Relational Association Programming

in the Tcl© Scripting Language

Command Reference Documentation

© Aaron Sami Abassi

Licensed under the Academic Free License version 3.0

Table Of Contents

1 Source File

1.1 rap.tcl

This file is the Tcl source code for the 'rap' package. It can be incorporated through the Tcl package management system, through source file inclusion or directly incorporated into project source code including it's original copyright notice.

Tcl Package Management Example

```
package require rap
namespace import ::rap::*
```

Source File Inclusion Example

```
source rap.tcl
namespace import ::rap::*
```

Direct Source Incorporation Example

```
# RAP-Tcl
# © Aaron Sami Abassi
# Licensed under the Academic Free License version 3.0
namespace eval ::rap {
  # ...
}
namespace import ::rap::*
```

2 Scripting Commands

Documentation Format

command

parameter

Description of the parameter which is optional when named within a pair of “?”.

...

Description of the command.

```
command example
```

2.1 Data Commands

2.1.1 conformation

name

The conformation array name which may be name space qualified.

?elements?

The optional list of conformation element names.

This command alias creates and optionally initializes a name space conformation array variable with semantics similar to the Tcl variable command. The element values in the resulting array yield list indices. When used in a procedure body and the optional list of element names is not present, this command links the local array to the name space array. Because Tcl does not have types, this command is strictly for facilitation and legibility of code and does not strictly match the RAP definition of a conformation.

```
conformation simple_CO {simple1_IN simple2_IN}
```

2.1.2 information

name

The information variable name which may be name space qualified.

?value?

The optional scalar value.

This command alias creates and optionally initializes a name space information variable with semantics similar to the Tcl variable command. When used in a procedure body and the optional list of element names is not present, this command links the local variable to the name space variable.

```
information simple_IN {0 0}
```

2.1.3 classification

name

The classification array name which may be name space qualified.

?elements?

The optional list of classification element names.

This command alias creates and optionally initializes a name space classification array variable with semantics similar to the Tcl variable command. The element values in the resulting array yield list indices. When used in a procedure body and the optional list of element names is not present, this command links the local variable to the name space variable. Because Tcl does not have types, this command is strictly for facilitation and legibility of code and does not strictly match the RAP definition of a classification.

```
classification simple_CL {simple1_FNLO simple2_FNLO}
```

2.1.4 objectification

name

The objectification variable name which may be name space qualified.

?functions?

The optional list of function commands which may each be name space qualified.

This command alias creates and optionally initializes a name space objectification variable with semantics similar to the Tcl variable command. When used in a procedure body and the optional list of function names is not present, this command links the local variable to the name space variable.

```
objectification simple_OB {simple1_FN simple2_FN}
```

2.1.5 situation

name

The situation array name which may be name space qualified.

?elements?

The optional list of classification element names.

This command alias creates and optionally initializes a name space situation array variable with semantics similar to the Tcl variable command. The element values in the resulting array yield list indices. When used in a procedure body and the optional list of element names is not present, this command links the local variable to the name space variable. Because Tcl does not have types, this command is strictly for facilitation and legibility of code and does not strictly match the RAP definition of a situation.

```
situation simple_COSI {simple1_INLO simple2_INLO}
```

2.1.6 location

name

The situation variable name which may be name space qualified.

?value?

The optional location value.

This command alias creates and optionally initializes a name space location variable with semantics similar to the Tcl variable command. When used in a procedure body and the optional list of element names is not present, this command links the local variable to the name space variable.

```
location simple_INLO {simple1_IN simple2_IN}
```

2.2 Code Commands

2.2.1 abstraction

name

The abstraction variable name which may be name space qualified.

?arguments?

The optional argument list.

This command alias creates and optionally initializes a name space abstraction variable with semantics similar to the Tcl variable command. When used in a procedure body and the optional list of element names is not present, this command links the local variable to the name space variable. Because Tcl does not have types, this command is strictly for facilitation and does not strictly match the RAP definition of an abstraction.

```
abstraction simple_AB param_IN
```

2.2.2 function

name

The new procedure command which may be name space qualified.

arguments

The list of arguments for the procedure such as an abstraction value.

body

The body of the procedure.

This command creates a procedure as a function and exports it from the name space.

```
function simple1_FN $simple_AB {puts $param_IN}
```

2.3 Modularization Commands

2.3.1 association

name

The association name space which may be name space qualified.

body

An association definition body.

?body ...?

Further definition body parts which are concatenated then evaluated.

This command alias defines an association in a name space. Subsequent calls to this command with the same name parameter will add to the definition of the association. This command is an alias for namespace eval.

```
association ::simple {
  abstraction simple_AB param_IN
  function simple_FN $simple_AB {
    puts $param_IN
  }
}
```

2.3.2 relations

?args?

The list of association name spaces.

This command relates all name space variables and exported commands from the listed name spaces into the current association name space. This command does not force the import of variables or commands, therefore if a name collision is unavoidable read the source code for the relations command in the rap.tcl file and import/upvar them individually.

```
relations ::simple
simple1_FN "Hello World"
```

3 Copyrights

Tcl/Tk is copyrighted by the Regents of the University of California, Sun Microsystems, Inc., Scriptics Corporation, and other parties.