and cache:512.
type number	Interface *type* and *number* of the interface buffer pool. The *type* value cannot be **fddi**.
small	Buffer size of this public buffer pool is 104 bytes.
middle	Buffer size of this public buffer pool is 600 bytes.
big	Buffer size of this public buffer pool is 1524 bytes.
verybig	Buffer size of this public buffer pool is 4520 bytes.
large	Buffer size of this public buffer pool is 5024 bytes.
huge	Public buffer pool can be configured with the **buffers huge size** command. Default buffer size of this public buffer pool, in bytes, is 18024.
initial	Number of additional temporary buffers that are to be allocated when the system is reloaded. This keyword can be used to ensure that the system has necessary buffers immediately after reloading in a high-traffic environment.
max-free	Maximum number of free or unallocated buffers in a buffer pool. The maximum number of small buffers that can be constructed in the pool is 20480.
min-free	Minimum number of free or unallocated buffers in a buffer pool.
permanent	Number of permanent buffers that the system tries to create and keep. Permanent buffers are normally not trimmed by the system.
buffers	Number of buffers to be allocated. The range is 0 to 65536.
particle-clone *particle-clone*	Number of particle clones to grow. The range is from 1024 to 65535. The default is 1024.
element	Buffer elements. The required keywords for the **element** keyword are as follows: • **permanent**—Permanent buffer elements. • **minimum**—Minimum buffer elements.
elements	Number of buffer elements. For permanent buffer elements. The range is from 500 to 65535. The default is 500. For minimum buffer elements. The range is from 500 to 65535.

Defaults

Buffers are set at default sizes that vary by hardware configuration.

Command Modes

Global configuration

Command History

Release	Modification
10.0	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2(31)SB	This command was integrated into Cisco IOS Release 12.2(31)SB.
12.4(10)	The **minimum** keyword was added to set the minimum number of buffer elements. The **particle-clone** keyword was added to set the number of particle clones in the buffer pool. The **header** keyword was added to set the number of particles in the header particle pool. The **fastswitching** keyword was added to set the number of particles in the fastswitching particle pool.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

The default number of buffers in a pool is determined by the hardware configuration and can be displayed with the **show buffers** command in user EXEC mode. Generally, buffer settings do not need to be adjusted. Consult with technical support personnel before making any changes.

Note Improper buffer settings can adversely impact system performance.

You cannot configure FDDI buffers.

Use the **element** keyword with the **permanent** *elements* keyword-argument combination to increase the number of permanent buffer elements to prevent packet loss. For example, in a multicasting environment, a higher number of buffer elements may be needed to accommodate bursts of traffic.

Use the **element** keyword with the **minimum** *elements* keyword-argument combination to set the minimum number of buffer elements.

Note It is preferable to use the **element** keyword with the **permanent** *elements* keyword-argument combination during system initialization because a higher number of permanent buffer elements will then be ready for use in case a burst of traffic occurs.

Use the **show buffers** command to display statistics such as the following:

- Free list (the total number of unallocated buffer elements)
- Max allowed (the maximum number of buffer elements that are available for allocation)
- Hits (the count of successful attempts to allocate a buffer when needed)

- Misses (the count of buffer allocation attempts that resulted in growing the buffer pool to allocate a buffer)
- Created (the count of new buffers created to satisfy buffer allocation attempts when the available buffers in the pool have already been allocated)

Note If the requested number of permanent buffer elements is fewer than the current number of permanent buffer elements, the configuration will not take effect until the next reload. Resetting the number of permanent buffer elements to the default value using the **no** form of this command will not take effect until the next reload.

Cisco 10000 Series Router

Table 21 lists the buffer sizes to configure if your network uses a RADIUS server for authentication.

Table 21 Buffer Sizes for RADIUS Authentication

Buffer	Size (in Bytes)
Small	15000
Middle	12000
Big	8000

Examples

Examples of Public Buffer Pool Tuning

The following example shows how to keep at least 50 small buffers free in the system:

```
Router(config)# buffers small min-free 50
```

The following example shows how to increase the permanent buffer pool allocation for big buffers to 200:

```
Router(config)# buffers big permanent 200
```

Example of Interface Buffer Pool Tuning

A general guideline is to display buffers with the **show buffers** command and to increase the buffer pool that is depleted.

The following example shows how to increase the permanent Ethernet interface 0 buffer pool on a Cisco 4000 router to 96 when the Ethernet 0 buffer pool is depleted:

```
Router(config)# buffers ethernet 0 permanent 96
```

Examples of Buffer Element Tuning

The following example shows how to configure the number of permanent buffer elements to 6,000:

```
Router(config)# buffers element permanent 6000
```

The following example shows how to configure the number of minimum buffer elements to 6,000:

```
Router(config)# buffers element minimum 6000
```

Related Commands

Command	Description
load-interval	Changes the length of time for which data is used to compute load statistics.
show buffers	Displays statistics for the buffer pools on the network server.

buffers huge size

To dynamically resize all huge buffers to the value you specify, use the **buffers huge size** command in global configuration mode. To restore the default buffer values, use the **no** form of this command.

buffers huge size *number-of-bytes*

no buffers huge size *number-of-bytes*

Syntax Description

number-of-bytes	Huge buffer size (in bytes). Valid range is from 18024 to 100000 bytes.

Defaults

18,024 bytes

Command Modes

Global configuration

Command History

Release	Modification
10.0	This command was introduced.
12.2(33)SRA	This command was integrated into Cisco IOS Release 12.2(33)SRA.
12.2SX	This command is supported in the Cisco IOS Release 12.2SX train. Support in a specific 12.2SX release of this train depends on your feature set, platform, and platform hardware.

Usage Guidelines

Use this command only after consulting with technical support personnel. The buffer size cannot be lowered below the default.

Note Improper buffer settings can adversely impact system performance.

Examples

The following example resizes huge buffers to 20,000 bytes:

```
Router(config)# buffers huge size 20000
```

Related Commands

Command	Description
buffers	Adjusts the initial buffer pool settings and the limits at which temporary buffers are created and destroyed.
show buffers	Displays statistics for the buffer pools on the network server.

buffers tune automatic

To enable automatic tuning of buffers, use the **buffers tune automatic** command in global configuration mode. To disable automatic tuning of buffers, use the **no** form of this command.

buffers tune automatic

no buffers tune automatic

Syntax Description

This command has no arguments or keywords.

Command Default

Disabled

Command Modes

Global configuration

Command History

Release	Modification
12.3(14)T	This command was introduced.
12.2(33)SRB	This command was integrated into Cisco IOS Release 12.2(33)SRB.

Usage Guidelines

This command enables automatic tuning of buffers. Even when the command is not enabled, the parameters are computed. When you enable the command later, the buffer parameters change to the computed values.

Examples

The following example shows how to enable automatic tuning of buffers:

```
Router(config)# buffers tune automatic
```

Related Commands

Command	Description
show buffers tune	Displays the automatic buffer tune details.