The omniORB utilities

Eoin Carroll

Revised: 13 July 2000

1 catior

Usage:

```
catior [-x] <stringified IOR>
```

catior is a utility for viewing components of a stringified IOR. It displays the components of the stringified object reference supplied to it.

The options are:

`-x` Display the object key in hexadecimal.

2 genior

Usage:

```
genior [-x] <Type ID> <hostname> <port number> [object key]
```

genior generates a stringified object reference from the arguments supplied to it. If an object key argument isn't supplied, it will use an object key generated by omniORB2.

The options are:

`-x` Interpret the object key as a hexadecimal value. This value should begin with "0x"

3 nameclt

Usage:

```
nameclt [-ior <object-reference>] [-advanced] <operation>
```

The nameclt command invokes operations on the Naming Service.

3.1 Operations

The allowed operations are:

`list <context-name>`

lists contexts and objects bound in the context with the specified name.

`bind_new_context <context-name>`

binds name to a new context, and returns the stringified context IOR.

`remove_context <context-name>`

unbinds and destroys the named context, as long as it is empty.

`bind <object-name> <stringified-IOR>`

binds name to object.

`unbind <object-name>`

unbinds name and object.

`resolve <object-name>`

returns stringified IOR bound to specified name.

3.2 Options

The options are:

`-ior <NameService-object-reference>`

Use the given stringified IOR as the "root" context of the naming service. By default, nameclt uses the object reference returned by calling:
`CORBA::ORB::resolve_initial_references("NameService")`

`-advanced`

Allow advanced operations. These are operations which should not normally need to be used. They may however be useful for testing the naming service and also for cleaning up in the event of a client messing up the namespace. The operations are:

`bind_context <context-name> <stringified-IOR>`
binds name to context.

`rebind <object-name> <stringified-IOR>`
binds name to object even if binding already exists.

`rebind_context <context-name> <stringified-IOR>`
binds name to context even if binding already exists.

`new_context`
returns stringified IOR for a new context.

`destroy`
destroys the naming context given with `-ior` flag.