



# EEM Context Library Command Extensions

All the Tcl context library command extensions belong to the `::cisco::eem` namespace.

- [context\\_retrieve](#), on page 1
- [context\\_save](#), on page 4

## context\_retrieve

Retrieves Tcl variable(s) identified by the given context name, and possibly the scalar variable name, the array variable name, and the array index. Retrieved information is automatically deleted.



**Note** Once saved information is retrieved, it is automatically deleted. If that information is needed by another policy, the policy that retrieves it (using the **context\_retrieve** command extension) should also save it again (using the **context\_save** command extension).

### Syntax

```
context_retrieve ctxt [var] [index_if_array]
```

### Arguments

ctxt	(Mandatory) Context name.
var	(Optional) Scalar variable name or array variable name. Defaults to a null string if this argument is not specified.
index_if_array	(Optional) The array index.



**Note** The `index_if_array` argument will be ignored when the `var` argument is a scalar variable.

If `var` is unspecified, retrieves the whole variable table saved in the context.

If `var` is specified and `index_if_array` is not specified, or if `index_if_array` is specified but `var` is a scalar variable, retrieves the value of `var`.

If `var` is specified, and `index_if_array` is specified, and `var` is an array variable, retrieves the value of the specified array element.

### Result String

Resets the Tcl global variables to the state that they were in when the save was performed.

### Set\_cerrno

- A string displaying `_cerrno`, `_cerr_sub_num`, `_cerr_sub_err`, `_cerr_posix_err`, `_cerr_str` due to `appl_reqinfo` error.
- Variable is not in the context.

### Sample Usage

The following examples show how to use the `context_save` and `context_retrieve` command extension functionality to save and retrieve data. The examples are shown in save and retrieve pairs.

#### Example 1: Save

If `var` is unspecified or if a pattern is specified, saves multiple variables to the context.

```
::cisco::eem::event_register_none
namespace import ::cisco::eem::*
namespace import ::cisco::lib::*
set testvara 123
set testvarb 345
set testvarc 789
if {[catch {context_save TESTCTX "testvar*"} errmsg]} {
    action_syslog msg "context_save failed: $errmsg"
} else {
    action_syslog msg "context_save succeeded"
}
```

#### Example 1: Retrieve

If `var` is unspecified, retrieves multiple variables from the context.

```
::cisco::eem::event_register_none
namespace import ::cisco::eem::*
namespace import ::cisco::lib::*

if {[catch {foreach {var value} [context_retrieve TESTCTX] {set $var $value}} errmsg]} {
    action_syslog msg "context_retrieve failed: $errmsg"
} else {
    action_syslog msg "context_retrieve succeeded"
}

if {[info exists testvara]} {
    action_syslog msg "testvara exists and is $testvara"
} else {
    action_syslog msg "testvara does not exist"
}

if {[info exists testvarb]} {
    action_syslog msg "testvarb exists and is $testvarb"
}
```

```

} else {
    action_syslog msg "testvarb does not exist"
}
if {[info exists testvarc]} {
    action_syslog msg "testvarc exists and is $testvarc"
} else {
    action_syslog msg "testvarc does not exist"
}

```

### Example 2: Save

If var is specified, saves the value of var.

```

::cisco::eem::event_register_none

namespace import ::cisco::eem::*
namespace import ::cisco::lib::*
set testvar 123
if {[catch {context_save TESTCTX testvar} errmsg]} {
    action_syslog msg "context_save failed: $errmsg"
} else {
    action_syslog msg "context_save succeeded"
}

```

### Example 2: Retrieve

If var is specified and index\_if\_array is not specified, or if index\_if\_array is specified but var is a scalar variable, retrieves the value of var.

```

::cisco::eem::event_register_none
namespace import ::cisco::eem::*
namespace import ::cisco::lib::*
if {[catch {set testvar [context_retrieve TESTCTX testvar]} errmsg]} {
    action_syslog msg "context_retrieve failed: $errmsg"
} else {
    action_syslog msg "context_retrieve succeeded"
}
if {[info exists testvar]} {
    action_syslog msg "testvar exists and is $testvar"
} else {
    action_syslog msg "testvar does not exist"
}

```

### Example 3: Save

If var is specified, saves the value of var even if it is an array.

```

::cisco::eem::event_register_none

namespace import ::cisco::eem::*
namespace import ::cisco::lib::*
array set testvar "testvar1 ok testvar2 not_ok"
if {[catch {context_save TESTCTX testvar} errmsg]} {
    action_syslog msg "context_save failed: $errmsg"
} else {
    action_syslog msg "context_save succeeded"
}

```

**Example 3: Retrieve**

If `var` is specified, and `index_if_array` is not specified, and `var` is an array variable, retrieves the entire array.

```
::cisco::eem::event_register_none
namespace import ::cisco::eem::*
namespace import ::cisco::lib::*
if {[catch {array set testvar [context_retrieve TESTCTX testvar]} errmsg]} {
    action_syslog msg "context_retrieve failed: $errmsg"
} else {
    action_syslog msg "context_retrieve succeeded"
}
if {[info exists testvar]} {
    action_syslog msg "testvar exists and is [array get testvar]"
} else {
    action_syslog msg "testvar does not exist"
}
```

**Example 4: Save**

If `var` is specified, saves the value of `var` even if it is an array.

```
::cisco::eem::event_register_none
namespace import ::cisco::eem::*
namespace import ::cisco::lib::*
array set testvar "testvar1 ok testvar2 not_ok"
if {[catch {context_save TESTCTX testvar} errmsg]} {
    action_syslog msg "context_save failed: $errmsg"
} else {
    action_syslog msg "context_save succeeded"
}
```

**Example 4: Retrieve**

If `var` is specified, and `index_if_array` is specified, and `var` is an array variable, retrieves the specified array element value.

```
::cisco::eem::event_register_none
namespace import ::cisco::eem::*
namespace import ::cisco::lib::*
if {[catch {set testvar [context_retrieve TESTCTX testvar testvar1]} errmsg]} {
    action_syslog msg "context_retrieve failed: $errmsg"
} else {
    action_syslog msg "context_retrieve succeeded"
}
if {[info exists testvar]} {
    action_syslog msg "testvar exists and is $testvar"
} else {
    action_syslog msg "testvar doesn't exist"
}
```

## context\_save

Saves Tcl variables that match a given pattern in current and global namespaces with the given context name as identification. Use this Tcl command extension to save information outside of a policy. Saved information can be retrieved by a different policy using the **context\_retrieve** command extension.



**Note** Once saved information is retrieved, it is automatically deleted. If that information is needed by another policy, the policy that retrieves it (using the **context\_retrieve** command extension) should also save it again (using the **context\_save** command extension).

### Syntax

```
context_save ctxt [pattern]
```

### Arguments

ctxt	(Mandatory) Context name.
pattern	<p>(Optional) The glob-style pattern as used by the <b>string match</b>Tcl command. If this argument is not specified, the pattern defaults to the wildcard *.</p> <p>There are three constructs used in glob patterns:</p> <ul style="list-style-type: none"> <li>• * = all characters</li> <li>• ? = 1 character</li> <li>• [abc] = match one of a set of characters</li> </ul>

### Result String

None

### Set\_cerrno

A string displaying \_cerrno, \_cerr\_sub\_num, \_cerr\_sub\_err, \_cerr\_posix\_err, \_cerr\_str due to appl\_setinfo error.

### Sample Usage

The following examples show how to use the **context\_save** and **context\_retrieve** command extension functionality to save and retrieve data. The examples are shown in save and retrieve pairs.

#### Example 1: Save

If var is unspecified or if a pattern is specified, saves multiple variables to the context.

```
::cisco::eem::event_register_none
namespace import ::cisco::eem::*
namespace import ::cisco::lib::*
set testvara 123
set testvarb 345
set testvarc 789
if {[catch {context_save TESTCTX "testvar*"} errmsg]} {
    action_syslog msg "context_save failed: $errmsg"
} else {
    action_syslog msg "context_save succeeded"
}
```

**Example 1: Retrieve**

If var is unspecified, retrieves multiple variables from the context.

```

::cisco::eem::event_register_none
namespace import ::cisco::eem::*
namespace import ::cisco::lib::*

if {[catch {foreach {var value} [context_retrieve TESTCTX] {set $var $value}} errmsg]} {
    action_syslog msg "context_retrieve failed: $errmsg"
} else {
    action_syslog msg "context_retrieve succeeded"
}

if {[info exists testvara]} {
    action_syslog msg "testvara exists and is $testvara"
} else {
    action_syslog msg "testvara does not exist"
}

if {[info exists testvarb]} {
    action_syslog msg "testvarb exists and is $testvarb"
} else {
    action_syslog msg "testvarb does not exist"
}

if {[info exists testvarc]} {
    action_syslog msg "testvarc exists and is $testvarc"
} else {
    action_syslog msg "testvarc does not exist"
}

```

**Example 2: Save**

If var is specified, saves the value of var.

```

::cisco::eem::event_register_none

namespace import ::cisco::eem::*
namespace import ::cisco::lib::*
set testvar 123
if {[catch {context_save TESTCTX testvar} errmsg]} {
    action_syslog msg "context_save failed: $errmsg"
} else {
    action_syslog msg "context_save succeeded"
}

```

**Example 2: Retrieve**

If var is specified and index\_if\_array is not specified, or if index\_if\_array is specified but var is a scalar variable, retrieves the value of var.

```

::cisco::eem::event_register_none
namespace import ::cisco::eem::*
namespace import ::cisco::lib::*
if {[catch {set testvar [context_retrieve TESTCTX testvar]} errmsg]} {
    action_syslog msg "context_retrieve failed: $errmsg"
} else {
    action_syslog msg "context_retrieve succeeded"
}

if {[info exists testvar]} {
    action_syslog msg "testvar exists and is $testvar"
} else {

```

```

        action_syslog msg "testvar does not exist"
    }

```

### Example 3: Save

If var is specified, saves the value of var even if it is an array.

```

::cisco::eem::event_register_none

namespace import ::cisco::eem::*
namespace import ::cisco::lib::*
array set testvar "testvar1 ok testvar2 not_ok"
if {[catch {context_save TESTCTX testvar} errmsg]} {
    action_syslog msg "context_save failed: $errmsg"
} else {
    action_syslog msg "context_save succeeded"
}

```

### Example 3: Retrieve

If var is specified, and index\_if\_array is not specified, and var is an array variable, retrieves the entire array.

```

::cisco::eem::event_register_none
namespace import ::cisco::eem::*
namespace import ::cisco::lib::*
if {[catch {array set testvar [context_retrieve TESTCTX testvar]} errmsg]} {
    action_syslog msg "context_retrieve failed: $errmsg"
} else {
    action_syslog msg "context_retrieve succeeded"
}
if {[info exists testvar]} {
    action_syslog msg "testvar exists and is [array get testvar]"
} else {
    action_syslog msg "testvar does not exist"
}

```

### Example 4: Save

If var is specified, saves the value of var even if it is an array.

```

::cisco::eem::event_register_none
namespace import ::cisco::eem::*
namespace import ::cisco::lib::*
array set testvar "testvar1 ok testvar2 not_ok"
if {[catch {context_save TESTCTX testvar} errmsg]} {
    action_syslog msg "context_save failed: $errmsg"
} else {
    action_syslog msg "context_save succeeded"
}

```

### Example 4: Retrieve

If var is specified, and index\_if\_array is specified, and var is an array variable, retrieves the specified array element value.

```

::cisco::eem::event_register_none
namespace import ::cisco::eem::*
namespace import ::cisco::lib::*
if {[catch {set testvar [context_retrieve TESTCTX testvar testvar1]} errmsg]} {

```

```
        action_syslog msg "context_retrieve failed: $errmsg"
    } else {
        action_syslog msg "context_retrieve succeeded"
    }
    if {[info exists testvar]} {
        action_syslog msg "testvar exists and is $testvar"
    } else {
        action_syslog msg "testvar doesn't exist"
    }
}
```