

EEM Action Tcl Command Extension

The following conventions are used for the syntax documented on the Tcl command extension pages:

- An optional argument is shown within square brackets, for example:

[type ?]

- A question mark ? represents a variable to be entered.
- Choices between arguments are represented by pipes, for example:

priority low|normal|high

Note For all EEM Tcl command extensions, if there is an error, the returned Tcl result string contains the error information.

Note Arguments for which no numeric range is specified take an integer from -2147483648 to 2147483647, inclusive.

- action_policy, on page 2
- action_process, on page 2
- action_program, on page 4
- action_reload, on page 4
- action_script, on page 5
- action_snmp_trap, on page 6
- action_snmp_object_value, on page 6
- action_switch, on page 7
- action_syslog, on page 8
- action_track_read, on page 8
- action_track_set, on page 9

action_policy

Allows a Tcl script to run an Embedded Event Manager (EEM) policy that has been registered with the None event detector. The action of running an EEM policy can also be performed using the **event manager run** command.

Syntax

```
action_policy ?
```

Arguments

? (represents a string)	(Mandatory) The name of the EEM policy to be scheduled for execution. The policy must have been previously registered with the None event detector.

None

Result String

None

Set _cerrno

Yes

```
(_cerr_sub_err = 2) FH_ESYSERR (generic/unknown error from OS/system)
```

This error means that the operating system reported an error. The POSIX errno value that is reported with the error should be used to determine the cause of the operating system error.

```
(_cerr_sub_err = 12)    FH_ENOSUCHEID (unknown event ID)
```

This error means that the policy is unknown because it is not registered.

```
(_cerr_sub_err = 14)    FH_ENOSUCHACTION  (unknown action type)
```

This error means that the action command requested was unknown.

action_process

Starts, restarts, or kills a Software Modularity process. This Tcl command extension is supported only in Software Modularity images.

Syntax

```
action_process start|restart|kill [job_id ?]
[process_name ?] [instance ?]
```

Arguments

start	(Mandatory) Specifies that a process is to be started.
restart	(Mandatory) Specifies that a process is to be restarted.
kill	(Mandatory) Specifies that a process is to be stopped (killed).
job_id	(Optional) System manager assigned job ID for the process. If you specify this argument, it must be an integer between 1 and 4294967295, inclusive.
process_name	(Optional) Process name. Either job_id must be specified or process_name and instance must be specified.
instance	(Optional) Process instance ID. If you specify this argument, it must be an integer between 1 and 4294967295, inclusive.

Result String

None

Set _cerrno

Yes

```
(_cerr_sub_err = 14)    FH_ENOSUCHACTION  (unknown action type)
```

This error means that the action command requested was unknown.

```
(_cerr_sub_num = 425, _cerr_sub_err = 1) SYSMGR_ERROR_INVALID_ARGS  (Invalid arguments
passed)
```

This error means that the arguments passed in were invalid.

```
(_cerr_sub_num = 425, _cerr_sub_err = 2) SYSMGR_ERROR_NO_MEMORY  (Could not allocate required
 memory)
```

This error means that an internal SYSMGR request for memory failed.

```
(_cerr_sub_num = 425, _cerr_sub_err = 5) SYSMGR_ERROR_NO_MATCH  (This process is not known
 to sysmgr)
```

This error means that the process name was not known.

```
(_cerr_sub_num = 425, _cerr_sub_err = 14) SYSMGR_ERROR_TOO_BIG  (outside the valid limit)
```

This error means that an object size exceeded its maximum.

```
(_cerr_sub_num = 425, _cerr_sub_err = 15) SYSMGR_ERROR_INVALID_OP  (Invalid operation for
this process)
```

This error means that the operation was invalid for the process.

action_program

Allows a Tcl script to run a POSIX process (program), optionally with a given argument string, environment string, Standard Input (stdin) pathname, Standard Output (stdout) pathname, or Standard Error (stderr) pathname. This Tcl command extension is supported only in Software Modularity images.

Syntax

```
action_program path ? [argv ?] [envp ?] [stdin ?] [stdout ?] [stderr ?]
```

Arguments

path	(Mandatory) The pathname of a program to run.
argv	(Optional) The argument string of the program.
envp	(Optional) The environment string of the program.
stdin	(Optional) The pathname for stdin.
stdout	(Optional) The pathname for stdout.
stderr	(Optional) The pathname for stderr.

Result String

None

Set _cerrno

Yes

```
(_cerr_sub_err =  2)    FH_ESYSERR  (generic/unknown error from OS/system)
```

This error means that the operating system reported an error. The POSIX errno value that is reported with the error should be used to determine the cause of the operating system error.

```
(_cerr_sub_err = 14)    FH_ENOSUCHACTION  (unknown action type)
```

This error means that the action command requested was unknown.

```
(_cerr_sub_err = 34)    FH_EMAXLEN  (maximum length exceeded)
```

This error means that the object length or number exceeded the maximum.

action_reload

Reloads the device.

Syntax

```
action_reload
```

Arguments

None

Result String

None

Set _cerrno

Yes

```
(_cerr_sub_err =  2)    FH_ESYSERR  (generic/unknown error from OS/system)
```

This error means that the operating system reported an error. The POSIX errno value that is reported with the error should be used to determine the cause of the operating system error.

```
(_cerr_sub_err = 14)    FH_ENOSUCHACTION  (unknown action type)
```

This error means that the action command requested was unknown.

action_script

Allows a Tcl script to enable or disable the execution of all Tcl scripts (enables or disables the script scheduler).

Syntax

```
action_script [status enable|disable]
```

Arguments

status	(Optional) Flag to indicate script execution status. If this argument is set to enable, script execution is enabled; if this argument is set to disable, script execution is disabled.

Result String

None

Set _cerrno

Yes

```
(_cerr_sub_err =  2)    FH_ESYSERR  (generic/unknown error from OS/system)
```

This error means that the operating system reported an error. The POSIX errno value that is reported with the error should be used to determine the cause of the operating system error.

```
(_cerr_sub_err = 14)    FH_ENOSUCHACTION  (unknown action type)
```

This error means that the action command requested was unknown.

```
(_cerr_sub_err = 52)    FH_ECONFIG  (configuration error)
```

This error means that a configuration error has occurred.

action_snmp_trap

Sends a Simple Network Management Protocol (SNMP) trap using the Embedded Event Manager Notification MIB.

Syntax

```
action_snmp_trap [intdata1 ?] [intdata2 ?] [strdata ?]
```

Arguments

intdata1	(Optional) Arbitrary integer sent in trap.
intdata2	(Optional) Arbitrary integer sent in trap.
strdata	(Optional) Arbitrary string data sent in trap.

Result String

None

Set _cerrno

Yes

```
(_cerr_sub_err =  2)    FH_ESYSERR  (generic/unknown error from OS/system)
```

This error means that the operating system reported an error. The POSIX errno value that is reported with the error should be used to determine the cause of the operating system error.

```
(_cerr_sub_err = 14)    FH_ENOSUCHACTION  (unknown action type)
```

This error means that the action command requested was unknown.

action_snmp_object_value

Sets a Simple Network Management Protocol (SNMP) object ID and value to be returned for the SNMP get request.

Syntax

```
action_snmp_object_value {int|uint|counter|gauge|ipv4|octet|counter64|string} ?
[next_oid ?]
```

Arguments

int	A 32-bit number used to specify a numbered type within the context of a managed object.
uint	A 32-bit number used to represent decimal value.
counter	A 32-bit number with a minimum value of 0.
gauge	A 32-bit number with a minimum value of 0.
ipv4	IP version 4 address.
octet	An octet string in hex notation used to represent physical addresses.
counter 64	A 64-bit number with a minimum value of 0.
string	An octet string in text notation used to represent text strings.
next_oid	The OID of the next object in the table; NULL if it is the last object in the table.

Result String

None

Set _cerrno

Yes

action_switch

Switches processing to a secondary processor in a fully redundant environment. Before using the **action_switch** Tcl command extension, you must install a backup processor in the device. If the hardware is not fully redundant, the switchover action will not be performed.

Syntax

```
action_switch
```

Arguments

None

Result String

None

Set _cerrno

Yes

```
(_cerr_sub_err = 2) FH_ESYSERR (generic/unknown error from OS/system)
```

This error means that the operating system reported an error. The POSIX errno value that is reported with the error should be used to determine the cause of the operating system error.

```
(_cerr_sub_err = 14)    FH_ENOSUCHACTION  (unknown action type)
```

This error means that the action command requested was unknown.

action_syslog

Generates a periodic syslog message using the specified facility when an EEM script is triggered.

Syntax

```
action_syslog [priority emerg|alert|crit|err|warning|notice|info|debug]
[msg ?] [facility ?]
```

Arguments

priority	(Optional) The action_syslog message facility level. If this argument is not specified, the default priority is LOG_INFO.
msg	(Optional) The message to be logged.
facility	(Optional) Syslog facility.

Result String

None

Set _cerrno

Yes

action_track_read

Reads the state of a tracked object when an Embedded Event Manager (EEM) script is triggered.

Syntax

```
action_track_read ?
```

Arguments

? (represents a number)	(Mandatory) Tracked object number in the range from 1 to 500, inclusive.

Result String

```
number {%u}
state {%s}
```

Set _cerrno

Yes

```
FH_ENOTRACK
```

This error means that the tracked object number was not found.

action_track_set

Sets the state of a tracked object when an Embedded Event Manager (EEM) script is triggered.

Syntax

```
action_track_set ? state up|down
```

Arguments

? (represents a number)	(Mandatory) Tracked object number in the range from 1 to 500, inclusive.
state	(Mandatory) Specifies that the state of the tracked object will be set. If up is specified, the state of the tracked object will be set to up. If down is specified, the state of the tracked object will be set to down.

Result String

None

Set _cerrno

Yes

```
FH_ENOTRACK
```

This error means that the tracked object number was not found.