TacoSilvia Reference Manual

0.1

Generated by Doxygen 1.3.5

Mon Jan 12 14:53:37 2004

Contents

Chapter 1

TACOSilvia - Linking Labview and TACO

1.1 Introduction

TACOSiliva is an application that provides an easy-to-use interface between TACO and Labview.

LabVIEW is a powerful graphical programming language from National Instruments that uses icons instead of lines of text to create applications. In contrast to text-based programming languages, where instructions determine program execution, LabVIEW uses dataflow programming, where the flow of data determines execution.

TACO is an object oriented control system developed and used at the ESRF (European Synchrotron Radiation Facility) and FRM2(Forschungs-Reaktor München 2) to control accelerators and beamlines and data acquisition systems.

TACOSiliva is a Qt-application that simplifies remote control of Labview VIs using the TACO client/server communication mechanism, enabling any TACO client to get access to local OR remote virtual instruments running in a special Labview context.

1.2 Installation

In order for TACOSilvia to work you must have Qt and a working TACO environment and the Labview-TACO-Lib developed by Andy Götz (see TACO homepage `http://www.esrf.fr/taco`).

1.2.1 Build the TACO-Labview-Lib:

To build the Taco-Labview-Lib simply execute the Makefile in the labview/server/src/ subdirectory. (Don't forget to adapt the correct Labview installation Path in the Makefile)

1.2.2 Build TACOSilvia:

To build TACOSiliva ensure that all TACO and Qt environment variables are properly set and simply execute the Makefile in the labview/client/ subdirectory. (Don't forget to adapt the correct Labview installation Path in the Makefile)

1.3 Getting Started

Just perform following steps to get the system running:

1.3.1 Start Labview:

Start Labview. Currently only Labview version 7.0 or higher are supported.

1.3.2 Run Device Server:

Open the TacoLabview.llb library from within Labview and select the LVTacoSelectViRemote.vi When you run the VI from within the Panel, the adequate TACO Server is exported automatically

1.3.2.1 Local VI Access:

- In the upper panel choose the directory where your VI is located via FileChooserDialog.
- Press , select your VI from the pulldown-list and run it via button.

1.3.2.2 Remote VI Access:

- Enter the machine name of the remote machine in the server list on the lower panel
- Open the VI Server(Server.vi) on the corresponding machine.
- Insert the desired VI in the export list and run the application.
- After pressing the exported VIs should appear in the list now.
- Select the VI from the list and press *Run*.

1.3.3 Import The Device And Access VI Controls

- Start **TacoSilvia**(p. 20)
- In the **TacoSilvia**(p. 20) GUI just click *Import Device*.

1.3.3.1 List the Controls:

- Retrieve a list of the available controls and indicators by pressing *GetControlList*
- Once listed, you can set and get the values of the controls and indicators in various ways.

1.3.3.2 Get a value (control OR indicator):

- Press the or the button
- Select the corresponding Pull-Down Menu Item
- Press the adequate Toolbar Icon.
- Let the data automatically be retrieved by activating the facility.

1.3.3.3 Set a value (only control):

- Double-click on an entry in the control list and fill in the value field.
- Select the control in the list and press the *SetControlValue* button.
- Select the corresponding Pull-Down Menu Item.
- Press the adequate Toolbar Icon.

1.3.4 Synchronize Data

- If desired, the control values are regularly synchronized with labview.
- Just mark the checkbox and enter the interval at which synchronization should occur.
- The control data is then retrieved automatically at regular intervals.

Chapter 2

TacoSilvia Data Structure Index

2.1 TacoSilvia Data Structures

Here are the data structures with brief descriptions:

Chapter 3

TacoSilvia File Index

3.1 TacoSilvia File List

Here is a list of all files with brief descriptions:

Chapter 4

TacoSilvia Data Structure Documentation

4.1 LabviewClient Class Reference

The **LabviewClient** represents a TACO client and acts as a type-wrapper for the Labview-Taco-Interface.

```
#include <LabviewClient.h>
```

Public Member Functions

- **LabviewClient** ()
- **LabviewClient** (char ∗)
- **LabviewClient** (devserver ∗)
- ∼**LabviewClient** ()
- int **InitDevice** ()
- void **InitVars** ()
- void **TestPutGetError** (char ∗calling_method)
- void **InitGetInputBuffer** ()
- void **InitGetOutputBuffer** (DevString name)
- void **InitSetOutputBuffer** (DevString name, DevString type, DevString value)
- int **GetViInfo** (DevVarStringArray ∗)
- int **GetControlList** (DevVarStringArray ∗lv_cntl_list, bool indicator=false)
- int **GetControlType** (DevString lv_cntl_name, bool indicator=false)
- int **GetControlInfo** (DevString lv_cntl_name, DevVarStringArray ∗lv_cntl_info, bool indicator=false)
- int **GetDoubleValue** (DevString lv_cntl_name, DevDouble ∗lv_cntl_value, bool indicator=false)
- int **GetFloatValue** (DevString lv_cntl_name, DevFloat ∗lv_cntl_value, bool indicator=false)
- int **GetShortValue** (DevString lv_cntl_name, DevShort ∗lv_cntl_value, bool indicator=false)
- int **GetUShortValue** (DevString lv_cntl_name, DevUShort ∗lv_cntl_value, bool indicator=false)

- int **GetLongValue** (DevString lv_cntl_name, DevLong *lv_cntl_value, bool indicator=false)
- int **GetULongValue** (DevString lv_cntl_name, DevULong *lv_cntl_value, bool indicator=false)
- int **GetStringValue** (DevString lv_cntl_name, DevString *lv_cntl_value, bool indicator=false)
- int **GetBooleanValue** (DevString lv_cntl_name, DevBoolean *lv_cntl_value, bool indicator=false)
- int **SetDoubleValue** (DevString lv_cntl_name, DevDouble *lv_cntl_value)
- int **SetFloatValue** (DevString lv_cntl_name, DevFloat *lv_cntl_value)
- int **SetShortValue** (DevString lv_cntl_name, DevShort *lv_cntl_value)
- int **SetUShortValue** (DevString lv_cntl_name, DevUShort *lv_cntl_value)
- int **SetLongValue** (DevString lv_cntl_name, DevLong *lv_cntl_value)
- int **SetULongValue** (DevString lv_cntl_name, DevULong *lv_cntl_value)
- int **SetStringValue** (DevString lv_cntl_name, DevString *lv_cntl_value)
- int **SetBooleanValue** (DevString lv_cntl_name, DevBoolean *lv_cntl_value)

Static Public Member Functions

- DevString **stringType** (int type)
- DevString **stringDType** (int type)
- int **typeFromString** (DevString type_string)
- DevDouble **DevStringToDouble** (DevString string)
- DevFloat **DevStringToFloat** (DevString string)
- DevShort **DevStringToShort** (DevString string)
- DevUShort **DevStringToUShort** (DevString string)
- DevLong **DevStringToLong** (DevString string)
- DevULong **DevStringToULong** (DevString string)
- DevBoolean **DevStringToBoolean** (DevString string)
- DevString **DevDoubleToString** (DevDouble value)
- DevString **DevFloatToString** (DevFloat value)
- DevString **DevShortToString** (DevShort value)
- DevString **DevUShortToString** (DevUShort value)
- DevString **DevLongToString** (DevLong value)
- DevString **DevULongToString** (DevULong value)
- DevString **DevBooleanToString** (DevBoolean value)
- DevVoid * **DevScanToType** (DevString string, int type)
- DevVoid * **DevStringToType** (DevString string, int type)
- DevString **DevTypeToString** (DevVoid *value, int type)

Static Public Attributes

- char * **device_name** = "test/labviewgeneric/1"
- const int **MAX_STRING_LENGTH** = 1000

Protected Attributes

- DevVarStringArray ∗ **cntl_array_in**
- DevVarStringArray ∗ **cntl_array_out**
- int **num_get_in_args**
- int **num_get_out_args**
- int **num_set_out_args**

4.1.1 Detailed Description

The **LabviewClient** represents a TACO client and acts as a type-wrapper for the Labview-Taco-Interface.

LabviewClient provides typespecific *get* and *set* methods. As the Labview-Taco-Interface is designed to be simple and generic, data is always passed as a sequence of strings between Labview and TACO.

This makes it necessary to convert the types on both sides. For reasons of convenience and in order to prevent the user from doing the annoying conversion himself, LabviewClient internally implements some conversion functions and provides a type-specific interface to higher level applications.

Definition at line 49 of file LabviewClient.h.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 LabviewClient::LabviewClient ()

Definition at line 55 of file LabviewClient.cpp.

References InitVars().

4.1.2.2 LabviewClient::LabviewClient (char ∗)

Definition at line 61 of file LabviewClient.cpp.

References InitVars().

4.1.2.3 LabviewClient::LabviewClient (devserver ∗)

Definition at line 76 of file LabviewClient.cpp.

References InitVars().

4.1.2.4 LabviewClient::∼LabviewClient ()

Definition at line 83 of file LabviewClient.cpp.

4.1.3 Member Function Documentation

4.1.3.1 DevString LabviewClient::DevBooleanToString (DevBoolean *value*) [static]

Definition at line 904 of file LabviewClient.cpp.

Referenced by SetBooleanValue().

4.1.3.2 DevString LabviewClient::DevDoubleToString (DevDouble *value*) [static]

Definition at line 864 of file LabviewClient.cpp.

Referenced by SetDoubleValue().

4.1.3.3 DevString LabviewClient::DevFloatToString (DevFloat *value*) [static]

Definition at line 872 of file LabviewClient.cpp.

Referenced by SetFloatValue().

4.1.3.4 DevString LabviewClient::DevLongToString (DevLong *value*) [static]

Definition at line 892 of file LabviewClient.cpp.

Referenced by SetLongValue().

4.1.3.5 DevVoid * LabviewClient::DevScanToType (DevString *string*, int *type*) [static]

Definition at line 1055 of file LabviewClient.cpp.

4.1.3.6 DevString LabviewClient::DevShortToString (DevShort *value*) [static]

Definition at line 880 of file LabviewClient.cpp.

Referenced by SetShortValue().

4.1.3.7 DevBoolean LabviewClient::DevStringToBoolean (DevString *string*) [static]

Definition at line 853 of file LabviewClient.cpp.

Referenced by GetBooleanValue(), main(), TacoSilvia::setControlValue(), and TacoSilvia::setTextControlValue().

4.1.3.8 DevDouble LabviewClient::DevStringToDouble (DevString *string*) [static]

Definition at line 733 of file LabviewClient.cpp.

Referenced by GetDoubleValue(), main(), and TacoSilvia::setTextControlValue().

4.1.3.9 DevFloat LabviewClient::DevStringToFloat (DevString *string*) [static]

Definition at line 758 of file LabviewClient.cpp.

Referenced by GetFloatValue(), main(), TacoSilvia::setControlValue(), and TacoSilvia::setTextControlValue().

4.1.3.10 DevLong LabviewClient::DevStringToLong (DevString *string*) [static]

Definition at line 817 of file LabviewClient.cpp.

Referenced by GetLongValue(), main(), TacoSilvia::setControlValue(), and TacoSilvia::setTextControlValue().

4.1.3.11 DevShort LabviewClient::DevStringToShort (DevString *string*) [static]

Definition at line 781 of file LabviewClient.cpp.

Referenced by GetShortValue(), main(), TacoSilvia::setControlValue(), and TacoSilvia::setTextControlValue().

4.1.3.12 DevVoid * LabviewClient::DevStringToType (DevString *string*, int *type*) [static]

Definition at line 919 of file LabviewClient.cpp.

4.1.3.13 DevULong LabviewClient::DevStringToULong (DevString *string*) [static]

Definition at line 833 of file LabviewClient.cpp.

Referenced by GetULongValue(), main(), TacoSilvia::setControlValue(), and TacoSilvia::setTextControlValue().

4.1.3.14 DevUShort LabviewClient::DevStringToUShort (DevString *string*) [static]

Definition at line 797 of file LabviewClient.cpp.

Referenced by GetUShortValue(), main(), TacoSilvia::setControlValue(), and TacoSilvia::setTextControlValue().

4.1.3.15 DevString LabviewClient::DevTypeToString (DevVoid * *value*, int *type*) [static]

Definition at line 1133 of file LabviewClient.cpp.

4.1.3.16 DevString LabviewClient::DevULongToString (DevULong *value*) [static]

Definition at line 898 of file LabviewClient.cpp.

Referenced by SetULongValue().

4.1.3.17 DevString LabviewClient::DevUShortToString (DevUShort *value*) [static]

Definition at line 886 of file LabviewClient.cpp.

Referenced by SetUShortValue().

4.1.3.18 int LabviewClient::GetBooleanValue (DevString *lv_cntl_name*, DevBoolean ∗ *lv_cntl_value*, bool *indicator* = false)

Definition at line 474 of file LabviewClient.cpp.

References cntl_array_in, cntl_array_out, DevStringToBoolean(), InitGetInputBuffer(), InitGetOutputBuffer(), and TestPutGetError().

Referenced by TacoSilvia::getControlList(), TacoSilvia::getControlValue(), TacoSilvia::getIndicatorList(), TacoSilvia::getIndicatorValue(), and main().

4.1.3.19 int LabviewClient::GetControlInfo (DevString *lv_cntl_name*, DevVarStringArray ∗ *lv_cntl_info*, bool *indicator* = false)

Definition at line 285 of file LabviewClient.cpp.

References cntl_array_in, InitGetOutputBuffer(), and TestPutGetError().

Referenced by GetControlType(), and main().

4.1.3.20 int LabviewClient::GetControlList (DevVarStringArray ∗ *lv_cntl_list*, bool *indicator* = false)

Definition at line 273 of file LabviewClient.cpp.

References TestPutGetError().

Referenced by TacoSilvia::getControlList(), TacoSilvia::getIndicatorList(), and main().

4.1.3.21 int LabviewClient::GetControlType (DevString *lv_cntl_name*, bool *indicator* = false)

Definition at line 220 of file LabviewClient.cpp.

References GetControlInfo().

Referenced by TacoSilvia::getControlList(), TacoSilvia::getIndicatorList(), and main().

4.1.3.22 int LabviewClient::GetDoubleValue (DevString *lv_cntl_name*, DevDouble ∗ *lv_cntl_value*, bool *indicator* = false)

Definition at line 313 of file LabviewClient.cpp.

References cntl_array_in, cntl_array_out, DevStringToDouble(), InitGetInputBuffer(), InitGetOutputBuffer(), and TestPutGetError().

Referenced by TacoSilvia::getControlList(), TacoSilvia::getControlValue(), TacoSilvia::getIndicatorList(), TacoSilvia::getIndicatorValue(), and main().

4.1.3.23 int LabviewClient::GetFloatValue (DevString *lv_cntl_name*, DevFloat * *lv_cntl_value*, bool *indicator* = false)

Definition at line 336 of file LabviewClient.cpp.

References cntl_array_in, cntl_array_out, DevStringToFloat(), InitGetInputBuffer(), InitGetOutputBuffer(), and TestPutGetError().

Referenced by TacoSilvia::getControlList(), TacoSilvia::getControlValue(), TacoSilvia::getIndicatorList(), TacoSilvia::getIndicatorValue(), and main().

4.1.3.24 int LabviewClient::GetLongValue (DevString *lv_cntl_name*, DevLong * *lv_cntl_value*, bool *indicator* = false)

Definition at line 405 of file LabviewClient.cpp.

References cntl_array_in, cntl_array_out, DevStringToLong(), InitGetInputBuffer(), InitGetOutputBuffer(), and TestPutGetError().

Referenced by TacoSilvia::getControlList(), TacoSilvia::getControlValue(), TacoSilvia::getIndicatorList(), TacoSilvia::getIndicatorValue(), and main().

4.1.3.25 int LabviewClient::GetShortValue (DevString *lv_cntl_name*, DevShort * *lv_cntl_value*, bool *indicator* = false)

Definition at line 359 of file LabviewClient.cpp.

References cntl_array_in, cntl_array_out, DevStringToShort(), InitGetInputBuffer(), InitGetOutputBuffer(), and TestPutGetError().

Referenced by TacoSilvia::getControlList(), TacoSilvia::getControlValue(), TacoSilvia::getIndicatorList(), TacoSilvia::getIndicatorValue(), and main().

4.1.3.26 int LabviewClient::GetStringValue (DevString *lv_cntl_name*, DevString * *lv_cntl_value*, bool *indicator* = false)

Definition at line 451 of file LabviewClient.cpp.

References cntl_array_in, cntl_array_out, InitGetInputBuffer(), InitGetOutputBuffer(), and TestPutGetError().

Referenced by TacoSilvia::getControlList(), TacoSilvia::getControlValue(), TacoSilvia::getIndicatorList(), TacoSilvia::getIndicatorValue(), and main().

4.1.3.27 int LabviewClient::GetULongValue (DevString *lv_cntl_name*, DevULong * *lv_cntl_value*, bool *indicator* = false)

Definition at line 428 of file LabviewClient.cpp.

References cntl_array_in, cntl_array_out, DevStringToULong(), InitGetInputBuffer(), InitGetOutputBuffer(), and TestPutGetError().

Referenced by TacoSilvia::getControlList(), TacoSilvia::getControlValue(), TacoSilvia::getIndicatorList(), TacoSilvia::getIndicatorValue(), and main().

4.1.3.28 int LabviewClient::GetUShortValue (DevString *lv_cntl_name*, DevUShort * *lv_cntl_value*, bool *indicator* = false)

Definition at line 382 of file LabviewClient.cpp.

References cntl_array_in, cntl_array_out, DevStringToUShort(), InitGetInputBuffer(), InitGetOutputBuffer(), and TestPutGetError().

Referenced by TacoSilvia::getControlList(), TacoSilvia::getControlValue(), TacoSilvia::getIndicatorList(), TacoSilvia::getIndicatorValue(), and main().

4.1.3.29 int LabviewClient::GetViInfo (DevVarStringArray *)

Definition at line 262 of file LabviewClient.cpp.

References TestPutGetError().

Referenced by TacoSilvia::getViInfo(), and main().

4.1.3.30 int LabviewClient::InitDevice ()

Definition at line 90 of file LabviewClient.cpp.

References device_name.

Referenced by TacoSilvia::importDevice(), and main().

4.1.3.31 void LabviewClient::InitGetInputBuffer ()

Definition at line 203 of file LabviewClient.cpp.

References cntl_array_out.

Referenced by GetBooleanValue(), GetDoubleValue(), GetFloatValue(), GetLongValue(), GetShortValue(), GetStringValue(), GetULongValue(), and GetUShortValue().

4.1.3.32 void LabviewClient::InitGetOutputBuffer (DevString *name*)

Definition at line 189 of file LabviewClient.cpp.

References cntl_array_in, and num_get_out_args.

Referenced by GetBooleanValue(), GetControlInfo(), GetDoubleValue(), GetFloatValue(), GetLongValue(), GetShortValue(), GetStringValue(), GetULongValue(), and GetUShortValue().

4.1.3.33 void LabviewClient::InitSetOutputBuffer (DevString *name*, DevString *type*, DevString *value*)

Definition at line 166 of file LabviewClient.cpp.

References cntl_array_in, and num_set_out_args.

Referenced by SetBooleanValue(), SetDoubleValue(), SetFloatValue(), SetLongValue(), SetShortValue(), SetStringValue(), SetULongValue(), and SetUShortValue().

4.1.3.34 void LabviewClient::InitVars ()

Definition at line 158 of file LabviewClient.cpp.

References cntl_array_in, cntl_array_out, num_get_in_args, num_get_out_args, and num_set_out_args.

Referenced by LabviewClient().

4.1.3.35 int LabviewClient::SetBooleanValue (DevString *lv_cntl_name*, DevBoolean * *lv_cntl_value*)

Definition at line 628 of file LabviewClient.cpp.

References cntl_array_in, DevBooleanToString(), InitSetOutputBuffer(), and TestPutGetError().

Referenced by main(), TacoSilvia::setControlValue(), and TacoSilvia::setTextControlValue().

4.1.3.36 int LabviewClient::SetDoubleValue (DevString *lv_cntl_name*, DevDouble * *lv_cntl_value*)

Definition at line 507 of file LabviewClient.cpp.

References cntl_array_in, DevDoubleToString(), InitSetOutputBuffer(), and TestPutGetError().

Referenced by main(), TacoSilvia::setControlValue(), and TacoSilvia::setTextControlValue().

4.1.3.37 int LabviewClient::SetFloatValue (DevString *lv_cntl_name*, DevFloat * *lv_cntl_value*)

Definition at line 525 of file LabviewClient.cpp.

References cntl_array_in, DevFloatToString(), InitSetOutputBuffer(), and TestPutGetError().

Referenced by main(), TacoSilvia::setControlValue(), and TacoSilvia::setTextControlValue().

4.1.3.38 int LabviewClient::SetLongValue (DevString *lv_cntl_name*, DevLong * *lv_cntl_value*)

Definition at line 576 of file LabviewClient.cpp.

References cntl_array_in, DevLongToString(), InitSetOutputBuffer(), and TestPutGetError().

Referenced by main(), TacoSilvia::setControlValue(), and TacoSilvia::setTextControlValue().

4.1.3.39 int LabviewClient::SetShortValue (DevString *lv_cntl_name*, DevShort * *lv_cntl_value*)

Definition at line 543 of file LabviewClient.cpp.

References cntl_array_in, DevShortToString(), InitSetOutputBuffer(), and TestPutGetError().

Referenced by main(), TacoSilvia::setControlValue(), and TacoSilvia::setTextControlValue().

4.1.3.40 int LabviewClient::SetStringValue (DevString *lv_cntl_name*, DevString ∗ *lv_cntl_value*)

Definition at line 612 of file LabviewClient.cpp.

References cntl_array_in, InitSetOutputBuffer(), and TestPutGetError().

Referenced by main(), TacoSilvia::setControlValue(), and TacoSilvia::setTextControlValue().

4.1.3.41 int LabviewClient::SetULongValue (DevString *lv_cntl_name*, DevULong ∗ *lv_cntl_value*)

Definition at line 594 of file LabviewClient.cpp.

References cntl_array_in, DevULongToString(), InitSetOutputBuffer(), and TestPutGetError().

Referenced by main(), TacoSilvia::setControlValue(), and TacoSilvia::setTextControlValue().

4.1.3.42 int LabviewClient::SetUShortValue (DevString *lv_cntl_name*, DevUShort ∗ *lv_cntl_value*)

Definition at line 558 of file LabviewClient.cpp.

References cntl_array_in, DevUShortToString(), InitSetOutputBuffer(), and TestPutGetError().

Referenced by main(), TacoSilvia::setControlValue(), and TacoSilvia::setTextControlValue().

4.1.3.43 DevString LabviewClient::stringDType (int *type*) `[static]`

Definition at line 690 of file LabviewClient.cpp.

Referenced by TacoSilvia::getControlList(), and TacoSilvia::getIndicatorList().

4.1.3.44 DevString LabviewClient::stringType (int *type*) `[static]`

Definition at line 648 of file LabviewClient.cpp.

4.1.3.45 void LabviewClient::TestPutGetError (char ∗ *calling_method*)

Definition at line 211 of file LabviewClient.cpp.

Referenced by GetBooleanValue(), GetControlInfo(), GetControlList(), GetDoubleValue(), GetFloatValue(), GetLongValue(), GetShortValue(), GetStringValue(), GetULongValue(), GetUShortValue(), GetViInfo(), SetBooleanValue(), SetDoubleValue(), SetFloatValue(), SetLongValue(), SetShortValue(), SetStringValue(), SetULongValue(), and SetUShortValue().

4.1.3.46 int LabviewClient::typeFromString (DevString *type_string*) `[static]`

Definition at line 1192 of file LabviewClient.cpp.

Referenced by TacoSilvia::getControlValue(), TacoSilvia::getIndicatorValue(), and TacoSilvia::setControlValue().

4.1.4 Field Documentation

4.1.4.1 DevVarStringArray* LabviewClient::cntl_array_in [protected]

Definition at line 60 of file LabviewClient.h.

Referenced by GetBooleanValue(), GetControlInfo(), GetDoubleValue(), GetFloatValue(), GetLongValue(), GetShortValue(), GetStringValue(), GetULongValue(), GetUShortValue(), InitGetOutputBuffer(), InitSetOutputBuffer(), InitVars(), SetBooleanValue(), SetDoubleValue(), SetFloatValue(), SetLongValue(), SetShortValue(), SetStringValue(), SetULongValue(), and SetUShortValue().

4.1.4.2 DevVarStringArray* LabviewClient::cntl_array_out [protected]

Definition at line 61 of file LabviewClient.h.

Referenced by GetBooleanValue(), GetDoubleValue(), GetFloatValue(), GetLongValue(), GetShortValue(), GetStringValue(), GetULongValue(), GetUShortValue(), InitGetInputBuffer(), and InitVars().

4.1.4.3 char * LabviewClient::device_name = "test/labviewgeneric/1" [static]

Definition at line 51 of file LabviewClient.cpp.

Referenced by InitDevice().

4.1.4.4 const int LabviewClient::MAX_STRING_LENGTH = 1000 [static]

Definition at line 70 of file LabviewClient.h.

4.1.4.5 int LabviewClient::num_get_in_args [protected]

Definition at line 62 of file LabviewClient.h.

Referenced by InitVars().

4.1.4.6 int LabviewClient::num_get_out_args [protected]

Definition at line 63 of file LabviewClient.h.

Referenced by InitGetOutputBuffer(), and InitVars().

4.1.4.7 int LabviewClient::num_set_out_args [protected]

Definition at line 64 of file LabviewClient.h.

Referenced by InitSetOutputBuffer(), and InitVars().

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

- **LabviewClient.h**
- **LabviewClient.cpp**

4.2 TacoSilvia Class Reference

TacoSilvia means **TACO S**erver **I**nterfacing **L**abview **V**irtual **I**nstrument **A**pplications. It represents a GUI application built with Qt and allows communication with arbitrary Labview Virtual Instruments(VIs) by means of getting and setting the control and indicator values of selected VIs. It uses TACO and the Labview-Taco-Lib to offer an interface to Labview applications.

`#include <TacoSilvia.h>`

Public Slots

- void **importDevice** ()

  Initializes the **LabviewClient**(p. 9) *which encapsulates the communication with Labview.*

- void **closeDevice** ()

  Frees the TACO device and erases all content from widgets.

- void **exit** ()

  Terminates the application.

- void **getViInfo** ()

  Retrieves a list of VI related informations and displays them in a modal dialog.

- void **getControlList** ()

  Retrieves a list of all controls using the **LabviewClient**(p. 9) *interface.*

- void **getIndicatorList** ()

  Retrieves a list of all indicators using the **LabviewClient**(p. 9) *interface and populates the corresponding table widget (same as above, now for the indicators).*

- void **getControlValue** ()

  getControlValue()(p. 26) *retrieves the value of the labview control using the* **LabviewClient**(p. 9) *interface method*

- void **setControlValue** ()

  Asks the user to input a value of the adequate type (corresponding to the selected control, and forwards it to the labview control using the **LabviewClient**(p. 9) *interface.*

- void **setTextControlValue** (int row, int col)

  setTextControlValue()(p. 31) *is called when the user has edited a table item on-the-fly by double-clicking it*

- void **setIndUpdateValue** (int row, int col)

  setIndUpdateValue()(p. 31) *is called when the user changes the update flag by clicking on it*

- void **getIndicatorValue** ()

  Retrieves the value of the labview indicator using the **LabviewClient**(p. 9) *interface method.*

- void **instructions** ()

  Displays a window with instructions how to use the client and server.

- void **about** ()

  Displays the about window.

- void **showTableContextMenu** (int row, int col, const QPoint &pos)

  showContextMenu() is used to display a context menu when the user clicks on a control in the QTable.

- void **showIndTableContextMenu** (int row, int col, const QPoint &pos)

  showIndTableContextMenu()(p. 32) *is used to display a context menu when the user clicks on an indicator in the QTable*

- void **clickTableItem** (int row, int col, int button, const QPoint &mousePos)

  Called when user double clicks on the table.

- void **updateControlsActivated** (int **state**)

  Called whenever the check-button changes state.

- void **updateIndicatorsActivated** (int **state**)

  Called whenever the check-button changes state.

- void **intervalControlsChanged** (int value)

  Handles timer management.

- void **intervalIndicatorsChanged** (int value)

  Handles timer management.

- void **deviceImportClose** ()

  Adapts the import button to the current state.

Public Member Functions

- **TacoSilvia** (QWidget *parent=0, const char *name=0, WFlags fl=WType_TopLevel)

  The constructor initializes all the widgets and displays them.

- ∼**TacoSilvia** ()

  brief Destroys the object and frees any allocated resources

- **LabviewClient** * **getLabviewClient** ()

  Grants access to the **LabviewClient**(p. 9) *object.*

- void **setNotImportedState** ()

  Grants access to the state machine.

Data Fields

- QAction * **deviceImportAction**
- QAction * **deviceCloseAction**

- QAction * **deviceExitAction**
- QAction * **commandsViInfoAction**
- QAction * **commandsControlListAction**
- QAction * **commandsIndicatorListAction**
- QAction * **commandsGetValueAction**
- QAction * **commandsSetValueAction**
- QAction * **commandsGetIndValueAction**
- QAction * **helpInstructionsAction**
- QAction * **helpAboutAction**

Protected Slots

- virtual void **languageChange** ()

  Sets the strings of the subwidgets using the current language.

Protected Member Functions

- void **initPalettes** ()

  Defines a customized color palette for the widgets.

- void **initIcons** ()

  * *Creates the Icons from ASCII bitmaps.*

- void **initActions** ()

  Defines the Actions and assigns icons to them.

- void **initWindow** (const char *name=0)

  Initializes the main window of the GUI.

- void **initMenus** ()

  Constructs menus and assign the corresponding actions.

- void **initTables** ()

  Creates and initializes the central QTable (actually a subclassed QTable) widgets that hold the controls and indicators.

- void **initWidgets** ()

  Initializes all remaining widgets such as buttons, labels, checkboxes etc.

- void **addWidgets** ()

  Arranges the widgets in the main window.

- void **initConnections** ()

  Realizes the signals and slots connections between the actions and widgets.

- void **setColors** ()

  Assigns the custom palette to the widgets.

- int **stateMachine** (int state_flags)

  This function checks in what kind of state the application finds itself and adapts the widgets correspondigly.

Protected Attributes

- QFrame ∗ **frame**
- QGridLayout ∗ **g**
- QMenuBar ∗ **menubar**
- QToolBar ∗ **toolbar**
- QPopupMenu ∗ **devicemenu**
- QPopupMenu ∗ **commandsmenu**
- QPopupMenu ∗ **helpmenu**
- QListBox ∗ **lbox**
- QListView ∗ **lview**
- **QControlsTable** ∗ **table**
- QTable ∗ **indtable**
- QLabel ∗ **label**
- QLabel ∗ **controls_label**
- QLabel ∗ **indicators_label**
- QLabel ∗ **statusLabel**
- QLabel ∗ **statusMessage**
- QSpinBox ∗ **interval_spinbox**
- QCheckBox ∗ **check_update_box**
- QSpinBox ∗ **ind_interval_spinbox**
- QCheckBox ∗ **ind_check_update_box**
- QPushButton ∗ **bimport**
- QPushButton ∗ **binfo**
- QPushButton ∗ **blist**
- QPushButton ∗ **bilist**
- QPushButton ∗ **bgetv**
- QPushButton ∗ **bsetv**
- QPushButton ∗ **bgetiv**
- QErrorMessage ∗ **errmsg**
- **QControlTip** ∗ **t**
- QTextEdit ∗ **vinfo**
- **LabviewClient** ∗ **lv_client**
- QHBoxLayout ∗ **hbl0**
- QHBoxLayout ∗ **hbl1**
- QHBoxLayout ∗ **hbl2**
- QHBoxLayout ∗ **hbl3**
- QHBoxLayout ∗ **hbl4**
- QHBoxLayout ∗ **hbl5**
- QPalette **palette**
- QPixmap **appicon**
- QPixmap **p_device_import**
- QPixmap **p_device_close**
- QPixmap **p_vi_info**
- QPixmap **p_list_controls**

- QPixmap **p_list_indicators**
- QPixmap **p_get_value**
- QPixmap **p_set_value**
- QPixmap **p_get_ind_value**
- QPixmap **p_about**
- QPixmap **p_instructions**
- int **state**

Static Protected Attributes

- int **STATE_NOTIMPORTED** = 0
- int **STATE_IMPORTED** = 1
- int **STATE_CONTROLSLISTED** = 2
- int **STATE_INDICATORSLISTED** = 3

4.2.1 Detailed Description

TacoSilvia means **TACO** Server **I**nterfacing **L**abview **V**irtual **I**nstrument **A**pplications. It represents a GUI application built with Qt and allows communication with arbitrary Labview Virtual Instruments(VIs) by means of getting and setting the control and indicator values of selected VIs. It uses TACO and the Labview-Taco-Lib to offer an interface to Labview applications.

The Labview-Taco-Lib is strongly tied to a corresponding particular Labview VI that itself loads the VI to be controlled remotely and it is able to handle TACO client requests in Labview context. In that context, this pair acts as the TACO Device Server whereas another class is responsible for the client side: **LabviewClient**(p. 9). The **LabviewClient**(p. 9) class encapsulates the communication with labview via TACO and does all the type conversion and checking. TacoSilvia just delegates the requests to that class in order to contact labview. Once a connection has been established (the device server has been imported) the controls (and the indicators too) of the selected virtual instrument can be listed and modified using the GUI functionality.

Definition at line 130 of file TacoSilvia.h.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 TacoSilvia::TacoSilvia (QWidget ∗ *parent* = 0, const char ∗ *name* = 0, WFlags *fl* = WType_TopLevel)

The constructor initializes all the widgets and displays them.

A QGridLayout class is used for the arranging and alignment of the widgets.

Definition at line 73 of file TacoSilvia.cpp.

References addWidgets(), initActions(), initConnections(), initIcons(), initMenus(), initPalettes(), initTables(), initWidgets(), initWindow(), languageChange(), setColors(), STATE_NOTIMPORTED, and stateMachine().

4.2.2.2 TacoSilvia::∼TacoSilvia ()

brief Destroys the object and frees any allocated resources

Definition at line 500 of file TacoSilvia.cpp.

4.2.3 Member Function Documentation

4.2.3.1 void TacoSilvia::about () `[slot]`

Displays the about window.

Shows some version info and the TACO Silvia Icon.

Definition at line 1285 of file TacoSilvia.cpp.

References appicon.

Referenced by initConnections().

4.2.3.2 void TacoSilvia::addWidgets () `[protected]`

Arranges the widgets in the main window.

TODO: detailed description

Definition at line 410 of file TacoSilvia.cpp.

References controls_label, g, hbl0, hbl2, hbl3, hbl4, hbl5, indicators_label, indtable, and table.

Referenced by TacoSilvia().

4.2.3.3 void TacoSilvia::clickTableItem (int *row*, int *col*, int *button*, const QPoint & *mousePos*) `[slot]`

Called when user double clicks on the table.

Invokes customized table editing and enables the use of subclassed QTableItems.

Definition at line 1975 of file TacoSilvia.cpp.

References table.

Referenced by initConnections().

4.2.3.4 void TacoSilvia::closeDevice () `[slot]`

Frees the TACO device and erases all content from widgets.

Invokes a call to **stateMachine()**(p. 32) to turn all widgets disabled.

Definition at line 893 of file TacoSilvia.cpp.

References indtable, lv_client, STATE_NOTIMPORTED, stateMachine(), and table.

Referenced by deviceImportClose(), exit(), and initConnections().

4.2.3.5 void TacoSilvia::deviceImportClose () `[slot]`

Adapts the import button to the current state.

As only one button is used for importing and exporting the device, the label of the button changes in dependancy of the current state.

Definition at line 2087 of file TacoSilvia.cpp.

References bimport, closeDevice(), importDevice(), state, and STATE_IMPORTED.

Referenced by initConnections().

4.2.3.6 void TacoSilvia::exit () [slot]

Terminates the application.

Shuts down the TACO device before closing the main window.

Definition at line 1344 of file TacoSilvia.cpp.

References closeDevice().

Referenced by initConnections().

4.2.3.7 void TacoSilvia::getControlList () [slot]

Retrieves a list of all controls using the **LabviewClient**(p. 9) interface.

Then the type of each control is retrieved sequentially so that the control may be displayed properly in the table (with the right icon and the right table item) As the QTable model doesn't allow indexing by caption names, we always have to check the header captions sequentially in order to find the correct position for the insertion of the corresponding item entry. For this purpose and in order to keep things consistant, the table captions are generalized and kept as static members. Before the table may be populated with vi data, we have to figure out the indices for the name, value, type field in the coresponding QTable

Definition at line 1003 of file TacoSilvia.cpp.

References LabviewClient::GetBooleanValue(), LabviewClient::GetControlList(), LabviewClient::GetControlType(), LabviewClient::GetDoubleValue(), LabviewClient::GetFloatValue(), LabviewClient::GetLongValue(), LabviewClient::GetShortValue(), LabviewClient::GetStringValue(), LabviewClient::GetULongValue(), LabviewClient::GetUShortValue(), lv_client, state, STATE_CONTROLSLISTED, STATE_IMPORTED, STATE_NOTIMPORTED, stateMachine(), LabviewClient::stringDType(), and table.

Referenced by initConnections().

4.2.3.8 void TacoSilvia::getControlValue () [slot]

getControlValue()(p. 26) retrieves the value of the labview control using the **LabviewClient**(p. 9) interface method

As the QTable model doesn't allow indexing by caption names, we always have to check the header captions sequentially in order to find the correct position for the insertion of the corresponding item entry. For this purpose and in order to keep things consistant, the table captions are generalized and kept as static members. Before the table may be populated with vi data, we have to figure out the indices for the name, value, type field in the coresponding QTable

Definition at line 1483 of file TacoSilvia.cpp.

References LabviewClient::GetBooleanValue(), LabviewClient::GetDoubleValue(), LabviewClient::GetFloatValue(), LabviewClient::GetLongValue(), LabviewClient::GetShortValue(), LabviewClient::GetStringValue(), LabviewClient::GetULongValue(), LabviewClient::GetUShortValue(), lv_client, state, STATE_CONTROLSLISTED, STATE_IMPORTED, STATE_NOTIMPORTED, stateMachine(), table, and LabviewClient::typeFromString().

Referenced by initConnections(), and showTableContextMenu().

4.2.3.9 void TacoSilvia::getIndicatorList () [slot]

Retrieves a list of all indicators using the **LabviewClient**(p. 9) interface and populates the corresponding table widget (same as above, now for the indicators).

As the QTable model doesn't allow indexing by caption names, the header captions are scanned sequentially in order to find the correct position for the insertion of the corresponding item entry. For this purpose and in order to keep things maintainable, the table captions are generalized and kept as static members. Before the table may be populated with vi data, the indices for the name, value, type fields have to be found in the coresponding QTable

Definition at line 1149 of file TacoSilvia.cpp.

References LabviewClient::GetBooleanValue(), LabviewClient::GetControlList(), LabviewClient::GetControlType(), LabviewClient::GetDoubleValue(), LabviewClient::GetFloatValue(), LabviewClient::GetLongValue(), LabviewClient::GetShortValue(), LabviewClient::GetStringValue(), LabviewClient::GetULongValue(), LabviewClient::GetUShortValue(), indtable, lv_client, STATE_INDICATORSLISTED, STATE_NOTIMPORTED, stateMachine(), LabviewClient::stringDType(), and table.

Referenced by initConnections().

4.2.3.10 void TacoSilvia::getIndicatorValue () [slot]

Retrieves the value of the labview indicator using the **LabviewClient**(p. 9) interface method.

As the QTable model doesn't allow indexing by caption names, we always have to check the header captions sequentially in order to find the correct position for the insertion of the corresponding item entry. For this purpose and in order to keep things consistant, the table captions are generalized and kept as static members. Before the table may be populated with vi data, we have to figure out the indices for the name, value, type field in the coresponding QTable

Definition at line 1361 of file TacoSilvia.cpp.

References LabviewClient::GetBooleanValue(), LabviewClient::GetDoubleValue(), LabviewClient::GetFloatValue(), LabviewClient::GetLongValue(), LabviewClient::GetShortValue(), LabviewClient::GetStringValue(), LabviewClient::GetULongValue(), LabviewClient::GetUShortValue(), indtable, lv_client, state, STATE_IMPORTED, STATE_INDICATORSLISTED, STATE_NOTIMPORTED, stateMachine(), and LabviewClient::typeFromString().

Referenced by initConnections(), and showIndTableContextMenu().

4.2.3.11 LabviewClient ∗ TacoSilvia::getLabviewClient ()

Grants access to the **LabviewClient**(p. 9) object.

Just returns the **LabviewClient**(p. 9) member object.

Definition at line 2055 of file TacoSilvia.cpp.

References lv_client.

4.2.3.12 void TacoSilvia::getViInfo () [slot]

Retrieves a list of VI related informations and displays them in a modal dialog.

The List is made up of strings each of which contains a param/value pair.

In order to highlight it with different styles in the QTextEdit they have to be splitted

Definition at line 931 of file TacoSilvia.cpp.

References LabviewClient::GetViInfo(), lv_client, state, and STATE_IMPORTED.

Referenced by initConnections().

4.2.3.13 void TacoSilvia::importDevice () [slot]

Initializes the **LabviewClient**(p. 9) which encapsulates the communication with Labview.

Definition at line 873 of file TacoSilvia.cpp.

References LabviewClient::InitDevice(), lv_client, STATE_IMPORTED, and stateMachine().

Referenced by deviceImportClose(), and initConnections().

4.2.3.14 void TacoSilvia::initActions () [protected]

Defines the Actions and assigns icons to them.

TODO: detailed description

Definition at line 160 of file TacoSilvia.cpp.

References commandsControlListAction, commandsGetIndValueAction, commandsGetValueAction, commandsIndicatorListAction, commandsSetValueAction, commandsViInfoAction, deviceCloseAction, deviceExitAction, deviceImportAction, helpAboutAction, helpInstructionsAction, p_about, p_device_close, p_device_import, p_get_ind_value, p_get_value, p_instructions, p_list_controls, p_list_indicators, p_set_value, and p_vi_info.

Referenced by TacoSilvia().

4.2.3.15 void TacoSilvia::initConnections () [protected]

Realizes the signals and slots connections between the actions and widgets.

Definition at line 432 of file TacoSilvia.cpp.

References about(), bgetiv, bgetv, bilist, bimport, binfo, blist, bsetv, check_update_box, clickTableItem(), closeDevice(), commandsControlListAction, commandsGetIndValueAction, commandsGetValueAction, commandsIndicatorListAction, commandsSetValueAction, commandsViInfoAction, deviceCloseAction, deviceExitAction, deviceImportAction, deviceImportClose(), exit(), getControlList(), getControlValue(), getIndicatorList(), getIndicatorValue(), getViInfo(), helpAboutAction, helpInstructionsAction, importDevice(), ind_check_update_box, ind_interval_spinbox, indtable, instructions(), interval_spinbox, intervalControlsChanged(), intervalIndicatorsChanged(), setControlValue(), setIndUpdateValue(), setTextControlValue(), showIndTableContextMenu(), showTableContextMenu(), table, updateControlsActivated(), and updateIndicatorsActivated().

Referenced by TacoSilvia().

4.2.3.16 void TacoSilvia::initIcons () [protected]

∗ Creates the Icons from ASCII bitmaps.

TODO: detailed description

Definition at line 127 of file TacoSilvia.cpp.

References p_about, p_device_close, p_device_import, p_get_ind_value, p_get_value, p_instructions, p_list_controls, p_list_indicators, p_set_value, and p_vi_info.

Referenced by TacoSilvia().

4.2.3.17 void TacoSilvia::initMenus () [protected]

Constructs menus and assign the corresponding actions.

TODO: detailed description

Definition at line 227 of file TacoSilvia.cpp.

References commandsControlListAction, commandsGetIndValueAction, commandsGetValueAction, commandsIndicatorListAction, commandsmenu, commandsSetValueAction, commandsViInfoAction, deviceCloseAction, deviceExitAction, deviceImportAction, devicemenu, helpAboutAction, helpInstructionsAction, helpmenu, menubar, and toolbar.

Referenced by TacoSilvia().

4.2.3.18 void TacoSilvia::initPalettes () [protected]

Defines a customized color palette for the widgets.

TODO: detailed description

Definition at line 94 of file TacoSilvia.cpp.

References palette.

Referenced by TacoSilvia().

4.2.3.19 void TacoSilvia::initTables () [protected]

Creates and initializes the central QTable (actually a subclassed QTable) widgets that hold the controls and indicators.

TODO: detailed description

Definition at line 288 of file TacoSilvia.cpp.

References frame, indtable, t, and table.

Referenced by TacoSilvia().

4.2.3.20 void TacoSilvia::initWidgets () [protected]

Initializes all remaining widgets such as buttons, labels, checkboxes etc.

TODO: detailed description

Definition at line 350 of file TacoSilvia.cpp.

References bgetiv, bgetv, bilist, bimport, binfo, blist, bsetv, check_update_box, controls_label, frame, hbl0, hbl1, hbl2, hbl3, hbl4, hbl5, ind_check_update_box, ind_interval_spinbox, indicators_label, interval_spinbox, statusLabel, and statusMessage.

Referenced by TacoSilvia().

4.2.3.21 void TacoSilvia::initWindow (const char ∗ *name* = 0) [protected]

Initializes the main window of the GUI.

Sets the name of the application, creates the layout and initializes the window size.

Definition at line 195 of file TacoSilvia.cpp.

References appicon, errmsg, frame, g, lv_client, and palette.

Referenced by TacoSilvia().

4.2.3.22 void TacoSilvia::instructions () [slot]

Displays a window with instructions how to use the client and server.

Provides the user with some infos of what it is and what you can do with it and how you should do it.

Definition at line 1298 of file TacoSilvia.cpp.

Referenced by initConnections().

4.2.3.23 void TacoSilvia::intervalControlsChanged (int *value*) [slot]

Handles timer management.

Whenever the interval is altered, the timer has to be stopped and invoked with the recent values.

Definition at line 2063 of file TacoSilvia.cpp.

References check_update_box, and table.

Referenced by initConnections().

4.2.3.24 void TacoSilvia::intervalIndicatorsChanged (int *value*) [slot]

Handles timer management.

Whenever the interval is altered, the timer has to be stopped and must be invoked with the recent parameters.

Definition at line 2074 of file TacoSilvia.cpp.

References ind_check_update_box, and indtable.

Referenced by initConnections().

4.2.3.25 void TacoSilvia::languageChange () [protected, virtual, slot]

Sets the strings of the subwidgets using the current language.

Definition at line 508 of file TacoSilvia.cpp.

References commandsControlListAction, commandsGetIndValueAction, commandsGetValueAction, commandsIndicatorListAction, commandsSetValueAction, commandsViInfoAction, deviceCloseAction, deviceExitAction, deviceImportAction, helpAboutAction, helpInstructionsAction, and toolbar.

Referenced by TacoSilvia().

4.2.3.26 void TacoSilvia::setColors () [protected]

Assigns the custom palette to the widgets.

Definition at line 473 of file TacoSilvia.cpp.

References bgetiv, bgetv, bilist, bimport, binfo, blist, bsetv, check_update_box, commandsmenu, devicemenu, helpmenu, ind_check_update_box, ind_interval_spinbox, interval_spinbox, menubar, palette, statusMessage, and toolbar.

Referenced by TacoSilvia().

4.2.3.27 void TacoSilvia::setControlValue () [slot]

Asks the user to input a value of the adequate type (corresponding to the selected control, and forwards it to the labview control using the **LabviewClient**(p. 9) interface.

This version opens a Dialog Box with an adequate edit widget for inserting a value

Definition at line 1640 of file TacoSilvia.cpp.

References blist, LabviewClient::DevStringToBoolean(), LabviewClient::DevStringToFloat(), LabviewClient::DevStringToLong(), LabviewClient::DevStringToShort(), LabviewClient::DevStringToULong(), LabviewClient::DevStringToUShort(), lv_client, LabviewClient::SetBooleanValue(), LabviewClient::SetDoubleValue(), LabviewClient::SetFloatValue(), LabviewClient::SetLongValue(), LabviewClient::SetShortValue(), LabviewClient::SetStringValue(), LabviewClient::SetULongValue(), LabviewClient::SetUShortValue(), state, STATE_CONTROLSLISTED, STATE_IMPORTED, table, and LabviewClient::typeFromString().

Referenced by initConnections(), and showTableContextMenu().

4.2.3.28 void TacoSilvia::setIndUpdateValue (int *row*, int *col*) [slot]

setIndUpdateValue()(p. 31) is called when the user changes the update flag by clicking on it

To make it clear the text is altered too.

Definition at line 1954 of file TacoSilvia.cpp.

References indtable, state, STATE_CONTROLSLISTED, and STATE_IMPORTED.

Referenced by initConnections().

4.2.3.29 void TacoSilvia::setNotImportedState ()

Grants access to the state machine.

Enables setting the state from outside of the class

Definition at line 2102 of file TacoSilvia.cpp.

References STATE_NOTIMPORTED, and stateMachine().

4.2.3.30 void TacoSilvia::setTextControlValue (int *row*, int *col*) [slot]

setTextControlValue()(p. 31) is called when the user has edited a table item on-the-fly by double-clicking it

This function is called whenever a control value has been edited on-the-fly. So we just figure out the corresponding indices, and push the control value to labview via the **LabviewClient**(p. 9) interface

Definition at line 1833 of file TacoSilvia.cpp.

References LabviewClient::DevStringToBoolean(), LabviewClient::DevStringToDouble(), LabviewClient::DevStringToFloat(), LabviewClient::DevStringToLong(), LabviewClient::DevStringToShort(), LabviewClient::DevStringToULong(), LabviewClient::DevStringToUShort(), lv_client, LabviewClient::SetBooleanValue(), LabviewClient::SetDoubleValue(), LabviewClient::SetFloatValue(), LabviewClient::SetLongValue(), LabviewClient::SetShortValue(), LabviewClient::SetStringValue(), LabviewClient::SetULongValue(), LabviewClient::SetUShortValue(), state, STATE_CONTROLSLISTED, STATE_IMPORTED, and table.

Referenced by initConnections().

4.2.3.31 void TacoSilvia::showIndTableContextMenu (int *row*, int *col*, const QPoint & *pos*) [slot]

showIndTableContextMenu()(p. 32) is used to display a context menu when the user clicks on an indicator in the QTable

Definition at line 2012 of file TacoSilvia.cpp.

References getIndicatorValue(), and palette.

Referenced by initConnections().

4.2.3.32 void TacoSilvia::showTableContextMenu (int *row*, int *col*, const QPoint & *pos*) [slot]

showContextMenu() is used to display a context menu when the user clicks on a control in the QTable.

The User may set or get a control value of the selected control.

Definition at line 1994 of file TacoSilvia.cpp.

References getControlValue(), palette, and setControlValue().

Referenced by initConnections().

4.2.3.33 int TacoSilvia::stateMachine (int *state_flags*) [protected]

This function checks in what kind of state the application finds itself and adapts the widgets correspondigly.

Definition at line 563 of file TacoSilvia.cpp.

References bgetiv, bgetv, bilist, bimport, binfo, blist, bsetv, check_update_box, commandsControlListAction, commandsGetIndValueAction, commandsGetValueAction, commandsIndicatorListAction, commandsSetValueAction, commandsViInfoAction, controls_label, deviceCloseAction, deviceImportAction, helpAboutAction, helpInstructionsAction, ind_check_update_box, ind_interval_spinbox, indicators_label, indtable, interval_spinbox, palette, state, STATE_CONTROLSLISTED, STATE_IMPORTED, STATE_INDICATORSLISTED, STATE_NOTIMPORTED, statusMessage, and table.

Referenced by closeDevice(), getControlList(), getControlValue(), getIndicatorList(), getIndicatorValue(), importDevice(), setNotImportedState(), and TacoSilvia().

4.2.3.34 void TacoSilvia::updateControlsActivated (int *state*) [slot]

Called whenever the check-button changes state.

If activated, the timer for synchronizing the values has to be set

Definition at line 2029 of file TacoSilvia.cpp.

References interval_spinbox, and table.

Referenced by initConnections().

4.2.3.35 void TacoSilvia::updateIndicatorsActivated (int *state*) [slot]

Called whenever the check-button changes state.

If activated, the timer for synchronizing the values has to be set

Definition at line 2042 of file TacoSilvia.cpp.

References ind_interval_spinbox, and indtable.

Referenced by initConnections().

4.2.4 Field Documentation

4.2.4.1 QPixmap TacoSilvia::appicon [protected]

Definition at line 229 of file TacoSilvia.h.

Referenced by about(), and initWindow().

4.2.4.2 QPushButton* TacoSilvia::bgetiv [protected]

Definition at line 213 of file TacoSilvia.h.

Referenced by initConnections(), initWidgets(), setColors(), and stateMachine().

4.2.4.3 QPushButton* TacoSilvia::bgetv [protected]

Definition at line 211 of file TacoSilvia.h.

Referenced by initConnections(), initWidgets(), setColors(), and stateMachine().

4.2.4.4 QPushButton* TacoSilvia::bilist [protected]

Definition at line 210 of file TacoSilvia.h.

Referenced by initConnections(), initWidgets(), setColors(), and stateMachine().

4.2.4.5 QPushButton* TacoSilvia::bimport [protected]

Definition at line 207 of file TacoSilvia.h.

Referenced by deviceImportClose(), initConnections(), initWidgets(), setColors(), and stateMachine().

4.2.4.6 QPushButton∗ TacoSilvia::binfo [protected]

Definition at line 208 of file TacoSilvia.h.

Referenced by initConnections(), initWidgets(), setColors(), and stateMachine().

4.2.4.7 QPushButton∗ TacoSilvia::blist [protected]

Definition at line 209 of file TacoSilvia.h.

Referenced by initConnections(), initWidgets(), setColors(), setControlValue(), and stateMachine().

4.2.4.8 QPushButton∗ TacoSilvia::bsetv [protected]

Definition at line 212 of file TacoSilvia.h.

Referenced by initConnections(), initWidgets(), setColors(), and stateMachine().

4.2.4.9 QCheckBox∗ TacoSilvia::check_update_box [protected]

Definition at line 203 of file TacoSilvia.h.

Referenced by initConnections(), initWidgets(), intervalControlsChanged(), setColors(), and stateMachine().

4.2.4.10 QAction∗ TacoSilvia::commandsControlListAction

Definition at line 142 of file TacoSilvia.h.

Referenced by initActions(), initConnections(), initMenus(), languageChange(), and stateMachine().

4.2.4.11 QAction∗ TacoSilvia::commandsGetIndValueAction

Definition at line 146 of file TacoSilvia.h.

Referenced by initActions(), initConnections(), initMenus(), languageChange(), and stateMachine().

4.2.4.12 QAction∗ TacoSilvia::commandsGetValueAction

Definition at line 144 of file TacoSilvia.h.

Referenced by initActions(), initConnections(), initMenus(), languageChange(), and stateMachine().

4.2.4.13 QAction∗ TacoSilvia::commandsIndicatorListAction

Definition at line 143 of file TacoSilvia.h.

Referenced by initActions(), initConnections(), initMenus(), languageChange(), and stateMachine().

4.2.4.14 QPopupMenu∗ TacoSilvia::commandsmenu [protected]

Definition at line 186 of file TacoSilvia.h.

Referenced by initMenus(), and setColors().

4.2.4.15 QAction∗ TacoSilvia::commandsSetValueAction

Definition at line 145 of file TacoSilvia.h.

Referenced by initActions(), initConnections(), initMenus(), languageChange(), and stateMachine().

4.2.4.16 QAction∗ TacoSilvia::commandsViInfoAction

Definition at line 141 of file TacoSilvia.h.

Referenced by initActions(), initConnections(), initMenus(), languageChange(), and stateMachine().

4.2.4.17 QLabel∗ TacoSilvia::controls_label [protected]

Definition at line 195 of file TacoSilvia.h.

Referenced by addWidgets(), initWidgets(), and stateMachine().

4.2.4.18 QAction∗ TacoSilvia::deviceCloseAction

Definition at line 138 of file TacoSilvia.h.

Referenced by initActions(), initConnections(), initMenus(), languageChange(), and stateMachine().

4.2.4.19 QAction∗ TacoSilvia::deviceExitAction

Definition at line 139 of file TacoSilvia.h.

Referenced by initActions(), initConnections(), initMenus(), and languageChange().

4.2.4.20 QAction∗ TacoSilvia::deviceImportAction

Definition at line 137 of file TacoSilvia.h.

Referenced by initActions(), initConnections(), initMenus(), languageChange(), and stateMachine().

4.2.4.21 QPopupMenu∗ TacoSilvia::devicemenu [protected]

Definition at line 185 of file TacoSilvia.h.

Referenced by initMenus(), and setColors().

4.2.4.22 QErrorMessage∗ TacoSilvia::errmsg [protected]

Definition at line 214 of file TacoSilvia.h.

Referenced by initWindow().

4.2.4.23 QFrame∗ TacoSilvia::frame [protected]

Definition at line 181 of file TacoSilvia.h.

Referenced by initTables(), initWidgets(), and initWindow().

4.2.4.24 QGridLayout∗ TacoSilvia::g [protected]

Definition at line 182 of file TacoSilvia.h.

Referenced by addWidgets(), and initWindow().

4.2.4.25 QHBoxLayout∗ TacoSilvia::hbl0 [protected]

Definition at line 221 of file TacoSilvia.h.

Referenced by addWidgets(), and initWidgets().

4.2.4.26 QHBoxLayout∗ TacoSilvia::hbl1 [protected]

Definition at line 222 of file TacoSilvia.h.

Referenced by initWidgets().

4.2.4.27 QHBoxLayout∗ TacoSilvia::hbl2 [protected]

Definition at line 223 of file TacoSilvia.h.

Referenced by addWidgets(), and initWidgets().

4.2.4.28 QHBoxLayout∗ TacoSilvia::hbl3 [protected]

Definition at line 224 of file TacoSilvia.h.

Referenced by addWidgets(), and initWidgets().

4.2.4.29 QHBoxLayout∗ TacoSilvia::hbl4 [protected]

Definition at line 225 of file TacoSilvia.h.

Referenced by addWidgets(), and initWidgets().

4.2.4.30 QHBoxLayout∗ TacoSilvia::hbl5 [protected]

Definition at line 226 of file TacoSilvia.h.

Referenced by addWidgets(), and initWidgets().

4.2.4.31 QAction* TacoSilvia::helpAboutAction

Definition at line 149 of file TacoSilvia.h.

Referenced by initActions(), initConnections(), initMenus(), languageChange(), and stateMachine().

4.2.4.32 QAction* TacoSilvia::helpInstructionsAction

Definition at line 148 of file TacoSilvia.h.

Referenced by initActions(), initConnections(), initMenus(), languageChange(), and stateMachine().

4.2.4.33 QPopupMenu* TacoSilvia::helpmenu [protected]

Definition at line 187 of file TacoSilvia.h.

Referenced by initMenus(), and setColors().

4.2.4.34 QCheckBox* TacoSilvia::ind_check_update_box [protected]

Definition at line 205 of file TacoSilvia.h.

Referenced by initConnections(), initWidgets(), intervalIndicatorsChanged(), setColors(), and stateMachine().

4.2.4.35 QSpinBox* TacoSilvia::ind_interval_spinbox [protected]

Definition at line 204 of file TacoSilvia.h.

Referenced by initConnections(), initWidgets(), setColors(), stateMachine(), and updateIndicatorsActivated().

4.2.4.36 QLabel* TacoSilvia::indicators_label [protected]

Definition at line 196 of file TacoSilvia.h.

Referenced by addWidgets(), initWidgets(), and stateMachine().

4.2.4.37 QTable* TacoSilvia::indtable [protected]

Definition at line 191 of file TacoSilvia.h.

Referenced by addWidgets(), closeDevice(), getIndicatorList(), getIndicatorValue(), initConnections(), initTables(), intervalIndicatorsChanged(), setIndUpdateValue(), stateMachine(), and updateIndicatorsActivated().

4.2.4.38 QSpinBox* TacoSilvia::interval_spinbox [protected]

Definition at line 202 of file TacoSilvia.h.

Referenced by initConnections(), initWidgets(), setColors(), stateMachine(), and updateControlsActivated().

4.2.4.39 QLabel* TacoSilvia::label [protected]

Definition at line 193 of file TacoSilvia.h.

4.2.4.40 QListBox* TacoSilvia::lbox [protected]

Definition at line 188 of file TacoSilvia.h.

4.2.4.41 LabviewClient* TacoSilvia::lv_client [protected]

Definition at line 219 of file TacoSilvia.h.

Referenced by closeDevice(), getControlList(), getControlValue(), getIndicatorList(), getIndicatorValue(), getLabviewClient(), getViInfo(), importDevice(), initWindow(), setControlValue(), and setTextControlValue().

4.2.4.42 QListView* TacoSilvia::lview [protected]

Definition at line 189 of file TacoSilvia.h.

4.2.4.43 QMenuBar* TacoSilvia::menubar [protected]

Definition at line 183 of file TacoSilvia.h.

Referenced by initMenus(), and setColors().

4.2.4.44 QPixmap TacoSilvia::p_about [protected]

Definition at line 241 of file TacoSilvia.h.

Referenced by initActions(), and initIcons().

4.2.4.45 QPixmap TacoSilvia::p_device_close [protected]

Definition at line 232 of file TacoSilvia.h.

Referenced by initActions(), and initIcons().

4.2.4.46 QPixmap TacoSilvia::p_device_import [protected]

Definition at line 231 of file TacoSilvia.h.

Referenced by initActions(), and initIcons().

4.2.4.47 QPixmap TacoSilvia::p_get_ind_value [protected]

Definition at line 239 of file TacoSilvia.h.

Referenced by initActions(), and initIcons().

4.2.4.48 QPixmap TacoSilvia::p_get_value [protected]

Definition at line 237 of file TacoSilvia.h.

Referenced by initActions(), and initIcons().

4.2.4.49 QPixmap TacoSilvia::p_instructions [protected]

Definition at line 242 of file TacoSilvia.h.

Referenced by initActions(), and initIcons().

4.2.4.50 QPixmap TacoSilvia::p_list_controls [protected]

Definition at line 235 of file TacoSilvia.h.

Referenced by initActions(), and initIcons().

4.2.4.51 QPixmap TacoSilvia::p_list_indicators [protected]

Definition at line 236 of file TacoSilvia.h.

Referenced by initActions(), and initIcons().

4.2.4.52 QPixmap TacoSilvia::p_set_value [protected]

Definition at line 238 of file TacoSilvia.h.

Referenced by initActions(), and initIcons().

4.2.4.53 QPixmap TacoSilvia::p_vi_info [protected]

Definition at line 234 of file TacoSilvia.h.

Referenced by initActions(), and initIcons().

4.2.4.54 QPalette TacoSilvia::palette [protected]

Definition at line 228 of file TacoSilvia.h.

Referenced by initPalettes(), initWindow(), setColors(), showIndTableContextMenu(), showTableContextMenu(), and stateMachine().

4.2.4.55 int TacoSilvia::state [protected]

Definition at line 249 of file TacoSilvia.h.

Referenced by deviceImportClose(), getControlList(), getControlValue(), getIndicatorValue(), getViInfo(), setControlValue(), setIndUpdateValue(), setTextControlValue(), and stateMachine().

4.2.4.56 int TacoSilvia::STATE_CONTROLSLISTED = 2 [static, protected]

Definition at line 58 of file TacoSilvia.cpp.

Referenced by getControlList(), getControlValue(), setControlValue(), setIndUpdateValue(), setTextControlValue(), and stateMachine().

4.2.4.57 int TacoSilvia::STATE_IMPORTED = 1 [static, protected]

Definition at line 57 of file TacoSilvia.cpp.

Referenced by deviceImportClose(), getControlList(), getControlValue(), getIndicatorValue(), getViInfo(), importDevice(), setControlValue(), setIndUpdateValue(), setTextControlValue(), and stateMachine().

4.2.4.58 int TacoSilvia::STATE_INDICATORSLISTED = 3 [static, protected]

Definition at line 59 of file TacoSilvia.cpp.

Referenced by getIndicatorList(), getIndicatorValue(), and stateMachine().

4.2.4.59 int TacoSilvia::STATE_NOTIMPORTED = 0 [static, protected]

Definition at line 56 of file TacoSilvia.cpp.

Referenced by closeDevice(), getControlList(), getControlValue(), getIndicatorList(), getIndicatorValue(), setNotImportedState(), stateMachine(), and TacoSilvia().

4.2.4.60 QLabel* TacoSilvia::statusLabel [protected]

Definition at line 198 of file TacoSilvia.h.

Referenced by initWidgets().

4.2.4.61 QLabel* TacoSilvia::statusMessage [protected]

Definition at line 199 of file TacoSilvia.h.

Referenced by initWidgets(), setColors(), and stateMachine().

4.2.4.62 QControlTip* TacoSilvia::t [protected]

Definition at line 216 of file TacoSilvia.h.

Referenced by initTables().

4.2.4.63 QControlsTable* TacoSilvia::table [protected]

Definition at line 190 of file TacoSilvia.h.

Referenced by addWidgets(), clickTableItem(), closeDevice(), getControlList(), getControlValue(), getIndicatorList(), initConnections(), initTables(), intervalControlsChanged(), setControlValue(), setTextControlValue(), stateMachine(), and updateControlsActivated().

4.2.4.64 QToolBar∗ TacoSilvia::toolbar `[protected]`

Definition at line 184 of file TacoSilvia.h.

Referenced by initMenus(), languageChange(), and setColors().

4.2.4.65 QTextEdit∗ TacoSilvia::vinfo `[protected]`

Definition at line 218 of file TacoSilvia.h.

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

- **TacoSilvia.h**
- **TacoSilvia.cpp**

Chapter 5

TacoSilvia File Documentation

5.1 LabviewClient.cpp File Reference

```
#include <macros.h>
#include <API.h>
#include <ApiP.h>
#include <Admin.h>
#include <BlcDsNumbers.h>
#include <DevServer.h>
#include <DevServerP.h>
#include <DevSignal.h>
#include <DevErrors.h>
#include <maxe_xdr.h>
#include <db_setup.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <errno.h>
#include <ctype.h>
#include "LabviewClient.h"
#include "LabViewGeneric.h"
```

5.2 LabviewClient.h File Reference

```
#include <DevServer.h>
```

Data Structures

- class **LabviewClient**

  The **LabviewClient** *represents a TACO client and acts as a type-wrapper for the Labview-Taco-Interface.*

Defines

- #define **WRONG_CONTROL_ERROR** -1
- #define **WRONG_TYPE_ERROR** -2
- #define **WRONG_ARGSNUM_ERROR** -3
- #define **CONVERSION_FAILED_ERROR** -4

5.2.1 Define Documentation

5.2.1.1 #define CONVERSION_FAILED_ERROR -4

Definition at line 34 of file LabviewClient.h.

5.2.1.2 #define WRONG_ARGSNUM_ERROR -3

Definition at line 33 of file LabviewClient.h.

5.2.1.3 #define WRONG_CONTROL_ERROR -1

Definition at line 31 of file LabviewClient.h.

5.2.1.4 #define WRONG_TYPE_ERROR -2

Definition at line 32 of file LabviewClient.h.

5.3 lv_menu.cpp File Reference

```
#include <Admin.h>
#include <API.h>
#include <DevServer.h>
#include <stdio.h>
#include "LabviewClient.h"
```

Functions

- void **getUserInput** (char *text, char **val_string)
- int **main** (int argc, char **argv)

5.3.1 Function Documentation

5.3.1.1 void getUserInput (char * *text*, char ** *val_string*)

Definition at line 8 of file lv_menu.cpp.

Referenced by main().

5.3.1.2 int main (int *argc*, char ** *argv*)

Definition at line 22 of file lv_menu.cpp.

References LabviewClient::DevStringToBoolean(), LabviewClient::DevStringToDouble(), LabviewClient::DevStringToFloat(), LabviewClient::DevStringToLong(), LabviewClient::DevStringToShort(), LabviewClient::DevStringToULong(), LabviewClient::DevStringToUShort(), LabviewClient::GetBooleanValue(), LabviewClient::GetControlInfo(), LabviewClient::GetControlList(), LabviewClient::GetControlType(), LabviewClient::GetDoubleValue(), LabviewClient::GetFloatValue(), LabviewClient::GetLongValue(), LabviewClient::GetShortValue(), LabviewClient::GetStringValue(), LabviewClient::GetULongValue(), getUserInput(), LabviewClient::GetUShortValue(), LabviewClient::GetViInfo(), LabviewClient::InitDevice(), LabviewClient::SetBooleanValue(), LabviewClient::SetDoubleValue(), LabviewClient::SetFloatValue(), LabviewClient::SetLongValue(), LabviewClient::SetShortValue(), LabviewClient::SetStringValue(), LabviewClient::SetULongValue(), and LabviewClient::SetUShortValue().

5.4 TacoSilvia.cpp File Reference

```
#include "TacoSilvia.h"
#include <qcursor.h>
#include <qpopupmenu.h>
#include <qmessagebox.h>
#include <qinputdialog.h>
#include <qerrormessage.h>
#include <qpixmap.h>
#include <qbitmap.h>
#include <qpainter.h>
#include <qtextedit.h>
#include <qlistbox.h>
#include <qlistview.h>
#include <qlayout.h>
#include <qpushbutton.h>
#include <qtable.h>
#include <qstringlist.h>
#include <qvariant.h>
#include <qtooltip.h>
#include <qwhatsthis.h>
#include <qaction.h>
#include <qmenubar.h>
#include <qtoolbar.h>
#include <qimage.h>
#include <qlabel.h>
#include <qvalidator.h>
#include <qstylefactory.h>
#include <qpalette.h>
#include <qevent.h>
#include <qheader.h>
#include <qcheckbox.h>
#include <qspinbox.h>
#include <qlineedit.h>
#include <qapplication.h>
#include <qdialog.h>
#include <Admin.h>
```

```
#include <API.h>
#include <DevServer.h>
#include "LabviewClient.h"
#include "Pixmaps.h"
```

5.5 TacoSilvia.h File Reference

```
#include <qvariant.h>
#include <qpixmap.h>
#include <qmainwindow.h>
#include <qtable.h>
#include <qtooltip.h>
```

Data Structures

- class **TacoSilvia**

  TacoSilvia *means* **TACO S***erver* **I***nterfacing* **L***abview* **V***irtual* **I***nstrument* **A***pplications. It represents a GUI application built with Qt and allows communication with arbitrary Labview Virtual Instruments(VIs) by means of getting and setting the control and indicator values of selected VIs. It uses TACO and the Labview-Taco-Lib to offer an interface to Labview applications.*

Index