

# MicrocodeDataViewer Tango Cpp Class

## Contents :

- [Description](#)
- [Properties](#)
- [Commands](#)
  - [State](#)
  - [Status](#)
  - [Start](#)
  - [Abort](#)
- [Attributes](#)
- [States](#)

## MicrocodeDataViewer Class

### Identification :

Contact : at null - null  
Class Family :  
Platform :  
Bus :  
Manufacturer :  
Manufacturer ref. :

## MicrocodeDataViewer Class

### Inheritance :

- [Tango::DeviceImpl](#)
  - MicrocodeDataViewer

## MicrocodeDataViewer Class Description :

This device allows to read/write microcode variables of a GalliBox using dynamic attributes defined in properties

## MicrocodeDataViewer Properties :

There is no class properties

Device Properties			
Name	Description	Type	Default Value
MicrocodeName	The GalliBox microcode name	String	none
ProcessId	The GalliBox microcode process id	String	none
GalilBoxDeviceName	The complete name of the GalilBox device	String	Test/GalilBox/dev1
Data	Describe the data that should be read/write on the GalilBox. For each line the device will create an attribute with the specified name and type. Format ::: : the name of the attribute that will be created : BOOLEAN, LONG, DOUBLE : READ or READ_WRITE : the name of the microcode var	String[]	none
ControlboxCommunicationCommand	The tango command to use to communicate with the control box. Microcode V1.0 : WriteRead Microcode V2.0 : ExecLowLevelCmd Default : ExecLowLevelCmd	String	ExecLowLevelCmd

MicrocodeDataViewer Class Commands				
Name	Input type	Output type	Level	Description
<a href="#">State</a>	DEV_VOID	DEV_STATE	OPERATOR	This command gets the device state (stored in its <i>device_state</i> data member) and returns it to the caller.
<a href="#">Status</a>	DEV_VOID	CONST_DEV_STRING	OPERATOR	This command gets the device status (stored in its <i>device_status</i> data member) and returns it to the caller.
<a href="#">Start</a>	DEV_VOID	DEV_VOID	OPERATOR	Start the microcode
<a href="#">Abort</a>	DEV_VOID	DEV_VOID	EXPERT	Abort the microcode execution. CAUTION : This function kills the running process, which is not the proper way to stop. USE WITH CARE

### **Command State :**

This command gets the device state (stored in its *device\_state* data member) and returns it to the caller.

State Definition		
Input Argument	Tango::DEV_VOID	none.
Output Argument	Tango::DEV_STATE	State Code
DisplayLevel	OPERATOR	..
Inherited	true	..
Abstract	true	..
Polling Period	Not polled	..
Command allowed for	All states	..

---

## **Command Status :**

This command gets the device status (stored in its *device\_status* data member) and returns it to the caller.

Status Definition		
Input Argument	Tango::DEV_VOID	none.
Output Argument	Tango::CONST_DEV_STRING	Status description
DisplayLevel	OPERATOR	..
Inherited	true	..
Abstract	true	..
Polling Period	Not polled	..
Command allowed for	All states	..

---

## **Command Start :**

Start the microcode

Start Definition		
Input Argument	Tango::DEV_VOID	
Output Argument	Tango::DEV_VOID	
DisplayLevel	OPERATOR	..
Inherited	false	..
Abstract	false	..
Polling Period	Not polled	..

Command allowed for	All states	..
---------------------	------------	----

---

## **Command Abort :**

Abort the microcode execution. CAUTION : This function kills the running process, which is not the proper way to stop. USE WITH CARE

Abort Definition		
Input Argument	Tango::DEV_VOID	
Output Argument	Tango::DEV_VOID	
DisplayLevel	EXPERT	..
Inherited	false	..
Abstract	false	..
Polling Period	Not polled	..
Command allowed for	All states	..

---

**There is no attribute defined.**

---

**There is no dynamic attribute defined.**

---



---

MicrocodeDataViewer Class States	
Name	Description
RUNNING	The microcode is RUNNING
STANDBY	The microcode is NOT RUNNING