



NAOMI Sequencer API ICD

Version : 1.0

Craige Bevil

Document Identifier : WHT-NAOMI-16

14 August 2000

**Isaac Newton Group,
Apartado 321, 38780 S/C La Palma,
Tenerife, Canary Islands**

Telephone +34 922 425400
Fax +34 922 425401
Internet cb@ing.iac.es

1. Overview

This document describes the interface and the functionality that the NAOMI sequencer API is expected to offer. In particular it describes the interface through which a remote client of the API can initiate commands within the NAOMI sequencer and subsequently monitor their progress.

This document will be enhanced as development of the NAOMI sequencer interface progresses but in the first instance, the document will serve as a standard to bind together various development processes.

2. Interface Description

This chapter describes how to include the API into a client application and the functionality that the API offers.

2.1 Importing the API into a Python Script

The API is coded in Python which means that all client software which makes immediate use of the interface *must* also be coded in Python. The API will be imported into a client application using the standard *import* mechanism offered by the Python interpreter. The name of the module will be *ElectraMPI*. Client applications will include the module into their process space by importing the module before any use of the API using the following device:

```
import ElectraMPI
```

Prior to loading the package, the user must ensure that the module is located in a directory that is in the search path of the Python interpreter. This can be done by setting the environment variable **PYTHONPATH** to include the directory that the module is located within before starting the API client task. An example of how this may be done from the *cs*h command line prompt is as follows;

```
setenv PYTHONPATH /opt/csg/lib
```

2.2 The *CommandToElectra* Command

The **CommandToElectra** function will be used by a client of the API to initiate commands remotely in the NAOMI sequencer. The function will return *immediately* after the remote NAOMI sequencer command has been initiated and will return as output, a Python dictionary which contains status information relating to the initial execution of the remote command in the NAOMI sequencer.

Under *no circumstances* will the function raise an exception, instead all internal exceptions will be caught internally and reported by means of the *command status dictionary* which is returned as the result of the function.

2.2.1 Function API

```
ElectraMPI.CommandToElectra (commandToBeExecuted)
```

2.2.2 Function Input

The command takes a single argument which is *commandToBeExecuted*. This will be the complete Python representation of the remote command in the NAOMI Sequencer to be executed. The command will subsequently be transferred to the NAOMI sequencer where it will be executed. An example of calling the function will be as follows;

```
ElectraMPI.CommandToElectra("OpenWFSloop()")
```

This will result in the *OpenWFSloop* function being executed in the NAOMI sequencer and the WFS control loop being opened.

2.2.3 Function Outputs

The *CommandToElectra* command will be expected in *all circumstances* to return a Python based dictionary consisting of *all* of the following elements.

2.2.3.1 CommandStatus

This is the current state of the command being executed in the sequencer and should be set to one of the following values.

Note : Character case is significant when specifying the value.

<i>Value</i>	<i>Semantics</i>
<i>Error</i>	An error was encountered when the NAOMI sequencer attempted to execute the command. Further information relating to the error should be provided in the <i>ErrorMessage</i> and the <i>ErrorStatus</i> elements of the command status dictionary.
<i>Running</i>	The command was started successfully and is now running inside of the NAOMI sequencer.
<i>CompletedOk</i>	The command was started successfully within the NAOMI sequencer and has completed without any error.
<i>TimedOut</i>	The amount of time allocated by the NAOMI sequencer for this command to complete has been exceeded and the command has been flagged as timed out.

2.2.3.2 StatusReturned

StatusReturned reflects the status of the interaction between the API and the remote NAOMI sequencer. It can be set to one of a number of values indicated in the following table.

Note : Character case is significant when specifying the value field.

<i>Value</i>	<i>Semantics</i>
<i>BE_FAILED</i>	The NAOMI sequencer API was unable to establish communications with the NAOMI sequencer.
<i>BE_OK</i>	The API was able to communicate with the NAOMI sequencer without error.
<i>BE_UNKNOWN</i>	The command that has been issued was not recognised by the NAOMI sequencer.

2.2.3.3 ErrorMessage

The *ErrorMessage* element in the command status dictionary contains textual information to accompany an error that might have occurred during the execution of a command in the NAOMI sequencer.

If no error message text is present, the value of this dictionary element should be set to "".

2.2.3.4 ErrorStatus

The *ErrorStatus* element in the command status dictionary contains an error code associated with an error that occurs throughout the execution of a command in the NAOMI sequencer. If no error status is present, the value of this dictionary element should be set to "".

2.2.3.5 Tag

This is the name of a unique *tag* which is generated by the API and which is returned to the client after a command is initially started within the NAOMI sequencer. This *tag* will subsequently be used as an *index* by the client of the API to re-interrogate the status of *this* particular invocation of the NAOMI sequencer command in order to establish if it completed successfully.

In the case that an error was encountered during the initial call to the **CommandToElectra** function for *whatever* reason, no tag will be returned by the API. Instead the *tag* element in the dictionary returned will be set to "".

2.2.3.6 Example Command Status Dictionary

Following is a number of cases of possible Python command dictionaries that may be returned in a variety of different situations. The dictionaries are expressed in Python to eliminate ambiguity.

In the case of the API client attempting to start a NAOMI sequencer command that is already in progress. Note that no *tag* is returned in this case.

```
{'ErrorStatus':100, 'ErrorMessage':'Action already in progress',
'StatusReturned':'BE_OK', 'Tag':'', 'CommandStatus' : 'Error'}
```

In the case of complete communications failure with NAOMI sequencer. Note that no *tag* is returned in this case.

```
{'ErrorStatus':'', 'ErrorMessage':'', 'StatusReturned':'BE_FAILED',
'Tag':'', 'CommandStatus' : 'Error'}
```

In the case that a command is started successfully. Note that in this case a *tag* is returned so that the client can monitor the progress of the execution of the command once control is returned to it.

```
{'ErrorStatus':'', 'ErrorMessage':'', 'StatusReturned':'BE_OK',
'Tag':'tag123', 'CommandStatus' : 'Running'}
```

2.3 The *CommandStatus* Command

This function will be used by a client of the NAOMI Sequencer API to interrogate the status of a command running in the NAOMI sequencer that it previously initiated. The *command status dictionary* returned by the **CommandToElectra** will include the name of a unique *tag* associated with that particular invocation of the sequencer command that can be used by the client to monitor the progress of the command.

2.3.1 Function API

```
ElectraMPI.commandStatus(tag)
```

2.3.2 Function Input

The command accepts a single argument which is the name of the *tag* associated with the command running in the NAOMI sequencer for which status information is required.

If the *tag* that is provided cannot be associated with a command running in the NAOMI sequencer, the function will raise an exception to that effect and no *command status dictionary* will be returned.

2.3.3 Function Output

The function will return to the caller a Python dictionary describing the current state of the command running in the NAOMI sequencer associated with the *tag* that was specified as the input argument to the function. See section 2.2.3 for details relating to the expected content of the command dictionary returned by the function.

2.4 The *FreeStatus* Command

This function will be used by a client of the NAOMI sequencer API to instruct the API that it no longer has an interest in the command status information associated with the specified tag. Any further reference to the *tag* using the **CommandStatus** function (see section 2.3) will result in an error.

Note : It will be the responsibility of the client of the API to ensure that the API is informed when a tag is no longer needed, the API will perform no automatic house keeping.

2.4.1 Function API

```
ElectraMPI.FreeStatus (Tag)
```

2.4.2 Function Input

The function accepts a single argument *Tag* which is the name of the tag which the API is to remove. In the case that the *tag* is not recognised, the function will raise an exception.

2.4.3 Function Output

The function does not return any output on completion.