TangoInterface

Generated by Doxygen 1.5.6

Fri Dec 4 12:49:14 2009

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all files with brief descriptions:

Chapter 3

Class Documentation

3.1 CTangoInterfaceApp Class Reference

`#include <TangoInterfaceApp.h>`

Public Member Functions

- CTangoInterfaceApp ()
- virtual BOOL InitInstance ()

3.1.1 Constructor & Destructor Documentation

3.1.1.1 CTangoInterfaceApp::CTangoInterfaceApp ()

3.1.2 Member Function Documentation

3.1.2.1 BOOL CTangoInterfaceApp::InitInstance () `[virtual]`

The documentation for this class was generated from the following files:

- /homenfs/amilan/workspace/Xradia-dll/LastVersion/TangoInterface_-20091201/TangoInterfaceApp.h
- /homenfs/amilan/workspace/Xradia-dll/LastVersion/TangoInterface_-20091201/TangoInterfaceApp.cpp

3.2 TangoInterface Class Reference

`#include <TangoInterface.h>`

Public Member Functions

- TangoInterface (void)
  Default constructor.
- ~TangoInterface (void)
  Destructor.
- bool GetErrorMessage (CString &strErrorMsg)
  Gets an error message.
- bool SetMotorSpeed (CString strMotorIdentifier, float fSpeed)
  Sets the motor speed.
- bool GetMotorPosition (CString strMotorIdentifier, float &fPosition)
  Gets the motor position.
- bool EnableMotor (CString strMotorIdentifier, bool bEnable)
  Enables the motor.
- bool JogMotor (CString strMotorIdentifier, bool bMovePositive)
  Jog motor.
- bool GetHomingStatus (CString strMotorIdentifier, int &nHomingStatus)
  Get the homing status.
- bool Home (CString strMotorIdentifier)
  Issues the Home command.
- bool MoveMotor (CString strMotorIdentifier, float fPosition)
  Move motor.
- bool SetVariable (CString strVariableName, CString strValue)
  Sets a variable.
- bool SetVariable (CString strVariableName, int nValue)
  Sets a variable.
- bool SetVariable (CString strVariableName, float fValue)
  Sets a variable.
- bool SetVariable (CString strVariableName, bool bValue)
  Sets a variable.
- bool GetVariableFloat (CString strVariableName, float &fValue)

Gets a variable (float version).

- bool GetVariableInt (CString strVariableName, int &nValue)

  Gets a variable (integer version).

- bool GetVariableString (CString strVariableName, CString &strReturn)

  Gets a variable (string version).

- bool GetVariableBool (CString strVariableName, bool &bValue)

  Gets a variable (boolean version).

- bool GetMotorVariableFloat (CString strMotorIdentifier, CString strVariableName, float &fValue)

  Get a motor variable (float version).

- bool GetMotorVariableInt (CString strMotorIdentifier, CString strVariableName, int &nValue)

  Get a motor variable (integer version).

- bool GetMotorVariableString (CString strMotorIdentifier, CString strVariableName, CString &strReturn)

  Get a motor variable (string version).

- bool GetMotorVariableBool (CString strMotorIdentifier, CString strVariableName, bool &bValue)

  Get a motor variable (boolean version).

- int GetMotorAttributeStrings (CString *&pstrVariableNames, int nNumAttributes)

  Gets a specified list of motor variable names.

3.2.1 Constructor & Destructor Documentation

3.2.1.1 TangoInterface::TangoInterface (void)

Default constructor.

3.2.1.2 TangoInterface::~TangoInterface (void)

Destructor.

3.2.2 Member Function Documentation

3.2.2.1 bool TangoInterface::GetErrorMessage (CString & *strErrorMsg*)

Gets an error message.

Will return an error string describing the most recent failure.

Parameters:

strErrorMsg CString to recieve the error message string.

Returns:

true if the command succeeds, false if the command fails.

3.2.2.2 bool TangoInterface::SetMotorSpeed (CString *strMotorIdentifier*, float *fSpeed*)

Sets the motor speed.

Parameters:

strMotorIdentifier The motor identifier for the motor of interest.

fSpeed The speed value to set in counts per second.

Returns:

true if the command succeeds, false if the command fails.

3.2.2.3 bool TangoInterface::GetMotorPosition (CString *strMotorIdentifier*, float & *fPosition*)

Gets the motor position.

Parameters:

strMotorIdentifier The motor identifier for the motor of interest.

fPosition The motor position in counts.

Returns:

true if the command succeeds, false if the command fails.

3.2.2.4 bool TangoInterface::EnableMotor (CString *strMotorIdentifier*, bool *bEnable*)

Enables the motor.

Parameters:

strMotorIdentifier The motor identifier for the motor of interest.

bEnable true to enable it, false to disable it.

Returns:

true if the command succeeds, false if the command fails.

3.2.2.5 bool TangoInterface::JogMotor (CString *strMotorIdentifier*, bool *bMovePositive*)

Jog motor.

This will cause the motor of interest to jog continuously in the specified direction.

Parameters:

strMotorIdentifier The motor identifier for the motor of interest.

bMovePositive Direction to Jog. True for Positive, False for negative.

Returns:

true if the command succeeds, false if the command fails.

3.2.2.6 bool TangoInterface::GetHomingStatus (CString *strMotorIdentifier*, int & *nHomingStatus*)

Get the homing status.

Minimum necessary implemenation requires this function to return values 0 and 15. These corresponds to "homing not started" and "homing complete". If the motor does not require or allow homing this function, this function should return 15, home complete. /remarks>

Parameters:

strMotorIdentifier The motor identifier for the motor of interest.

nHomingStatus The homing status defined by the values 0-17 with the following definitions: HomingRoutineNotStartedYet = 0, WaitingforOther = 1, MovingToLimit = 2, IsAtLimit = 3, MovingToCoarseHome = 4, MovingToFixedOffset = 5, MovingToFineHome = 6, IsAtHome = 7, MovingToAlignedPos = 8, FollowedoutOnCoarse = 9, FollowedoutOnFixedOffset = 10, FollowedoutOnFineHome = 11, FollowedoutOnAlignedPos = 12, HitLimitOnFineHome = 13, HitLimitOnAlignedPos = 14, HomeComplete = 15, GeneralError = 16, PreconditionNotMet = 17

Returns:

true if the command succeeds, false if the command fails.

3.2.2.7 bool TangoInterface::Home (CString *strMotorIdentifier*)

Issues the Home command.

This command will cause the specified motor to initiate the homing process. If the motor does not require or allow homing this function, this function should return true.

Parameters:

strMotorIdentifier The motor identifier for the motor of interest.

Returns:

true if the command succeeds, false if the command fails.

3.2.2.8 bool TangoInterface::MoveMotor (CString *strMotorIdentifier*, float *fPosition*)

Move motor.

Parameters:

strMotorIdentifier The motor identifier for the motor of interest.

fPosition The ABSOLUTE position in counts.

Returns:

true if the command succeeds, false if the command fails.

3.2.2.9 bool TangoInterface::SetVariable (CString *strVariableName*, CString *strValue*)

Sets a variable.

Sets the specified variable to the specified value.

Parameters:

strVariableName The variable name.

strValue The value as a string.

Returns:

true if the command succeeds, false if the command fails.

3.2.2.10 bool TangoInterface::SetVariable (CString *strVariableName*, int *nValue*)

Sets a variable.

Sets the specified variable to the specified value.

Parameters:

strVariableName The variable name.

strValue The value as an integer.

Returns:

true if the command succeeds, false if the command fails.

3.2.2.11 bool TangoInterface::SetVariable (CString *strVariableName*, float *fValue*)

Sets a variable.

Sets the specified variable to the specified value.

Parameters:

strVariableName The variable name.

strValue The value as a float.

Returns:

true if the command succeeds, false if the command fails.

3.2.2.12 bool TangoInterface::SetVariable (CString *strVariableName*, bool *bValue*)

Sets a variable.

Sets the specified variable to the specified value.

Parameters:

strVariableName The variable name.

strValue The value as a boolean.

Returns:

true if the command succeeds, false if the command fails.

3.2.2.13 bool TangoInterface::GetVariableFloat (CString *strVariableName*, float & *fValue*)

Gets a variable (float version).

Gets the value of the specified variable.

Parameters:

strVariableName The variable name.

fValue the variable value.

Returns:

true if the command succeeds, false if the command fails.

3.2.2.14 bool TangoInterface::GetVariableInt (CString *strVariableName*, int & *nValue*)

Gets a variable (integer version).

Gets the value of the specified variable.

Parameters:

strVariableName The variable name.

nValue the variable value.

Returns:

true if the command succeeds, false if the command fails.

3.2.2.15 bool TangoInterface::GetVariableString (CString *strVariableName*, CString & *strReturn*)

Gets a variable (string version).

Gets the value of the specified variable.

Parameters:

strVariableName The variable name.

strReturn The variable value.

Returns:

true if the command succeeds, false if the command fails.

3.2.2.16 bool TangoInterface::GetVariableBool (CString *strVariableName*, bool & *bValue*)

Gets a variable (boolean version).

Gets the value of the specified variable.

Parameters:

strVariableName The variable name.

bValue The variable value.

Returns:

true if the command succeeds, false if the command fails.

3.2.2.17 bool TangoInterface::GetMotorVariableFloat (CString *strMotorIdentifier*, CString *strVariableName*, float & *fValue*)

Get a motor variable (float version).

Get the value of the specified variable for the specified motor.

Parameters:

strMotorIdentifier The motor identifier for the motor of interest.

strVariableName The variable name.

fValue the variable value.

Returns:

true if the command succeeds, false if the command fails.

3.2.2.18 bool TangoInterface::GetMotorVariableInt (CString *strMotorIdentifier*, CString *strVariableName*, int & *nValue*)

Get a motor variable (integer version).

Get the value of the specified variable for the specified motor.

Parameters:

strMotorIdentifier The motor identifier for the motor of interest.

strVariableName The variable name.

nValue the variable value.

Returns:

true if the command succeeds, false if the command fails.

3.2.2.19 bool TangoInterface::GetMotorVariableString (CString *strMotorIdentifier*, CString *strVariableName*, CString & *strReturn*)

Get a motor variable (string version).

Get the value of the specified variable for the specified motor.

Parameters:

strMotorIdentifier The motor identifier for the motor of interest.

strVariableName The variable name.

strReturn The variable value.

Returns:

true if the command succeeds, false if the command fails.

3.2.2.20 bool TangoInterface::GetMotorVariableBool (CString *strMotorIdentifier*, CString *strVariableName*, bool & *bValue*)

Get a motor variable (boolean version).

Get the value of the specified variable for the specified motor.

Parameters:

strMotorIdentifier The motor identifier for the motor of interest.

strVariableName The variable name.

bValue the variable value.

Returns:

true if the command succeeds, false if the command fails.

3.2.2.21 int TangoInterface::GetMotorAttributeStrings (CString *& *pstrVariableNames*, int *nNumAttributes*)

Gets a specified list of motor variable names.

This function is used to retrieve the variable names for a specific list of motor attributes. This list is as follows, in order: 0. Negative Limit – Variable is True when motor is at the negative-most limit 1. Positive Limit – Variable is True when motor is at the positive-most limit 2. In Position – Variable is True when the motor has stopped moving and it is within an error bound of its last commanded position 3. Following Error Occurred – Variable is True when the last commanded motion was aborted due to excessive following error 4. Enabled Status – Variable is True when motor is enabled 5. Position – Variable contains the current motor position in counts 6. Set Velocity – Variable contains the current motor target jog velocity

The pstrVariableNames parameter should be populated with the variable names for these motor attributes. This allows xradia software to use the GetMotorVariableXXXXX functions to check motor status.

It is an error to return more or less than the specified number of variables names or to return them in the wrong order.

Parameters:

pstrVariableNames a pointer to an array of CString objects which will recieve the expected data.

nNumAttributes the size of the allocated array.

Returns:

the number of CStrings returned.

The documentation for this class was generated from the following files:

- /homenfs/amilan/workspace/Xradia-dll/LastVersion/TangoInterface_20091201/TangoInterface.h
- /homenfs/amilan/workspace/Xradia-dll/LastVersion/TangoInterface_20091201/TangoInterface.cpp

Chapter 4

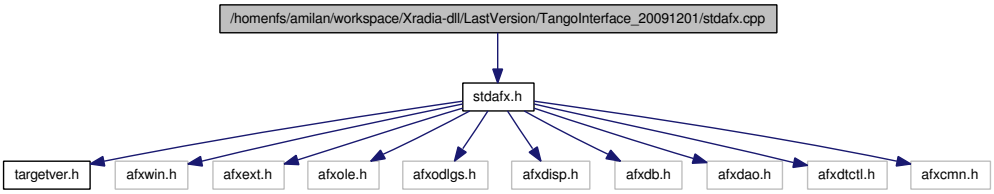
File Documentation

4.1 /homenfs/amilan/workspace/Xradia-dll/LastVersion/TangoInterface_-20091201/Resource.h File Reference

4.2 /homenfs/amilan/workspace/Xradia-dll/LastVersion/TangoInterface_-20091201/stdafx.cpp File Reference

```
#include "stdafx.h"
```

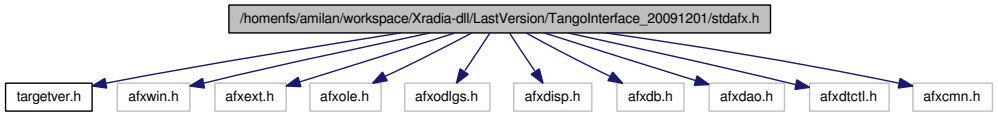
Include dependency graph for stdafx.cpp:



4.3 /homenfs/amilan/workspace/Xradia-dll/LastVersion/TangoInterface_-20091201/stdafx.h File Reference

```
#include "targetver.h"
#include <afxwin.h>
#include <afxext.h>
#include <afxole.h>
#include <afxodlgs.h>
#include <afxdisp.h>
#include <afxdb.h>
#include <afxdao.h>
#include <afxdtctl.h>
#include <afxcmn.h>
```

Include dependency graph for stdafx.h:



This graph shows which files directly or indirectly include this file:

Defines

- #define VC_EXTRALEAN
- #define _ATL_CSTRING_EXPLICIT_CONSTRUCTORS

4.3.1 Define Documentation

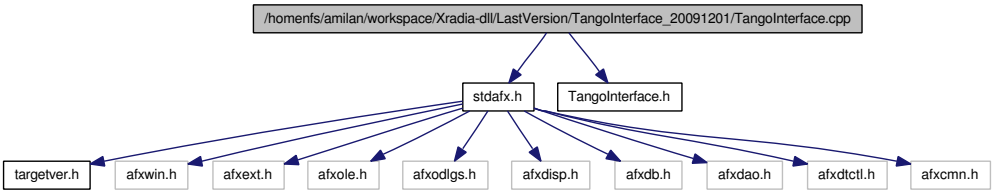
4.3.1.1 #define _ATL_CSTRING_EXPLICIT_CONSTRUCTORS

4.3.1.2 #define VC_EXTRALEAN

4.4 /homenfs/amilan/workspace/Xradia-dll/LastVersion/TangoInterface_-20091201/TangoInterface.cpp File Reference

```
#include "stdafx.h"
#include "TangoInterface.h"
```

Include dependency graph for TangoInterface.cpp:



Defines

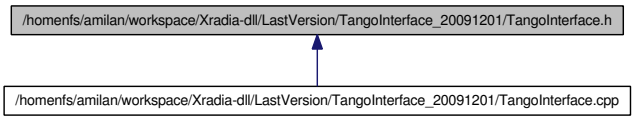
- #define NUM_STRINGS 7

4.4.1 Define Documentation

4.4.1.1 #define NUM_STRINGS 7

4.5 /homenfs/amilan/workspace/Xradia-dll/LastVersion/TangoInterface_-20091201/TangoInterface.h File Reference

This graph shows which files directly or indirectly include this file:



Classes

- class TangoInterface

Enumerations

- enum TANGO_ATTRIBUTES {
  NEG_LIMIT = 0, POS_LIMIT, IN_POS, FOLLOWING_ERROR,
  ENABLED, MOTOR_POSITION, SET_VELO, CURRENT_SIZE_TANGO_ATTRIBUTES_-ENUM }
  Tango interface.

4.5.1 Enumeration Type Documentation

4.5.1.1 enum TANGO_ATTRIBUTES

Tango interface.

One object will be used for contacting all Tango axes. The primary design of this class is to communicate with Tango motors. It also has the facility to communicate with any Tango variable for recording external data.

Units: Assumed in all functions that the units of the passed in values are the same. Xradia software will have a conversion factor that will allow the Xradia software to use Engineering units (ex. mm, deg, eV) independent of the actual hardware. In all descriptions below, the units will be referred to generically as "counts".

Motor Identifier: The motor identifier is a freeform string (configured per motor) that contains all information necessary to connect to a particular hosted motor. The definition of this string is left to the implementor of this interface.

Enumerator:

NEG_LIMIT

POS_LIMIT

IN_POS

FOLLOWING_ERROR

ENABLED

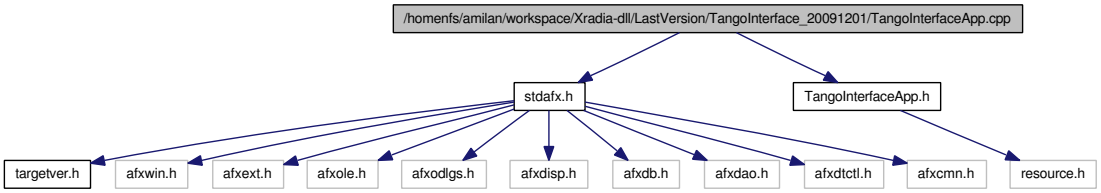
MOTOR_POSITION

SET_VELO
CURRENT_SIZE_TANGO_ATTRIBUTES_ENUM

4.6 /homenfs/amilan/workspace/Xradia-dll/LastVersion/TangoInterface_-20091201/TangoInterfaceApp.cpp File Reference

```
#include "stdafx.h"
#include "TangoInterfaceApp.h"
```

Include dependency graph for TangoInterfaceApp.cpp:



Variables

- CTangoInterfaceApp theApp

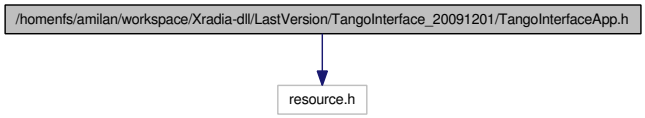
4.6.1 Variable Documentation

4.6.1.1 CTangoInterfaceApp theApp

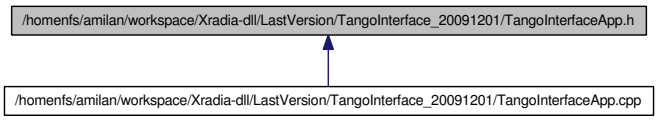
4.7 /homenfs/amilan/workspace/Xradia-dll/LastVersion/TangoInterface_-20091201/TangoInterfaceApp.h File Reference

```
#include "resource.h"
```

Include dependency graph for TangoInterfaceApp.h:



This graph shows which files directly or indirectly include this file:

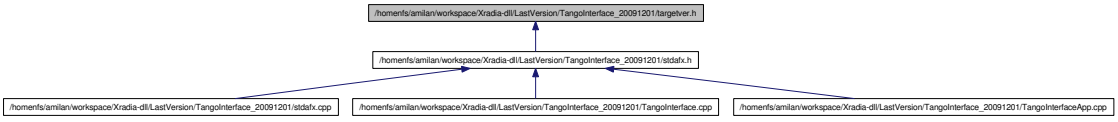


Classes

- class CTangoInterfaceApp

4.8 /homenfs/amilan/workspace/Xradia-dll/LastVersion/TangoInterface_-20091201/targetver.h File Reference

This graph shows which files directly or indirectly include this file:



Defines

- #define WINVER 0x0502
- #define _WIN32_WINNT 0x0502
- #define _WIN32_WINDOWS 0x0410
- #define _WIN32_IE 0x0700

4.8.1 Define Documentation

4.8.1.1 #define _WIN32_IE 0x0700

4.8.1.2 #define _WIN32_WINDOWS 0x0410

4.8.1.3 #define _WIN32_WINNT 0x0502

4.8.1.4 #define WINVER 0x0502

Index