RF Cavity wrapper
User's Guide

RFCavityWrapper Class

Revision: FirstRelease - Author: vedder
Implemented in C++ - CVS repository: ESRF

Introduction:

This TANGO class will wrap TACO High level server for the control of the rf cavities.

Class Identification:

- **Contact :** at esrf.fr - vedder
- **Class Family :** RadioFrequency (Specific Specific)
- **Platform :** All Platforms
- **Bus :** Not Applicable

Class Inheritance:

- Tango::Device_4Impl
  - RFCavityWrapper

Properties:

<table>
<tr><th colspan="3">Device Properties</th></tr>
<tr><th>Property name</th><th>Property type</th><th>Description</th></tr>
<tr><td>**Cavity_taco_device**</td><td>Tango::DEV_STRING</td><td>This is the TACO high level server that will be wrapped.</td></tr>
<tr><td>**Hardware_access_delay**</td><td>Tango::DEV_USHORT</td><td>Read on taco server are done in one pass. This property is the delay between two access to the taco device. This is done to avoid taco access on each attribute read.</td></tr>
<tr><td>**Configuration_file_path**</td><td>Tango::DEV_STRING</td><td>This is the default configuration file directory.</td></tr>
</table>

Device Properties Default Values:

Property Name	Default Values
Cavity_taco_device	No default value
Hardware_access_delay	3
Configuration_file_path	No default value

There is no Class properties.

Attributes:

<table>
<tr><th colspan="4">Scalar Attributes</th></tr>
<tr><th>Attribute name</th><th>Data Type</th><th>R/W Type</th><th>Expert</th></tr>
<tr><td>**Cavity_mode**</td><td>DEV_STRING</td><td>READ</td><td>No</td></tr>
<tr><td>**Configuration_file**</td><td>DEV_STRING</td><td>READ</td><td>No</td></tr>
<tr><td>**Cavity_1_voltage**</td><td>DEV_DOUBLE</td><td>READ</td><td>No</td></tr>
<tr><td>**Cavity_2_voltage**</td><td>DEV_DOUBLE</td><td>READ</td><td>No</td></tr>
<tr><td>**Cavity_1_vacuum**</td><td>DEV_DOUBLE</td><td>READ</td><td>No</td></tr>
<tr><td>**Cavity_2_vacuum**</td><td>DEV_DOUBLE</td><td>READ</td><td>No</td></tr>
<tr><td>**Tuner_1_cav_l_pos**</td><td>DEV_DOUBLE</td><td>READ</td><td>No</td></tr>
<tr><td>**Tuner_1_cav_r_pos**</td><td>DEV_DOUBLE</td><td>READ</td><td>No</td></tr>
<tr><td>**Tuner_2_cav_l_pos**</td><td>DEV_DOUBLE</td><td>READ</td><td>No</td></tr>
<tr><td>**Tuner_2_cav_r_pos**</td><td>DEV_DOUBLE</td><td>READ</td><td>No</td></tr>
<tr><td>**Configuration_file_path**</td><td>DEV_STRING</td><td>READ</td><td>No</td></tr>
<tr><td>**ShortStatus**</td><td>DEV_STRING</td><td>READ</td><td>No</td></tr>
</table>

Spectrum Attributes

Attribute name	Data Type	X Data Length	Expert
SubDevicesStates	DEV_STATE	22	No
SequenceHistory	DEV_STRING	4096	No

Commands:

More Details on commands....

Device Commands for Operator Level

Command name	Argument In	Argument Out
Init	DEV_VOID	DEV_VOID
State	DEV_VOID	DEV_STATE
Status	DEV_VOID	CONST_DEV_STRING
On	DEV_VOID	DEV_VOID
Off	DEV_VOID	DEV_VOID
Stop	DEV_VOID	DEV_VOID
Reset	DEV_VOID	DEV_VOID
ReadConfigurationFile	DEV_STRING	DEV_VOID
WriteConfigurationFile	DEV_STRING	DEV_VOID
GetConfigurationFilePath	DEV_VOID	DEV_STRING

1 - Init

- **Description:** This commands re-initialise a device keeping the same network connection.
  After an Init command executed on a device, it is not necessary for client to re-connect to the device.
  This command first calls the device *delete_device()* method and then execute its *init_device()* method.
  For C++ device server, all the memory allocated in the *nit_device()* method must be freed in the *delete_device()* method.
  The language device desctructor automatically calls the *delete_device()* method.

- **Argin:**
  DEV_VOID : none.

- **Argout:**
  DEV_VOID : none.

- **Command allowed for:**

2 - State

- **Description:** This command gets the device state (stored in its *device_state* data member) and returns it to the caller.

- **Argin:**
  DEV_VOID : none.

- **Argout:**
  DEV_STATE : State Code

- **Command allowed for:**

3 - Status

- **Description:** This command gets the device status (stored in its *device_status* data member) and returns it to the caller.

- **Argin:**
  DEV_VOID : none.

- **Argout:**
  CONST_DEV_STRING : Status description

- **Command allowed for:**

4 - On

- **Description:**

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**

5 - Off

- **Description:**

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**

6 - Stop

- **Description:**

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**

7 - Reset

- **Description:**

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**

8 - ReadConfigurationFile

- **Description:** Load a configuration file (wraps DevRead taco command).

- **Argin:**
  DEV_STRING :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**

9 - WriteConfigurationFile

- **Description:** Save a configuration file (wraps DevWrite taco command).

- **Argin:**
  DEV_STRING :

- **Argout:**
  DEV_VOID :

- **Command allowed for:**

10 - GetConfigurationFilePath

- **Description:** Configuration_file_path device property is returned by this command.

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_STRING : Configuration_file_path is returned.

- **Command allowed for:**