

NAME

cmip – A Tcl interface to the CMIP protocol.

DESCRIPTION

The **cmip** command provides an interface to the Common Management Information Protocol. It enables Tcl scripts to connect to a remote OSI management agent and to use the Common Management Information Services (CMIS).

CMIP COMMAND**cmip connect** *agent host*

The **cmip connect** command establishes an association to a remote *agent* running on *host*. This command returns a cmip association object which allows to invoke association specific commands (see below).

cmip wait

The **cmip wait** command blocks until all asynchronous requests for all CMIP associations have been processed. Events are processed while waiting for outstanding responses which can have arbitrary side effects.

cmip info

The **cmip info** command returns a list of all existing cmip association handles that have been created using the **cmip connect** command.

CMIP ASSOCIATION COMMAND**cmip# get** *class instance ?options?*

Retrieve management information on this association (cmip#). The *class* and the *instance* define the base managed object that will be returned, if not modified by the options.

The available *options* for get are:

-scope *scope* *?-atomic?*

Set the *scope* for the request. If *-atomic* is given, it tells the agent to perform the request only, if it can be performed on all scoped managed objects.

-filter *filter*

Set a *filter* on the scoped managed object(s).

-attributes *attributes*

The *attributes* is a list, that may contain the attributes to ask for. {} for no attributes. Default is all attributes. Each Element of the list may be a list, but only the first element is used as an attribute to look for (so e.g. the result of an get or set request may be used as the input).

-callback *callback*

Specify a *callback* that will be executed, if the result of the asynchronous request is completely answered.

cmip# set *class instance attributes ?options?*

Set management information at the agent (cmip#). The *class* and the *instance* define the base managed object on which *attributes* will be set, if not modified by the options. The *attributes* argument is a list whose elements are a list of attribute-type, attribute-value and an *modify-operation* triples. The possible *modify-operations* are replace, addValue, removeValue and setToDefault. If no *modify-operation* is given, than replace is used.

The available *options* for set are:

-scope *scope* *?-atomic?*

Set the *scope* for the request. If -atomic is given, it tells the agent to perform the request only, if it can be performed on all scoped managed objects.

-filter *filter*

Set a *filter* on the scoped managed object(s).

-callback *callback*

Specify a *callback* that will be executed, if the result of the asynchronous request is completely answered.

-nonconfirmed

The set-option may be used as a confirmed or non-confirmed service. Default is to use the confirmed service, but if you want to use the unconfirmed service just use -nonconfirmed as one of the options.

cmip# action *class instance action ?options?*

Perform an action on management information at the agent (cmip#). The *class* and the *instance* define the base managed object on which the *action* will be performed, if not modified by the options. The *action* is a list containing the action-type and an optional action-value.

The available *options* for set are:

-scope *scope* *?-atomic?*

Set the *scope* for the request. If -atomic is given, it tells the agent to perform the request only, if it can be performed on all scoped managed objects.

-filter *filter*

Set a *filter* on the scoped managed object(s).

-callback *callback*

Specify a *callback* that will be executed, if the result of the asynchronous request is completely answered.

-nonconfirmed

The action-option may be used as a confirmed or non-confirmed service. Default is to use the confirmed service, but if you want to use the unconfirmed service just use

-nonconfirmed as one of the options.

cmip# create *class ?options?*

Create a managed object (MO) at the agent (cmip#).

The available *options* for create are:

-instance *instance* | **-superior** *superiorInst*

The *instance* specifies the object instance name, alternatively the *superiorInst* specifies the object that is the parent in the containment tree

-reference *referenceInst*

The *referenceInst* specifies a reference object instance to be used for initial attribute values to be copied.

-attributes *attributes*

The *attributes* specifies initial attribute values, it contains a list of attribute-type and attribute-value pairs.

-callback *callback*

Specify a *callback* that will be executed, if the result of the asynchronous request is complete.

cmip# delete *class instance ?options?*

Delete managed object(s) at the agent (cmip#). The *class* and the *instance* define the base managed object which will be deleted, if not modified by the options.

The available *options* for delete are:

-scope *scope* *?-atomic?*

Set the *scope* for the request. If *-atomic* is given, it tells the agent to perform the request only, if it can be performed on all scoped managed objects.

-filter *filter*

Set a *filter* on the scoped managed object(s).

-callback *callback*

Specify a *callback* that will be executed, if the result of the asynchronous request is complete.

cmip# cancelGet *requestHandle* *?-callback callback?*

Cancel the outstanding asynchronous GET request at the agent (cmip#), given by *requestHandle*. If a callback is given, it will be executed, if the result of the asynchronous request is complete.

cmip# eventSink ?-callback callback?

Initialize a callback for event reports. If no callback is given, this will wait for event reports till eternity and show event reports emitted by MOs of the agent.

cmip# requests

The **cmip# requests** command returns a list of all outstanding asynchronous requests.

cmip# wait

The **cmip# wait** command blocks until all asynchronous requests for this association are processed. Events are processed while waiting for outstanding responses which can have arbitrary side effects.

cmip# release

The **cmip# release** command releases the association and deletes the cmip# object. The **cmip# release** command should be the normal way to terminate an association.

cmip# abort

The **cmip# abort** command aborts the association immediately and deletes the cmip# object.

BUGS

The current implementation of the **cmip** commands described above depends on the OSIMIS package which itself depends on the ISODE package. Both packages are no longer freely available and both packages tend to be complicated to install.

SEE ALSO

scotty(1), Tcl(n)

AUTHORS

Michael Kernchen <kernchen@ibr.cs.tu-bs.de>

Juergen Schoenwaelder <schoenw@cs.utwente.nl>