



## action string through D Commands

---

- [action string through D Commands](#), page 1

## action string through D Commands

## 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

<i>label</i>	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.
<b>nocase</b>	(Optional) Specifies case insensitive comparison.
<b>length</b>	(Optional) Limits the comparison to the first integer character.
<i>integer</i>	(Optional) Valid values for the length argument range from 1 to 4294967295.
<i>string1</i>	Sequence of characters.
<i>string2</i>	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 **strencp** command.

The table below shows the built-in variable in which the results of the **action string compare** command are stored.

**Table 1: EEM Built-in Variables for action string compare Command**

Built-in Variable	Description
\$_string_result	The result of the <b>action string compare</b> 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
<b>event manager applet</b>	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

<i>label</i>	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.
<b>nocase</b>	(Optional) Specifies case insensitive comparison.
<b>length</b>	(Optional) Specifies the length of the value to limit the comparison.
<i>integer</i>	(Optional) Valid values for the length argument range from 1 to 4294967295.
<i>string1</i>	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
<i>string2</i>	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.

The table below shows the built-in variable in which the results of the **action string equal** command are stored.

**Table 2: EEM Built-in Variables for action string equal Command**

Built-in Variable	Description
\$_string_result	The result of the <b>action string equal</b> 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
<b>event manager applet</b>	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

<i>label</i>	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.
<i>string1</i>	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
<i>string2</i>	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
<i>index-value</i>	(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.

The table below shows the built-in variable in which the results of the **action string first** command are stored.

**Table 3: EEM Built-in Variables for action string first Command**

Built-in Variable	Description
<code>\$_string_result</code>	The result of the <b>action string first</b> 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
Router#
```

## Related Commands

Command	Description
<b>action string last</b>	Returns the index on the last occurrence of <i>string1</i> within <i>string2</i> .
<b>event manager applet</b>	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

<i>label</i>	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.
<i>string</i>	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
<i>value</i>	(Optional) The index value. Number in the range from 0 to 4294967295. The count starts from 0.
<i>end</i>	(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.

The table below shows the built-in variable in which the **action string index** command stores the characters.



**Table 4: EEM Built-in Variables for action string index Command**

Built-in Variable	Description
\$_string_result	The <b>action string index</b> 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
Router#
```

## Related Commands

Command	Description
<b>event manager applet</b>	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 *string2* 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

<i>label</i>	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.
<i>string1</i>	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
<i>string2</i>	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
<i>index-value</i>	(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.

The table below shows the built-in variable in which the results of the **action string last** command are stored.

**Table 5: EEM Built-in Variables for action string last Command**

Built-in Variable	Description
\$_string_result	The result of the <b>action string last</b> 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
Router#
```

**Related Commands**

Command	Description
<b>action string first</b>	Returns the index on the first occurrence of <i>string1</i> within <i>string2</i> .
<b>event manager applet</b>	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

<i>label</i>	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.
<i>string</i>	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.

The table below shows the built-in variable in which the results of the **action string length** command are stored.

**Table 6: EEM Built-in Variables for action string length Command**

Built-in Variable	Description
\$_string_result	The result of the <b>action string length</b> 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
<b>event manager applet</b>	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

<i>label</i>	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.
<b>nocase</b>	(Optional) Specifies case insensitive comparison.
<i>string-pattern</i>	The pattern for case insensitive comparison.
<i>string</i>	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.

The table below shows the built-in variable in which the results of the **action string match** command is stored.

**Table 7: EEM Built-in Variables for action string match Command**

Built-in Variable	Description
\$_string_result	The result of the <b>action string match</b> 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
Router#
```

## Related Commands

Command	Description
<b>event manager applet</b>	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 a pplet 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

<i>label</i>	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.
<i>string</i>	Sequence of characters which can be up to 4294967295. If the string contains embedded blanks, enclose it in double quotation marks.
<i>start-index</i>	The starting index string value. The range is from 0 to 4294967295.
<i>end-index</i>	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.

The table below shows the built-in variable in which the result of the **action string range** command is stored.



**Table 8: EEM Built-in Variables for action string range Command**

Built-in Variable	Description
\$_string_result	The result of the <b>action string range</b> 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
<b>event manager applet</b>	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

<i>label</i>	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.
<i>string</i>	Sequence of characters, which can be up to 4294967295. If the string contains embedded blanks, enclose it in double quotation marks.
<i>start-index</i>	The starting index string value. The range is from 0 to 4294967295.
<i>end-index</i>	The ending index string value. The range is from 0 to 4294967295.
<i>new-string</i>	(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.

The table below shows the built-in variable in which the result of the **action string replace** command is stored.

**Table 9: EEM Built-in Variables for action string replace Command**

Built-in Variable	Description
<code>\$_string_result</code>	The result of the <b>action string replace</b> 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
<b>event manager applet</b>	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

<i>label</i>	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.
<i>string</i>	The sequence of characters that needs to be replaced. If the string contains embedded blanks, enclose it in double quotation marks.
<i>start-index</i>	(Optional) The starting index string value. The range is from 0 to 4294967295.
<i>end-index</i>	(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.

The table below shows the built-in variable in which the result of the **action string tolower** command is stored.

**Table 10: EEM Built-in Variables for action string tolower Command**

Built-in Variable	Description
\$_string_result	The result of the <b>action string tolower</b> 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
<b>action string toupper</b>	Stores a specific range of characters of a string in uppercase.
<b>event manager applet</b>	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

<i>label</i>	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.
<i>string</i>	Specifies the sequence of characters, that needs to be replaced. If the string contains embedded blanks, enclose it in double quotation marks.
<i>start-index</i>	(Optional) The starting index string value. The range is from 0 to 4294967295.
<i>end-index</i>	(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.

The table below shows the built-in variable in which the result of the **action string toupper** command is stored.

**Table 11: EEM Built-in Variables for action string toupper Command**

Built-in Variable	Description
\$_string_result	The result of the <b>action string toupper</b> 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
<b>action string tolower</b>	Stores a specific range of characters of a string in lowercase.
<b>event manager applet</b>	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

<i>label</i>	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.
<i>string1</i>	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
<i>string2</i>	(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.

The table below shows the built-in variable in which the result of the **action string trim** command is stored.



**Table 12: EEM Built-in Variables for action string trim Command**

Built-in Variable	Description
\$_string_result	The result of the <b>action string trim</b> 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?
Router#
```

**Related Commands**

Command	Description
<b>action string trimleft</b>	Trims the characters by one string from the left end of another string.
<b>action string trimright</b>	Trims the characters by one string from the right end of another string.
<b>event manager applet</b>	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

<i>label</i>	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.
<i>string1</i>	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
<i>string2</i>	(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.

The table below shows the built-in variable in which the result of the **action string trimleft** command is stored.

**Table 13: EEM Built-in Variables for action string trimleft Command**

Built-in Variable	Description
\$_string_result	The result of the <b>action string trimleft</b> 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
<b>action string trim</b>	Trims a string.
<b>action string trimright</b>	Trims the characters by one string from the right end of another string.
<b>event manager applet</b>	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

<i>label</i>	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.
<i>string1</i>	Sequence of characters. If the string contains embedded blanks, enclose it in double quotation marks.
<i>string2</i>	(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.

The table below shows the built-in variable in which the result of the **action string trimright** command is stored.

**Table 14: EEM Built-in Variables for action string trimright Command**

Built-in Variable	Description
\$_string_result	The result of the <b>action string trimright</b> 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
<b>action string trim</b>	Trims a string.
<b>action string trimleft</b>	Trims the characters by one string from the left end of another string.
<b>event manager applet</b>	Registers an event applet with the EEM and enters applet configuration mode.

## action subtract

To specify the action of subtracting 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

<i>label</i>	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.
<b>subtract</b>	Subtracts the value of a variable from the value of another variable.
<i>variable-name</i>	String value to be placed as the variable name.
<i>long-integer</i>	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, 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
<b>event manager applet</b>	Registers an event applet with the Embedded Event Manager 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

<i>label</i>	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.
<i>object-number</i>	Tracked object number in the range from 1 to 500, inclusive. The number is defined using the <b>track stub</b> 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
<b>action track set</b>	Specifies the action of setting the state of a tracked object when an EEM applet is triggered.
<b>event manager applet</b>	Registers an event applet with the Embedded Event Manager and enters applet configuration mode.
<b>show track</b>	Displays tracking information.
<b>track stub</b>	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

<i>label</i>	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.
<i>object-number</i>	Tracked object number in the range from 1 to 500, inclusive. The number is defined using the <b>track stub</b> command.
<b>state</b>	Specifies the state to which the tracked object will be set.
<b>up</b>	Specifies that the state of the tracked object will be set to up.
<b>down</b>	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
<b>action track read</b>	Specifies the action of reading the state of a tracked object when an EEM applet is triggered.
<b>event manager applet</b>	Registers an event applet with the Embedded Event Manager and enters applet configuration mode.
<b>show track</b>	Displays tracking information.
<b>track stub</b>	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 a pplet 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

<i>label</i>	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.
<i>string-op1</i>	Specifies the first operand.
<i>operator</i>	Value used with the <i>string-op1</i> and <i>string-op2</i> operands that determines how the current counter value is compared to the entry value or the exit value. Valid values are: <ul style="list-style-type: none"> <li>• <b>gt</b> --Greater than.</li> <li>• <b>ge</b> --Greater than or equal to.</li> <li>• <b>eq</b> --Equal to.</li> <li>• <b>ne</b> --Not equal to.</li> <li>• <b>lt</b> --Less than.</li> <li>• <b>le</b> --Less than or equal to.</li> </ul>
<i>string-op2</i>	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
<b>action else</b>	Identifies the beginning of an else block in the if/else conditional block.
<b>action elseif</b>	Identifies the beginning of the if/else conditional block.
<b>action if</b>	Identifies the beginning of an if conditional block.
<b>event manager applet</b>	Registers an event applet with the EEM and enters applet 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

<b>tag</b>	Specifies a tag using the <i>event-tag</i> argument that can be used with the <b>attribute</b> command to associate an event.
<i>event-tag</i>	String that identifies the tag.
<b>occurs</b>	(Optional) Specifies the number of occurrences before an EEM event is triggered. If not specified, an EEM event is triggered on the first occurrence.
<i>occurs-value</i>	(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
<b>correlate</b>	Builds a single complex event.
<b>trigger (EEM)</b>	Enters trigger applet configuration mode and specifies the multiple event configuration statements for an EEM applet.

## description (EEM)

To describe what an Embedded Event Manager (EEM) applet does, use the **description** (EEM) command in applet configuration mode. To remove the description of an applet, use the **no** form of this command.

**description** *line*

**no description**

### Syntax Description

<i>line</i>	A brief description of a policy, upto 240 characters.
-------------	---

### Command Default

By default, no description is specified for an applet.

### Command Modes

Applet configuration (config-applet)

### Command History

Release	Modification
15.0(1)M	This command was introduced.

### Usage Guidelines

Use this command to describe what an EEM applet does. It is valid to have applets without a description. The Description of an applet can be added in any order, before or after any other applet configuration. Configuring a new description for an applet that already has a description, overwrites the current description.

### Examples

The following example shows how to add or modify the description for an EEM:

```
Router(config)# event manager applet one
Router(config-applet)# description "This applet looks for the word count in syslog messages"
Router(config-applet)# event syslog pattern
"count"
Router(config-applet)# action 1 syslog msg hi
```

### Related Commands

Command	Description
<b>show event manager policy active</b>	Displays EEM policies that are executed.
<b>show event manager policy available</b>	Displays EEM policies that are available to be registered.