


TANGO Device Server

gpidDeviceServer User's Guide

GpibDeviceServer Class

Revision: release_3_0_4 - Author: buteau
Implemented in C++

Introduction:

This server is a generic gpib interface.

Class Inheritance:

- Tango::Device_3Impl
  - GpibDeviceServer

Properties:

Device Properties		
Property name	**Property type**	**Description**
GpibBoardName	Tango::DEV_STRING	This is the name of the board where gpib device is plugged. e.g : gpib1
GpibDeviceAddress	Tango::DEV_SHORT	This is gpidDevice address.
GpibDeviceTimeOut	Tango::DEV_SHORT	This is the GPIB device Time Out. Warning this is a predefined value: #define TNONE 0 Infinite timeout (disabled) #define T10us 1 Timeout of 10 us (ideal) #define T30us 2 Timeout of 30 us (ideal) #define T100us 3 Timeout of 100 us (ideal) #define T300us 4 Timeout of 300 us (ideal) #define T1ms 5 Timeout of 1 ms (ideal) #define T3ms 6 Timeout of 3 ms (ideal) #define T10ms 7 Timeout of 10 ms (ideal) #define T30ms 8 Timeout of 30 ms (ideal) #define T100ms 9 Timeout of 100 ms (ideal) #define T300ms 10 Timeout of 300 ms (ideal) #define T1s 11 Timeout of 1 s (ideal) #define T3s 12 Timeout of 3 s (ideal) #define T10s 13 Timeout of 10 s (ideal) #define T30s 14 Timeout of 30 s (ideal) #define T100s 15 Timeout of 100 s (ideal) #define T300s 16 Timeout of 300 s (ideal) #define T1000s 17 Timeout of 1000 s (maximum

Device Properties Default Values:

Property Name	**Default Values**
GpibBoardName	No default value
GpibDeviceAddress	No default value
GpibDeviceTimeOut	No default value

There is no Class properties.

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
Write	DEV_STRING	DEV_VOID
Read	DEV_VOID	DEV_STRING
WriteRead	DEV_STRING	DEV_STRING
ReceiveBinData	DEV_LONG	DEVVAR_CHARARRAY
SendBinData	DEVVAR_CHARARRAY	DEV_VOID

Device Commands for Expert Level Only		
Command name	**Argument In**	**Argument Out**
ReadLongString	DEV_LONG	DEV_STRING
SetTimeOut	DEV_SHORT	DEV_VOID
Trigger	DEV_VOID	DEV_VOID
Getiberr	DEV_VOID	DEV_LONG
Getibsta	DEV_VOID	DEV_LONG
Getibcnt	DEV_VOID	DEV_ULONG

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 - Write

- **Description:** This command gets the device status (stored in its *device_status* data member) and returns it to the caller.
- **Argin:**
  DEV_STRING : String to send to the device
- **Argout:**
  DEV_VOID :
- **Command allowed for:**

5 - Read

- **Description:** This command reads a string from a gpib device. Throws an DevFailed exception on error
- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_STRING : Returned string.

- **Command allowed for:**

6 - ReadLongString (for expert only)

- **Description:** For most gpib device the read command is enough to talk with the device. In certain case, the gpibDevice returns a very big string, which is larger than the read buffer. For concret example, on :CALC:DATA? command, the hp3588 returns 400 strings representing a spectrum, for a total size > 4Kbytes. This method reads these sort of long string. For efficiency, its better to use the read command instead of readLongString. This method is provided for exceptionnal case. Throws DevFailed on error.

- **Argin:**
  DEV_LONG : Max expected string length.

- **Argout:**
  DEV_STRING : The readed string.

- **Command allowed for:**

7 - WriteRead

- **Description:** This command perform a write on the GPIB device, and then perform a read to get the answer, before returning it.

- **Argin:**
  DEV_STRING : String to send to the gpib device.

- **Argout:**
  DEV_STRING : String returned by the gpib Device.

- **Command allowed for:**

8 - ReceiveBinData

- **Description:** This command reads an array of binary data from a gpib device. Up to 65536 bytes. In generaly, a Gpib device can send or receive 64Ko. Throws an DevFailed exception on error

- **Argin:**
  DEV_LONG : length of the data to receive from the Gpib device

- **Argout:**
  DEVVAR_CHARARRAY : Array of binary data

- **Command allowed for:**

9 - SendBinData

- **Description:** This command send an array of binary data to the device through the GPIB bus. Throws devFailed on error.

- **Argin:**
  DEVVAR_CHARARRAY : Array of binary data to send to the device

- **Argout:**
  DEV_VOID :

- **Command allowed for:**

10 - SetTimeOut (for expert only)

- **Description:** This command set Time Out value for the gpib device. Warning these values are predefined, cf gpibDevice.h accepted value are [0-15]. Throws DevFailed exception on error.

- **Argin:**
  DEV_SHORT : accepted value are [0-15]

- **Argout:**
  DEV_VOID : no argout

- **Command allowed for:**

11 - Trigger (for expert only)

- **Description:** This command sends a trigger signal to the GPIB device. If the device was previously set up, it can make its measurment, and send it on the bus. Measure is now get with a read command.

- **Argin:**
  DEV_VOID : no argin

- **Argout:**
  DEV_VOID : no argout

- **Command allowed for:**

12 - Getiberr (for expert only)

- **Description:** This command returns last gpib device error code (lib gpib iberr). Throws DevFailed on error.
- **Argin:**
  DEV_VOID : no argin
- **Argout:**
  DEV_LONG : no argout
- **Command allowed for:**

13 - Getibsta (for expert only)

- **Description:** This command returns last gpib device state code (lib gpib ibsta). Throws DevFailed on error.
- **Argin:**
  DEV_VOID : no argin
- **Argout:**
  DEV_LONG : no argout
- **Command allowed for:**

14 - Getibcnt (for expert only)

- **Description:** This command returns last gpib device count var (lib gpib ibcnt). Throws DevFailed on error.
- **Argin:**
  DEV_VOID : no argin
- **Argout:**
  DEV_ULONG : no argout
- **Command allowed for:**

ESRF - Software Engineering Group




TANGO Device Server

gpidDeviceServer User's Guide

GpibDeviceServer Class

Revision: release_3_0_4 - Author: buteau
Implemented in C++

Introduction:

This server is a generic gpib interface.

Class Inheritance:

- Tango::Device_3Impl
  - GpibDeviceServer

Properties:

Device Properties		
Property name	**Property type**	**Description**
GpibBoardName	Tango::DEV_STRING	This is the name of the board where gpib device is plugged. e.g : gpib1
GpibDeviceAddress	Tango::DEV_SHORT	This is gpidDevice address.
GpibDeviceTimeOut	Tango::DEV_SHORT	This is the GPIB device Time Out. Warning this is a predefined value: #define TNONE 0 Infinite timeout (disabled) #define T10us 1 Timeout of 10 us (ideal) #define T30us 2 Timeout of 30 us (ideal) #define T100us 3 Timeout of 100 us (ideal) #define T300us 4 Timeout of 300 us (ideal) #define T1ms 5 Timeout of 1 ms (ideal) #define T3ms 6 Timeout of 3 ms (ideal) #define T10ms 7 Timeout of 10 ms (ideal) #define T30ms 8 Timeout of 30 ms (ideal) #define T100ms 9 Timeout of 100 ms (ideal) #define T300ms 10 Timeout of 300 ms (ideal) #define T1s 11 Timeout of 1 s (ideal) #define T3s 12 Timeout of 3 s (ideal) #define T10s 13 Timeout of 10 s (ideal) #define T30s 14 Timeout of 30 s (ideal) #define T100s 15 Timeout of 100 s (ideal) #define T300s 16 Timeout of 300 s (ideal) #define T1000s 17 Timeout of 1000 s (maximum

Device Properties Default Values:

Property Name	**Default Values**
GpibBoardName	No default value
GpibDeviceAddress	No default value
GpibDeviceTimeOut	No default value

There is no Class properties.

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
Write	DEV_STRING	DEV_VOID
Read	DEV_VOID	DEV_STRING
WriteRead	DEV_STRING	DEV_STRING
ReceiveBinData	DEV_LONG	DEVVAR_CHARARRAY
SendBinData	DEVVAR_CHARARRAY	DEV_VOID

Device Commands for Expert Level Only		
Command name	**Argument In**	**Argument Out**
ReadLongString	DEV_LONG	DEV_STRING
SetTimeOut	DEV_SHORT	DEV_VOID
Trigger	DEV_VOID	DEV_VOID
Getiberr	DEV_VOID	DEV_LONG
Getibsta	DEV_VOID	DEV_LONG
Getibcnt	DEV_VOID	DEV_ULONG

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 - Write

- **Description:** This command gets the device status (stored in its *device_status* data member) and returns it to the caller.
- **Argin:**
  DEV_STRING : String to send to the device
- **Argout:**
  DEV_VOID :
- **Command allowed for:**

5 - Read

- **Description:** This command reads a string from a gpib device. Throws an DevFailed exception on error
- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_STRING : Returned string.

- **Command allowed for:**

6 - ReadLongString (for expert only)

- **Description:** For most gpib device the read command is enough to talk with the device. In certain case, the gpibDevice returns a very big string, which is larger than the read buffer. For concret example, on :CALC:DATA? command, the hp3588 returns 400 strings representing a spectrum, for a total size > 4Kbytes. This method reads these sort of long string. For efficiency, its better to use the read command instead of readLongString. This method is provided for exceptionnal case. Throws DevFailed on error.

- **Argin:**
  DEV_LONG : Max expected string length.

- **Argout:**
  DEV_STRING : The readed string.

- **Command allowed for:**

7 - WriteRead

- **Description:** This command perform a write on the GPIB device, and then perform a read to get the answer, before returning it.

- **Argin:**
  DEV_STRING : String to send to the gpib device.

- **Argout:**
  DEV_STRING : String returned by the gpib Device.

- **Command allowed for:**

8 - ReceiveBinData

- **Description:** This command reads an array of binary data from a gpib device. Up to 65536 bytes. In generaly, a Gpib device can send or receive 64Ko. Throws an DevFailed exception on error

- **Argin:**
  DEV_LONG : length of the data to receive from the Gpib device

- **Argout:**
  DEVVAR_CHARARRAY : Array of binary data

- **Command allowed for:**

9 - SendBinData

- **Description:** This command send an array of binary data to the device through the GPIB bus. Throws devFailed on error.

- **Argin:**
  DEVVAR_CHARARRAY : Array of binary data to send to the device

- **Argout:**
  DEV_VOID :

- **Command allowed for:**

10 - SetTimeOut (for expert only)

- **Description:** This command set Time Out value for the gpib device. Warning these values are predefined, cf gpibDevice.h accepted value are [0-15]. Throws DevFailed exception on error.

- **Argin:**
  DEV_SHORT : accepted value are [0-15]

- **Argout:**
  DEV_VOID : no argout

- **Command allowed for:**

11 - Trigger (for expert only)

- **Description:** This command sends a trigger signal to the GPIB device. If the device was previously set up, it can make its measurment, and send it on the bus. Measure is now get with a read command.

- **Argin:**
  DEV_VOID : no argin

- **Argout:**
  DEV_VOID : no argout

- **Command allowed for:**

12 - Getiberr (for expert only)

- **Description:** This command returns last gpib device error code (lib gpib iberr). Throws DevFailed on error.
- **Argin:**
  DEV_VOID : no argin
- **Argout:**
  DEV_LONG : no argout
- **Command allowed for:**

13 - Getibsta (for expert only)

- **Description:** This command returns last gpib device state code (lib gpib ibsta). Throws DevFailed on error.
- **Argin:**
  DEV_VOID : no argin
- **Argout:**
  DEV_LONG : no argout
- **Command allowed for:**

14 - Getibcnt (for expert only)

- **Description:** This command returns last gpib device count var (lib gpib ibcnt). Throws DevFailed on error.
- **Argin:**
  DEV_VOID : no argin
- **Argout:**
  DEV_ULONG : no argout
- **Command allowed for:**

ESRF - Software Engineering Group

Frame Alert

This document is designed to be viewed using the frames feature. If you see this message, you are using a non-frame-capable web client.
Link to Non-frame version.




TANGO Device Server

---

gpidDeviceServer
Device Commands Description
GpibDeviceServer Class

Revision: release_3_0_4 - Author: buteau

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 - Write

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

- **Argin:**
  DEV_STRING : String to send to the device

- **Argout:**
  DEV_VOID :

- **Command allowed for:**

5 - Read

- **Description:** This command reads a string from a gpib device. Throws an DevFailed exception on error

- **Argin:**
  DEV_VOID :

- **Argout:**
  DEV_STRING : Returned string.

- **Command allowed for:**

6 - ReadLongString (for expert only)

- **Description:** For most gpib device the read command is enough to talk with the device. In certain case, the gpibDevice returns a very big string, which is larger than the read buffer. For concret example, on :CALC:DATA? command, the hp3588 returns 400 strings representing a spectrum, for a total size > 4Kbytes. This method reads these sort of long string. For efficiency, its better to

use the read command instead of readLongString. This method is provided for exceptionnal case. Throws DevFailed on error.

- **Argin:**
  DEV_LONG : Max expected string length.

- **Argout:**
  DEV_STRING : The readed string.

- **Command allowed for:**

7 - WriteRead

- **Description:** This command perform a write on the GPIB device, and then perform a read to get the answer, before returning it.

- **Argin:**
  DEV_STRING : String to send to the gpib device.

- **Argout:**
  DEV_STRING : String returned by the gpib Device.

- **Command allowed for:**

8 - ReceiveBinData

- **Description:** This command reads an array of binary data from a gpib device. Up to 65536 bytes. In generaly, a Gpib device can send or receive 64Ko. Throws an DevFailed exception on error

- **Argin:**
  DEV_LONG : length of the data to receive from the Gpib device

- **Argout:**
  DEVVAR_CHARARRAY : Array of binary data

- **Command allowed for:**

9 - SendBinData

- **Description:** This command send an array of binary data to the device through the GPIB bus. Throws devFailed on error.

- **Argin:**
  DEVVAR_CHARARRAY : Array of binary data to send to the device

- **Argout:**
  DEV_VOID :

- **Command allowed for:**

10 - SetTimeOut (for expert only)

- **Description:** This command set Time Out value for the gpib device. Warning these values are predefined, cf gpibDevice.h accepted value are [0-15]. Throws DevFailed exception on error.

- **Argin:**
  DEV_SHORT : accepted value are [0-15]

- **Argout:**
  DEV_VOID : no argout

- **Command allowed for:**

11 - Trigger (for expert only)

- **Description:** This command sends a trigger signal to the GPIB device. If the device was previously set up, it can make its measurment, and send it on the bus. Measure is now get with a read command.

- **Argin:**
  DEV_VOID : no argin

- **Argout:**
  DEV_VOID : no argout

- **Command allowed for:**

12 - Getiberr (for expert only)

- **Description:** This command returns last gpib device error code (lib gpib iberr). Throws DevFailed on error.

- **Argin:**
  DEV_VOID : no argin

- **Argout:**
  DEV_LONG : no argout

- **Command allowed for:**

13 - Getibsta (for expert only)

- **Description:** This command returns last gpib device state code (lib gpib ibsta). Throws DevFailed on error.

- **Argin:**
  DEV_VOID : no argin

- **Argout:**
  DEV_LONG : no argout

- **Command allowed for:**

14 - Getibcnt (for expert only)

- **Description:** This command returns last gpib device count var (lib gpib ibcnt). Throws DevFailed on error.

- **Argin:**
  DEV_VOID : no argin

- **Argout:**
  DEV_ULONG : no argout

- **Command allowed for:**
