

XML Viewer Reference Manual

Generated by Doxygen 1.2.5

Mon Mar 26 23:54:25 2001

Contents

1	XML Viewer Hierarchical Index	1
1.1	XML Viewer Class Hierarchy	1
2	XML Viewer Compound Index	3
2.1	XML Viewer Compound List	3
3	XML Viewer File Index	5
3.1	XML Viewer File List	5
4	XML Viewer Class Documentation	7
4.1	DialogThread Class Reference	7
4.2	QThread Class Reference	9
4.3	SelectionGUI Class Reference	10
4.4	VCell Class Reference	11
4.5	vect_code_names Struct Reference	13
4.6	VObject Class Reference	14
4.7	VRow Class Reference	17
4.8	VSelectionGUI Class Reference	21
5	XML Viewer File Documentation	25
5.1	DialogThread.cpp File Reference	25
5.2	DialogThread.h File Reference	26
5.3	Interface.cpp File Reference	27
5.4	Interface.h File Reference	28
5.5	main.cpp File Reference	30
5.6	VCell.cpp File Reference	31
5.7	VCell.h File Reference	32
5.8	VObject.cpp File Reference	33
5.9	VObject.h File Reference	34

5.10 VRow.cpp File Reference	35
5.11 VRow.h File Reference	36
5.12 VSelectionGUI.cpp File Reference	37
5.13 VSelectionGUI.h File Reference	38

Chapter 1

XML Viewer Hierarchical Index

1.1 XML Viewer Class Hierarchy

This inheritance list is sorted roughly, but not completely, alphabetically:

QThread	9
DialogThread	7
SelectionGUI	10
VSelectionGUI	21
VCell	11
vect_code_names	13
VObject	14
VRow	17

Chapter 2

XML Viewer Compound Index

2.1 XML Viewer Compound List

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

DialogThread (This class is the execution Thread of the GUI)	7
QThread	9
SelectionGUI	10
VCell (This class is the abstract representation of the <cell> tag and all the tags inside of it)	11
vect_code_names	13
VObject (This class is the abstract representation of the <module>, <process> and the <category> tags and together the data inside them)	14
VRow (This class is the abstract representation of the <row> tag, with all the data inside it. (Same as VObject (p.14), but specialized for the <row> tags)) . . .	17
VSelectionGUI (This class is the main widget)	21

Chapter 3

XML Viewer File Index

3.1 XML Viewer File List

Here is a list of all files with brief descriptions:

DialogThread.cpp	25
DialogThread.h	26
Interface.cpp	27
Interface.h	28
main.cpp	30
VCell.cpp	31
VCell.h	32
VObject.cpp	33
VObject.h	34
VRow.cpp	35
VRow.h	36
VSelectionGUI.cpp	37
VSelectionGUI.h	38

Chapter 4

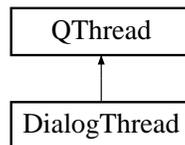
XML Viewer Class Documentation

4.1 DialogThread Class Reference

This class is the execution Thread of the GUI.

```
#include <DialogThread.h>
```

Inheritance diagram for DialogThread:



Public Methods

- **DialogThread** (char *xmlText)
- virtual void **run** ()
- char* **getName** (char *code)
- vect_code_names* **getChoice** ()

Private Attributes

- VSelectionGUI* **selectionGUI**
- bool **accessOk**
- char* **aux**
- char* **xmlStream**

4.1.1 Detailed Description

This class is the execution Thread of the GUI.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 DialogThread::DialogThread (char * *xmlText*)

Constructs the thread for the GUI. *xmlText* is a char buffer which contains the XML input file for the **VSelectionGUI** (p. 21).

4.1.3 Member Function Documentation

4.1.3.1 vect_code_names * DialogThread::getChoice ()

Return the current selection of the GUI. (only the codes from the variables

- the column which contains the codes is defined by "codecolumn" attribute in the XML file.

4.1.3.2 char * DialogThread::getName (char * *code*)

Get the description of the code (returns the entire line from the QListView).

4.1.3.3 void DialogThread::run () [virtual]

The run method - this method is invocated when start is called (starts the thread).

4.1.4 Member Data Documentation

4.1.4.1 bool DialogThread::accessOk [private]

4.1.4.2 char * DialogThread::aux [private]

4.1.4.3 VSelectionGUI * DialogThread::selectionGUI [private]

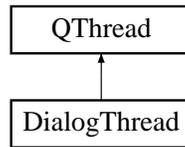
4.1.4.4 char * DialogThread::xmlStream [private]

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

- **DialogThread.h**
- **DialogThread.cpp**

4.2 QThread Class Reference

Inheritance diagram for QThread:

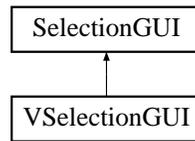


The documentation for this class was generated from the following file:

- **DialogThread.h**

4.3 SelectionGUI Class Reference

Inheritance diagram for SelectionGUI:



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

- **VSelectionGUI.h**

4.4 VCell Class Reference

This class is the abstract representation of the <cell> tag and all the tags inside of it.

```
#include <VCell.h>
```

Public Methods

- **VCell** ()
- **VCell** (QDomElement elem)
- **~VCell** ()
- **QString wordAt** (uint index)
- **uint countWords** ()
- **QString getString** ()

Protected Attributes

- **QList<QString> words**

4.4.1 Detailed Description

This class is the abstract representation of the <cell> tag and all the tags inside of it.

4.4.2 Constructor & Destructor Documentation

4.4.2.1 VCell::VCell ()

Default constructor

4.4.2.2 VCell::VCell (QDomElement *elem*)

Dom Element constructor

4.4.2.3 VCell::~~VCell ()

Destructor

4.4.3 Member Function Documentation

4.4.3.1 uint VCell::countWords ()

Returns the number of words inside the current cell.

4.4.3.2 QString VCell::getString ()

4.4.3.3 QString VCell::wordAt (uint *index*)

Returns the word at position index

4.4.4 Member Data Documentation

4.4.4.1 `QList< QString > VCell::words` [protected]

Internal representation of the word's list.

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

- `VCell.h`
- `VCell.cpp`

4.5 vect_code_names Struct Reference

```
#include <Interface.h>
```

Public Attributes

- char** **single_code_name**
- unsigned int **num_code_names**

4.5.1 Detailed Description

Structure used as return type for gui_choice.

4.5.2 Member Data Documentation

4.5.2.1 unsigned int vect_code_names::num_code_names

4.5.2.2 char ** vect_code_names::single_code_name

The documentation for this struct was generated from the following file:

- **Interface.h**

4.6 VObject Class Reference

This class is the abstract representation of the <module>, <process> and the <category> tags and together the data inside them.

```
#include <VObject.h>
```

Public Methods

- **VObject** ()
- **VObject** (QDomElement element)
- virtual **~VObject** ()
- void **setType** (const QString type)
- const QString **getType** ()
- void **setText** (const QString text)
- const QString **getText** ()
- void **addChild** (VObject *child)
- VObject* **childAt** (int index)
- VRow* **rowAt** (int index)
- void **clearChildren** ()
- uint **countChildren** ()
- uint **countRows** ()
- bool **hasChildren** ()
- void **resetSelection** ()
- void **getSelection** (VSelectionGUI *selGUI)

Private Attributes

- QString **objectType**
- QString **objectText**
- QList<VObject> **childrensList**
- VRow* **rowList**

4.6.1 Detailed Description

This class is the abstract representation of the <module>, <process> and the <category> tags and together the data inside them.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 VObject::VObject ()

Default constructor - constructs an empty object of the type "VObject" (undefined type).

4.6.2.2 VObject::VObject (QDomElement *element*)

Dom Element constructor (element is read recursively to create this object and it's children).

4.6.2.3 VObject::~~VObject () [virtual]

Destructor

4.6.3 Member Function Documentation**4.6.3.1 void VObject::addChild (VObject * *child*)**

Add a child to the VObject.

4.6.3.2 VObject * VObject::childAt (int *index*)

Returns the child number index.

4.6.3.3 void VObject::clearChildren ()

Delete all children.

4.6.3.4 uint VObject::countChildren ()

Returns the number of children.

4.6.3.5 uint VObject::countRows ()

Returns the number of children of the type row.

4.6.3.6 void VObject::getSelection (VSelectionGUI * *selGUI*)

Return the current selection to the GUI - Apply pressed.

4.6.3.7 const QString VObject::getText ()

Return the current text.

4.6.3.8 const QString VObject::getType ()

Returns the VObject's type (module, process or category).

4.6.3.9 bool VObject::hasChildren ()

Returns true if VObject has children.

4.6.3.10 void VObject::resetSelection ()

Reset the selection to default - Reset pressed.

4.6.3.11 `VRow * VObject::rowAt (int index)`

Returns the child of type `VRow` (p. 17) number `index`.

4.6.3.12 `void VObject::setText (const QString text)`

Set the text from the label tag.

4.6.3.13 `void VObject::setType (const QString type)`

The type is the tag's name.

4.6.4 Member Data Documentation**4.6.4.1** `QList< VObject > VObject::childrenList [private]`**4.6.4.2** `QString VObject::objectText [private]`**4.6.4.3** `QString VObject::objectType [private]`**4.6.4.4** `VRow * VObject::rowList [private]`

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

- `VObject.h`
- `VObject.cpp`

4.7 VRow Class Reference

This class is the abstract representation of the <row> tag, with all the data inside it. (Same as **VObject** (p. 14), but specialized for the <row> tags).

```
#include <VRow.h>
```

Public Methods

- **VRow** ()
- **VRow** (QDomElement elem, bool diveInto=TRUE)
- **~VRow** ()
- **VCell*** **cellAt** (uint index)
- uint **countCells** ()
- void **addChild** (VCell *child)
- void **addChild** (VRow *child)
- VRow* **childAt** (int index)
- void **clearChildren** ()
- uint **countChildren** ()
- bool **hasChildren** ()
- void **setSelected** (bool sel)
- bool **isSelected** ()
- void **resetSelection** ()
- void **getSelection** (VSelectionGUI *selGUI)
- void **setDefault** (bool sel)
- bool **isDefault** ()
- void **setType** (const QString type)
- const QString **getType** ()
- QListViewItem* **getQListViewItem** (QListView *parent, VSelectionGUI *selectionGUI)
- QListViewItem* **getQListViewItem** (QListViewItem *parent, VSelectionGUI *selectionGUI)

Private Attributes

- QList<VRow> **children**
- QList<VCell> **cells**
- bool **selected**
- bool **defaultSelected**
- QString **objectType**

4.7.1 Detailed Description

This class is the abstract representation of the <row> tag, with all the data inside it. (Same as **VObject** (p. 14), but specialized for the <row> tags).

4.7.2 Constructor & Destructor Documentation

4.7.2.1 VRow::VRow ()

Default constructor - constructs an empty object.

4.7.2.2 VRow::VRow (QDomElement *elem*, bool *diveInto* = TRUE)

Dom Element constructor (element is red recursively to create this object and it's children).

4.7.2.3 VRow::~~VRow ()

Destructor

4.7.3 Member Function Documentation**4.7.3.1 void VRow::addChild (VRow * *child*)**

Add a child of type VRow to the VRow.

4.7.3.2 void VRow::addChild (VCell * *child*)

Add a child of type VCell (p. 11) to the VRow.

4.7.3.3 VCell * VRow::cellAt (uint *index*)

Returns the cell number index.

4.7.3.4 VRow * VRow::childAt (int *index*)

Returns the row number index.

4.7.3.5 void VRow::clearChildren ()

Delete all row children.

4.7.3.6 uint VRow::countCells ()

Returns the cell count.

4.7.3.7 uint VRow::countChildren ()

Returns the row count.

4.7.3.8 QListViewItem * VRow::getQListViewItem (QListViewItem * *parent*, VSelectionGUI * *selectionGUI* = 0)

????.

4.7.3.9 QListViewItem * VRow::getQListViewItem (QListView * *parent*, VSelectionGUI * *selectionGUI* = 0)

????.

4.7.3.10 void VRow::getSelection (VSelectionGUI * *selGUI*)

Return the current selection to the GUI - Apply pressed.

4.7.3.11 const QString VRow::getType ()

Must be removed.

4.7.3.12 bool VRow::hasChildren ()

Returns true if VRow has children.

4.7.3.13 bool VRow::isDefault ()

Returns if the object is selected by default.

4.7.3.14 bool VRow::isSelected ()

Returns if item is selected.

4.7.3.15 void VRow::resetSelection ()

Reset the current selection to default.

4.7.3.16 void VRow::setDefault (bool *sel*)

Set wheder the object is selected by default or not.

4.7.3.17 void VRow::setSelected (bool *sel*)

Set wheder item is selected or not.

4.7.3.18 void VRow::setType (const QString *type*)

Must be removed.

4.7.4 Member Data Documentation

4.7.4.1 `QList< VCell > VRow::cells` [private]

4.7.4.2 `QList< VRow > VRow::children` [private]

4.7.4.3 `bool VRow::defaultSelected` [private]

4.7.4.4 `QString VRow::objectType` [private]

4.7.4.5 `bool VRow::selected` [private]

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

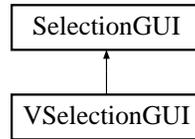
- `VRow.h`
- `VRow.cpp`

4.8 VSelectionGUI Class Reference

This class is the main widget.

```
#include <VSelectionGUI.h>
```

Inheritance diagram for VSelectionGUI:



Public Methods

- **VSelectionGUI** (const char *xmlText, QWidget *parent=0, const char *name=0)
- **~VSelectionGUI** ()
- void **addListViewItem** (QListViewItem *item, **VRow** *row)
- char* **getName** (char *code)
- **vect_code_names* getChoice** ()
- void **addListSelected** (**VRow** *row)

Protected Methods

- void **setXML** (const char *xmlText)

Private Attributes

- QMutex **myMutex**
- QString* **helpText**
- HelpGUI* **myHelp**
- char** **currentChoice**
- QList<QString> **currentChoiceList**
- QList<QString> **descriptionList**
- QList<QString> **codeList**
- QList<int> **codeIndexList**
- **vect_code_names** **newChoice**
- unsigned int **codeColumn**
- unsigned int **unitsColumn**
- unsigned int **descriptionColumn**
- unsigned int **moduleName**
- unsigned int **processNumber**
- unsigned int **categoryNumber**
- **VObject*** **rootObject**
- QList<QListViewItem> **listOfListViewRows**
- QList<**VRow**> **listOfRows**

Private Slots

- void **helpAction** ()
- void **resetAction** ()
- void **applyAction** ()
- void **moduleChanged** (int *index*)
- void **processChanged** (int *index*)
- void **categoryChanged** (int *index*)

4.8.1 Detailed Description

This class is the main widget.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 VSelectionGUI::VSelectionGUI (const char * *xmlText*, QWidget * *parent* = 0, const char * *name* = 0)

Constructs the GUI and initializes it with the buffer readed from the XML file.

4.8.2.2 VSelectionGUI::~~VSelectionGUI ()

Destructor

4.8.3 Member Function Documentation

4.8.3.1 void VSelectionGUI::addListSelected (VRow * *row*)

Collects the current selected variable codes - invoked from each **VRow** (p.17) object. Only for internal use regarding the selection mechanism.

4.8.3.2 void VSelectionGUI::addListViewItem (QListViewItem * *item*, VRow * *row*)

Adds a QListViewItem to an internal list. Only for internal use regarding the selection mechanism.

4.8.3.3 void VSelectionGUI::applyAction () [private, slot]

The Apply button is connected to this function.

4.8.3.4 void VSelectionGUI::categoryChanged (int *index*) [private, slot]

The Category Combobox sends a signal to this function.

4.8.3.5 vect_code_names * VSelectionGUI::getChoice ()

Return the current selection.

4.8.3.6 `char * VSelectionGUI::getName (char * code)`

Get the description of a code of a variable (the entire row in the QListView).

4.8.3.7 `void VSelectionGUI::helpAction () [private, slot]`

The Help button is connected to this function.

4.8.3.8 `void VSelectionGUI::moduleChanged (int index) [private, slot]`

The Module Combobox sends a signal to this function.

4.8.3.9 `void VSelectionGUI::processChanged (int index) [private, slot]`

The Process Combobox sends a signal to this function.

4.8.3.10 `void VSelectionGUI::resetAction () [private, slot]`

The Reset button is connected to this function.

4.8.3.11 `void VSelectionGUI::setXML (const char * xmlText) [protected]`

Set the XML Document dor initializing the GUI.

4.8.4 Member Data Documentation

- 4.8.4.1 unsigned int VSelectionGUI::categoryNumber [private]
- 4.8.4.2 unsigned int VSelectionGUI::codeColumn [private]
- 4.8.4.3 QList< int > VSelectionGUI::codeIndexList [private]
- 4.8.4.4 QList< QString > VSelectionGUI::codeList [private]
- 4.8.4.5 char ** VSelectionGUI::currentChoice [private]
- 4.8.4.6 QList< QString > VSelectionGUI::currentChoiceList [private]
- 4.8.4.7 unsigned int VSelectionGUI::descriptionColumn [private]
- 4.8.4.8 QList< QString > VSelectionGUI::descriptionList [private]
- 4.8.4.9 QString * VSelectionGUI::helpText [private]
- 4.8.4.10 QList< QListViewItem > VSelectionGUI::listOfListViewRows [private]
- 4.8.4.11 QList< VRow > VSelectionGUI::listOfRows [private]
- 4.8.4.12 unsigned int VSelectionGUI::moduleNumber [private]
- 4.8.4.13 HelpGUI * VSelectionGUI::myHelp [private]
- 4.8.4.14 QMutex VSelectionGUI::myMutex [private]
- 4.8.4.15 vect_code_names VSelectionGUI::newChoice [private]
- 4.8.4.16 unsigned int VSelectionGUI::processNumber [private]
- 4.8.4.17 VObject * VSelectionGUI::rootObject [private]
- 4.8.4.18 unsigned int VSelectionGUI::unitsColumn [private]

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

- VSelectionGUI.h
- VSelectionGUI.cpp

Chapter 5

XML Viewer File Documentation

5.1 DialogThread.cpp File Reference

```
#include <iostream.h>
#include <qapplication.h>
#include <qthread.h>
#include <qplatinumstyle.h>
#include "VSelectionGUI.h"
#include "DialogThread.h"
```

5.2 DialogThread.h File Reference

```
#include <qthread.h>  
#include "VSelectionGUI.h"
```

Compounds

- class **DialogThread**

This class is the execution Thread of the GUI.

5.3 Interface.cpp File Reference

```
#include "Interface.h"
#include "VSelectionGUI.h"
#include "DialogThread.h"
#include <iostream.h>
#include <qapplication.h>
```

Functions

- `boolean gui_init (char *xml_stream)`
- `vect_code_names* gui_choice (void)`
- `char* gui_name (char *single_code_name)`
- `boolean gui_clean (void)`

Variables

- `DialogThread* dialogThread`

5.3.1 Function Documentation

5.3.1.1 `vect_code_names * gui_choice (void)`

Returns the string list of variable codes which has to be computed.

5.3.1.2 `boolean gui_clean (void)`

Garbage collector of the GUI.

5.3.1.3 `boolean gui_init (char * xml_stream)`

Start up the GUI.

5.3.1.4 `char * gui_name (char * single_code_name)`

Returns the description of a variable code.

5.3.2 Variable Documentation

5.3.2.1 `DialogThread * dialogThread`

5.4 Interface.h File Reference

Compounds

- struct `vect_code_names`

Typedefs

- typedef struct `vect_code_names` `vect_code_names`

Enumerations

- enum `boolean` { `NO` = 0, `YES` }

Functions

- `boolean` `gui_init` (`char *xml_stream`)
- `vect_code_names*` `gui_choice` (`void`)
- `char*` `gui_name` (`char *single_code_name`)
- `boolean` `gui_clean` (`void`)

5.4.1 Typedef Documentation

5.4.1.1 typedef struct `vect_code_names` `vect_code_names`

5.4.2 Enumeration Type Documentation

5.4.2.1 enum `boolean`

Result type for `gui_init` and `gui_clean` (`NO` = false, `YES` = true).

Enumeration values:

NO

YES

5.4.3 Function Documentation

5.4.3.1 `vect_code_names*` `gui_choice` (`void`)

Returns the string list of variable codes which has to be computed.

5.4.3.2 `boolean` `gui_clean` (`void`)

Garbage collector of the GUI.

5.4.3.3 `boolean` `gui_init` (`char * xml_stream`)

Start up the GUI.

5.4.3.4 char* gui_name (char * *single_code_name*)

Returns the description of a variable code.

5.5 main.cpp File Reference

```
#include "Interface.h"  
#include <stdio.h>
```

Functions

- `int main (void)`

5.5.1 Function Documentation

5.5.1.1 `int main (void)`

The main program is here.

5.6 VCell.cpp File Reference

```
#include <iostream.h>
#include "VCell.h"
#include <qdom.h>
```

5.7 VCell.h File Reference

```
#include <qlist.h>  
#include <qstring.h>  
#include <qdom.h>
```

Compounds

- class **VCell**

This class is the abstract representation of the <cell> tag and all the tags inside of it.

5.8 VObject.cpp File Reference

```
#include <qstring.h>
#include <qlist.h>
#include <qxml.h>
#include <qdom.h>
#include "VRow.h"
#include "VObject.h"
#include "VSelectionGUI.h"
```

5.9 VObject.h File Reference

```
#include <qlist.h>
#include <qstring.h>
#include <qxml.h>
#include <qdom.h>
#include "VRow.h"
#include "VSelectionGUI.h"
```

Compounds

- class **VObject**

This class is the abstract representation of the <module>, <process> and the <category> tags and together the data inside them.

5.10 VRow.cpp File Reference

```
#include <qxml.h>
#include <qdom.h>
#include <qlistview.h>
#include "VRow.h"
#include "VObject.h"
#include "DialogThread.h"
#include "VSelectionGUI.h"
```

5.11 VRow.h File Reference

```
#include <qxml.h>
#include <qdom.h>
#include <qlistview.h>
#include <qstring.h>
#include "VCell.h"
#include "VSelectionGUI.h"
```

Compounds

- class **VRow**

*This class is the abstract representation of the <row> tag, with all the data inside it. (Same as **VObject** (p. 14), but specialized for the <row> tags).*

5.12 VSelectionGUI.cpp File Reference

```
#include <iostream.h>
#include <qcombobox.h>
#include <qxml.h>
#include <qdom.h>
#include <qpushbutton.h>
#include <qmessagebox.h>
#include <qlistview.h>
#include <qgroupbox.h>
#include <qtextbrowser.h>
#include <qlist.h>
#include <qthread.h>
#include <string.h>
#include "VObject.h"
#include "VRow.h"
#include "VCell.h"
#include "VSelectionGUI.h"
#include "HelpGUI.h"
#include "Interface.h"
```

5.13 VSelectionGUI.h File Reference

```
#include <qxml.h>
#include <qdom.h>
#include <qlist.h>
#include <qlistview.h>
#include <qthread.h>
#include "SelectionGUI.h"
#include "HelpGUI.h"
#include "VObject.h"
#include "VRow.h"
#include "VCell.h"
#include "Interface.h"
```

Compounds

- class **VSelectionGUI**

This class is the main widget.

Index

- ~VCell
 - VCell, 11
 - ~VObject
 - VObject, 14
 - ~VRow
 - VRow, 18
 - ~VSelectionGUI
 - VSelectionGUI, 22
 - accessOk
 - DialogThread, 8
 - addChild
 - VObject, 15
 - VRow, 18
 - addListSelected
 - VSelectionGUI, 22
 - addListViewItem
 - VSelectionGUI, 22
 - applyAction
 - VSelectionGUI, 22
 - aux
 - DialogThread, 8
 - boolean
 - Interface.h, 28
 - categoryChanged
 - VSelectionGUI, 22
 - categoryNumber
 - VSelectionGUI, 24
 - cellAt
 - VRow, 18
 - cells
 - VRow, 20
 - childAt
 - VObject, 15
 - VRow, 18
 - children
 - VRow, 20
 - childrensList
 - VObject, 16
 - clearChildren
 - VObject, 15
 - VRow, 18
 - codeColumn
 - VSelectionGUI, 24
 - codeIndexList
 - VSelectionGUI, 24
 - codeList
 - VSelectionGUI, 24
 - countCells
 - VRow, 18
 - countChildren
 - VObject, 15
 - VRow, 18
 - countRows
 - VObject, 15
 - countWords
 - VCell, 11
 - currentChoice
 - VSelectionGUI, 24
 - currentChoiceList
 - VSelectionGUI, 24
 - defaultSelected
 - VRow, 20
 - descriptionColumn
 - VSelectionGUI, 24
 - descriptionList
 - VSelectionGUI, 24
 - DialogThread
 - DialogThread, 8
 - DialogThread, 7
 - accessOk, 8
 - aux, 8
 - DialogThread, 8
 - getChoice, 8
 - getName, 8
 - run, 8
 - selectionGUI, 8
 - xmlStream, 8
 - dialogThread
 - Interface.cpp, 27
 - DialogThread.cpp, 25
 - DialogThread.h, 26
 - getChoice
 - DialogThread, 8
 - VSelectionGUI, 22
 - getName
-

- DialogThread, 8
- VSelectionGUI, 22
- getQListViewItem
 - VRow, 18
- getSelection
 - VObject, 15
 - VRow, 18
- getString
 - VCell, 11
- getText
 - VObject, 15
- getType
 - VObject, 15
 - VRow, 19
- gui_choice
 - Interface.cpp, 27
 - Interface.h, 28
- gui_clean
 - Interface.cpp, 27
 - Interface.h, 28
- gui_init
 - Interface.cpp, 27
 - Interface.h, 28
- gui_name
 - Interface.cpp, 27
 - Interface.h, 28
- hasChildren
 - VObject, 15
 - VRow, 19
- helpAction
 - VSelectionGUI, 23
- helpText
 - VSelectionGUI, 24
- Interface.cpp, 27
 - dialogThread, 27
 - gui_choice, 27
 - gui_clean, 27
 - gui_init, 27
 - gui_name, 27
- Interface.h, 28
 - boolean, 28
 - gui_choice, 28
 - gui_clean, 28
 - gui_init, 28
 - gui_name, 28
 - NO, 28
 - vect_code_names, 28
 - YES, 28
- isDefault
 - VRow, 19
- isSelected
 - VRow, 19
- listOfListviewRows
 - VSelectionGUI, 24
- listOfRows
 - VSelectionGUI, 24
- main
 - main.cpp, 30
- main.cpp, 30
 - main, 30
- moduleChanged
 - VSelectionGUI, 23
- moduleName
 - VSelectionGUI, 24
- myHelp
 - VSelectionGUI, 24
- myMutex
 - VSelectionGUI, 24
- newChoice
 - VSelectionGUI, 24
- NO
 - Interface.h, 28
- num_code_names
 - vect_code_names, 13
- objectText
 - VObject, 16
- objectType
 - VObject, 16
 - VRow, 20
- processChanged
 - VSelectionGUI, 23
- processNumber
 - VSelectionGUI, 24
- QThread, 9
- resetAction
 - VSelectionGUI, 23
- resetSelection
 - VObject, 15
 - VRow, 19
- rootObject
 - VSelectionGUI, 24
- rowAt
 - VObject, 15
- rowList
 - VObject, 16
- run
 - DialogThread, 8
- selected
 - VRow, 20
- SelectionGUI, 10

- selectionGUI
 - DialogThread, 8
- setDefault
 - VRow, 19
- setSelected
 - VRow, 19
- setText
 - VObject, 16
- setType
 - VObject, 16
 - VRow, 19
- setXML
 - VSelectionGUI, 23
- single_code_name
 - vect_code_names, 13
- unitsColumn
 - VSelectionGUI, 24
- VCell, 11
 - ~VCell, 11
 - countWords, 11
 - getString, 11
 - VCell, 11
 - wordAt, 11
 - words, 12
- VCell.cpp, 31
- VCell.h, 32
- vect_code_names, 13
 - Interface.h, 28
 - num_code_names, 13
 - single_code_name, 13
- VObject, 14
 - ~VObject, 14
 - addChild, 15
 - childAt, 15
 - childrenList, 16
 - clearChildren, 15
 - countChildren, 15
 - countRows, 15
 - getSelection, 15
 - getText, 15
 - getType, 15
 - hasChildren, 15
 - objectText, 16
 - objectType, 16
 - resetSelection, 15
 - rowAt, 15
 - rowList, 16
 - setText, 16
 - setType, 16
 - VObject, 14
- VObject.cpp, 33
- VObject.h, 34
- VRow, 17
 - ~VRow, 18
 - addChild, 18
 - cellAt, 18
 - cells, 20
 - childAt, 18
 - children, 20
 - clearChildren, 18
 - countCells, 18
 - countChildren, 18
 - defaultSelected, 20
 - getQListViewItem, 18
 - getSelection, 18
 - getType, 19
 - hasChildren, 19
 - isDefault, 19
 - isSelected, 19
 - objectType, 20
 - resetSelection, 19
 - selected, 20
 - setDefault, 19
 - setSelected, 19
 - setType, 19
 - VRow, 17
- VRow.cpp, 35
- VRow.h, 36
- VSelectionGUI
 - VSelectionGUI, 22
- VSelectionGUI, 21
 - ~VSelectionGUI, 22
 - addListSelected, 22
 - addListViewItem, 22
 - applyAction, 22
 - categoryChanged, 22
 - categoryNumber, 24
 - codeColumn, 24
 - codeIndexList, 24
 - codeList, 24
 - currentChoice, 24
 - currentChoiceList, 24
 - descriptionColumn, 24
 - descriptionList, 24
 - getChoice, 22
 - getName, 22
 - helpAction, 23
 - helpText, 24
 - listOfListViewRows, 24
 - listOfRows, 24
 - moduleChanged, 23
 - moduleName, 24
 - myHelp, 24
 - myMutex, 24
 - newChoice, 24
 - processChanged, 23

- processNumber, 24
- resetAction, 23
- rootObject, 24
- setXML, 23
- unitsColumn, 24
- VSelectionGUI, 22
- VSelectionGUI.cpp, 37
- VSelectionGUI.h, 38

- wordAt
 - VCell, 11
- words
 - VCell, 12

- xmlStream
 - DialogThread, 8

- YES
 - Interface.h, 28