EEM コンテキスト ライブラリのコマンド拡張

すべての Tcl コンテキスト ライブラリ コマンド拡張は、::cisco::eem 名前空間に属します。

- context_retrieve （1 ページ）
- context_save （5 ページ）

context_retrieve

該当するコンテキスト名、使用されている可能性があるスカラ変数名、配列型変数名、および配列の索引によって指定される Tcl 変数を取得します。取得される情報は、自動的に削除されます。

（注） 保存される情報が一度取得されると、自動的に削除されます。その情報が別のポリシーで必要な場合、（**context_retrieve** コマンド拡張を使用して）それを取得するポリシーも、（**context_save** コマンド拡張を使用して）再度保存する必要があります。

Syntax

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

引数

ctxt	（必須）コンテキスト名。
var	（任意）スカラ変数名または配列型変数名。この引数が指定されない場合、ヌル文字列を定義します。
index_if_array	（任意）配列の索引。

（注） var 引数がスカラ変数の場合、index_if_array 引数は無視されます。

var が未指定の場合、コンテキストに保存されている変数テーブル全体を取得します。

var が指定され、index_if_array が指定されない場合、または、index_if_array が指定されるが var がスカラ変数の場合、var の値を取得します。

var が指定され、index_if_array が指定され、var が配列変数の場合、指定された配列エレメントの値を取得します。

結果文字列

保存が実行されたときの状態に、Tcl グローバル変数をリセットします。

_cerrno を設定

- appl_reqinfo エラーが原因で、_cerrno、_cerr_sub_num、_cerr_sub_err、_cerr_posix_err、_cerr_str を表示する文字列。
- 変数がコンテキストにない。

使用例

次に、**context_save** コマンド拡張機能および **context_retrieve** コマンド拡張機能を使用して、データを保存し、取得する例を示します。例は、保存と取得のペアで示されます。

例 1：保存

var が未指定か、またはパターンが指定される場合、複数の変数をコンテキストに保存します。

```
::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"
}
```

例 1：取得

var が未指定の場合、複数の変数をコンテキストから取得します。

```
::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"
}
```

例 2：保存

var が指定される場合、var の値を保存します。

```
::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"
}
```

例 2：取得

var が指定され、index_if_array が指定されない場合、または、index_if_array が指定されるが var がスカラ変数の場合、var の値を取得します。

```
::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"
}
```

例 3：保存

var が指定される場合、それが配列の場合でも、var の値を保存します。

```
::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"
}
```

例 3：取得

var が指定され、index_if_array が指定されず、var が配列変数の場合、配列全体を取得します。

```
::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"
}
```

例 4：保存

var が指定される場合、それが配列の場合でも、var の値を保存します。

```
::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"
}
```

例 4：取得

var が指定され、index_if_array が指定され、var が配列変数の場合、指定された配列エレメントの値を取得します。

```
::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

現在およびグローバルな名前空間で、指定されたパターンが、識別情報として指定されたコンテキスト名と一致する、Tcl変数を保存します。このTclコマンド拡張を使用すると、ポリシー外の情報が保存されます。保存された情報は、**context_retrieve**コマンド拡張を使用して、異なるポリシーによって取得できます。

（注） 保存される情報が一度取得されると、自動的に削除されます。その情報が別のポリシーで必要な場合、（**context_retrieve** コマンド拡張を使用して）それを取得するポリシーも、（**context_save** コマンド拡張を使用して）再度保存する必要があります。

Syntax

```
context_save ctxt [pattern]
```

引数

ctxt	（必須）コンテキスト名。
pattern	（任意）**string match** コマンドによって使用される、glob-style パターン。この引数が指定されない場合、パターンのデフォルトは、ワイルドカード * です。 glob パターンで使用されている、3 つの構成があります。 • * = すべての文字 • ? = 1 文字 • [abc] = 文字のセットの 1 つと照合

結果文字列

なし

_cerrno を設定

appl_setinfo エラーが原因で、_cerrno、_cerr_sub_num、_cerr_sub_err、_cerr_posix_err、_cerr_str を表示する文字列。

使用例

次に、**context_save** コマンド拡張機能および **context_retrieve** コマンド拡張機能を使用して、データを保存し、取得する例を示します。例は、保存と取得のペアで示されます。

例 1：保存

varが未指定か、またはパターンが指定される場合、複数の変数をコンテキストに保存します。

```
::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"
}
```

例 1：取得

var が未指定の場合、複数の変数をコンテキストから取得します。

```
::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"
}
```

例 2：保存

var が指定される場合、var の値を保存します。

```
::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"
}
```

例 2：取得

var が指定され、index_if_array が指定されない場合、または、index_if_array が指定されるが var がスカラ変数の場合、var の値を取得します。

```
::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"
}
```

例 3：保存

var が指定される場合、それが配列の場合でも、var の値を保存します。

```
::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"
}
```

例 3：取得

var が指定され、index_if_array が指定されず、var が配列変数の場合、配列全体を取得します。

```
::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"
}
```

例 4：保存

var が指定される場合、それが配列の場合でも、var の値を保存します。

```
::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"
}
```

例 4：取得

var が指定され、index_if_array が指定され、var が配列変数の場合、指定された配列エレメントの値を取得します。

```
::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"
}
```