
SiQt Documentation

Release 0.3.0

Symerio

Jun 01, 2018

Contents:

1	Installation	3
2	Unit tests	5
2.1	API Reference	5

Convenience tools for building PyQt5 / PySide2 based GUI applications.

See the documentation for more details: <http://siqt.readthedocs.io/en/latest/>

CHAPTER 1

Installation

This package requires Python 3.5+ and either `PyQt5` or `PySide2`.

It can be installed with,

```
pip install SiQt
```


The test suite can be run with,

```
py.test -sv .
```

2.1 API Reference

2.1.1 Definitions

<i>SiQt.definitions.SiQtMixin</i>	A mixin to extend <code>QtWidgets.QMainWindow</code>
<i>SiQt.definitions.SiQtItem</i> (qtobj[, position, ...])	A thin wrapper around any <code>QObject</code>

SiQt.definitions.SiQtMixin

class `SiQt.definitions.SiQtMixin`

A mixin to extend `QtWidgets.QMainWindow`

This class adds a few convenient functions to `QtWidgets.QMainWindow` to add actions and generate menus

`__init__()`

Initialize self. See `help(type(self))` for accurate signature.

Methods

`add_actions(target, actions)`

`create_action(text[, slot, shortcut, icon, ...])`

Continued on next page

Table 2 – continued from previous page

<code>menu_generator(name, label, elements)</code>	Generate menu
<code>set_dep_flag_recursive(key, value[, sync])</code>	Change status for a dependent field, and pull all the fields that are dependent on it.

menu_generator (*name, label, elements*)

Generate menu

set_dep_flag_recursive (*key, value, sync=False*)

Change status for a dependent field, and pull all the fields that are dependent on it.

SiQt.definitions.SiqtItem

class SiQt.definitions.**SiqtItem** (*qobj, position=None, depends=[], dtype=<class 'str'>, layout=None, **args*)

A thin wrapper around any QObject

Supports dependencies, a straightforward way of getting / setting value.

__init__ (*qobj, position=None, depends=[], dtype=<class 'str'>, layout=None, **args*)

Initialize self. See help(type(self)) for accurate signature.

Methods

<code>__init__(qobj[, position, depends, dtype, ...])</code>	Initialize self.
<code>clear()</code>	
<code>copy()</code>	
<code>fromkeys</code>	Returns a new dict with keys from iterable and values equal to value.
<code>get(k[,d])</code>	
<code>items()</code>	
<code>keys()</code>	
<code>pop(k[,d])</code>	If key is not found, d is returned if given, otherwise KeyError is raised
<code>popitem()</code>	2-tuple; but raise KeyError if D is empty.
<code>set_choices(choices)</code>	
<code>set_text(value)</code>	
<code>set_value(value)</code>	
<code>setdefault(k[,d])</code>	
<code>update([E,]**F)</code>	If E is present and has a .keys() method, then does: for k in E: D[k] = E[k] If E is present and lacks a .keys() method, then does: for k, v in E: D[k] = v In either case, this is followed by: for k in F: D[k] = F[k]
<code>values()</code>	
<code>with_layout(layout)</code>	Pre-initialize SiqtItem with layout

Attributes

<code>value</code>	Get the value of the QObject
--------------------	------------------------------

value

Get the value of the QObject

classmethod with_layout (*layout*)

Pre-initialize SiQtItem with layout

2.1.2 Dependency resolution

SiQt.dep_resolv.

show_qt_control_element(*el*)

SiQt.dep_resolv.sync_gui([*lock*, *update*, ...]) Process GUI events and determine which elements should be enabled

SiQt.dep_resolv.

check_depflags(*dep_flags*, *expr*)

Check if the boolean expression *expr* is verified in *dep_flags* Two possibilities for *expr*: ['flag1', 'flag2', etc] a boolean expression 'flag1 and (flag2 or not flag3)'

SiQt.dep_resolv.

calculate_dependencies(*self*)

Determine which GUI elements should be enabled

SiQt.dep_resolv.show_qt_control_element

`SiQt.dep_resolv.show_qt_control_element` (*el*)

SiQt.dep_resolv.sync_gui

`SiQt.dep_resolv.sync_gui` (*lock*=[], *update*=[], *view_mode*=None, *background*=False)

Process GUI events and determine which elements should be enabled

SiQt.dep_resolv.check_depflags

`SiQt.dep_resolv.check_depflags` (*dep_flags*, *expr*)

Check if the boolean expression *expr* is verified in *dep_flags* Two possibilities for *expr*:

['flag1', 'flag2', etc]

a boolean expression 'flag1 and (flag2 or not flag3)'

SiQt.dep_resolv.calculate_dependencies

`SiQt.dep_resolv.calculate_dependencies` (*self*, *verbose*=False, *initialize*=False)

Determine which GUI elements should be enabled

2.1.3 Utils

SiQt.deployment._resource_path(*relative*) Normalisation of the ressource path for PyInstaller

SiQt.matplotlib.

NavigationToolbar(*canvas*, *parent*)

SiQt.deployment._resource_path

SiQt.deployment._resource_path (*relative*)
Normalisation of the ressource path for PyInstaller

SiQt.matplotlib.NavigationToolbar

class SiQt.matplotlib.NavigationToolbar (*canvas, parent, coordinates=True*)

__init__ (*canvas, parent, coordinates=True*)
coordinates: should we show the coordinates on the right?

Methods

<code>__init__(canvas, parent[, coordinates])</code>	coordinates: should we show the coordinates on the right?
<code>acceptDrops(self)</code>	
<code>accessibleDescription(self)</code>	
<code>accessibleName(self)</code>	
<code>actionAt(self, QPoint)</code>	actionAt(self, int, int) -> QAction
<code>actionEvent(self, QActionEvent)</code>	
<code>actionGeometry(self, QAction)</code>	
<code>actions(self)</code>	
<code>activateWindow(self)</code>	
<code>addAction(self, QAction)</code>	addAction(self, str) -> QAction addAction(self, QIcon, str) -> QAction addAction(self, str, PYQT_SLOT) -> QAction addAction(self, QIcon, str, PYQT_SLOT) -> QAction
<code>addActions(self, object)</code>	
<code>addSeparator(self)</code>	
<code>addWidget(self, QWidget)</code>	
<code>adjustSize(self)</code>	
<code>allowedAreas(self)</code>	
<code>autoFillBackground(self)</code>	
<code>back(*args)</code>	move back up the view lim stack
<code>backgroundRole(self)</code>	
<code>baseSize(self)</code>	
<code>blockSignals(self, bool)</code>	
<code>changeEvent(self, QEvent)</code>	
<code>childAt(self, QPoint)</code>	childAt(self, int, int) -> QWidget
<code>childEvent(self, QChildEvent)</code>	
<code>children(self)</code>	
<code>childrenRect(self)</code>	
<code>childrenRegion(self)</code>	
<code>clear(self)</code>	
<code>clearFocus(self)</code>	
<code>clearMask(self)</code>	
<code>close(self)</code>	
<code>closeEvent(self, QCloseEvent)</code>	

Continued on next page

Table 7 – continued from previous page

<code>colorCount(self)</code>	
<code>configure_subplots()</code>	
<code>connectNotify(self, QMetaMethod)</code>	
<code>contentsMargins(self)</code>	
<code>contentsRect(self)</code>	
<code>contextMenuEvent(self, QContextMenuEvent)</code>	
<code>contextMenuPolicy(self)</code>	
<code>create(self, window, initializeWindow, ...)</code>	
<code>createWindowContainer(QWindow, parent, ...)</code>	
<code>cursor(self)</code>	
<code>customEvent(self, QEvent)</code>	
<code>deleteLater(self)</code>	
<code>depth(self)</code>	
<code>destroy(self, destroyWindow, destroySubWindows)</code>	
<code>devType(self)</code>	
<code>devicePixelRatio(self)</code>	
<code>devicePixelRatioF(self)</code>	
<code>devicePixelRatioFScale()</code>	
<code>disconnect(self)</code>	
<code>disconnectNotify(self, QMetaMethod)</code>	
<code>dragEnterEvent(self, QDragEnterEvent)</code>	
<code>dragLeaveEvent(self, QDragLeaveEvent)</code>	
<code>dragMoveEvent(self, QDragMoveEvent)</code>	
<code>drag_pan(event)</code>	Callback for dragging in pan/zoom mode.
<code>drag_zoom(event)</code>	Callback for dragging in zoom mode.
<code>draw()</code>	Redraw the canvases, update the locators.
<code>draw_rubberband(event, x0, y0, x1, y1)</code>	Draw a rectangle rubberband to indicate zoom limits.
<code>dropEvent(self, QDropEvent)</code>	
<code>dumpObjectInfo(self)</code>	
<code>dumpObjectTree(self)</code>	
<code>dynamicPropertyNames(self)</code>	
<code>dynamic_update()</code>	
<code>edit_parameters()</code>	
<code>effectiveWinId(self)</code>	
<code>ensurePolished(self)</code>	
<code>enterEvent(self, QEvent)</code>	
<code>event(self, QEvent)</code>	
<code>eventFilter(self, QObject, QEvent)</code>	
<code>find(sip.voidptr)</code>	
<code>findChild(self, type, name, options, ...)</code>	<code>findChild(self, Tuple, name: str = "", options: Union[Qt.FindChildOptions, Qt.FindChildOption] = Qt.FindChildrenRecursively) -> QObject</code>

Continued on next page

Table 7 – continued from previous page

<code>findChildren(self, type, name, options, ...)</code>	<code>findChildren(self, Tuple, name: str = '', options: Union[Qt.FindChildOptions, Qt.FindChildOption] = Qt.FindChildrenRecursively) -> List[QObject]</code> <code>findChildren(self, type, QRegExp, options: Union[Qt.FindChildOptions, Qt.FindChildOption] = Qt.FindChildrenRecursively) -> List[QObject]</code> <code>findChildren(self, Tuple, QRegExp, options: Union[Qt.FindChildOptions, Qt.FindChildOption] = Qt.FindChildrenRecursively) -> List[QObject]</code> <code>findChildren(self, type, QRegularExpression, options: Union[Qt.FindChildOptions, Qt.FindChildOption] = Qt.FindChildrenRecursively) -> List[QObject]</code> <code>findChildren(self, Tuple, QRegularExpression, options: Union[Qt.FindChildOptions, Qt.FindChildOption] = Qt.FindChildrenRecursively) -> List[QObject]</code>
<code>focusInEvent(self, QFocusEvent)</code>	
<code>focusNextChild(self)</code>	
<code>focusNextPrevChild(self, bool)</code>	
<code>focusOutEvent(self, QFocusEvent)</code>	
<code>focusPolicy(self)</code>	
<code>focusPreviousChild(self)</code>	
<code>focusProxy(self)</code>	
<code>focusWidget(self)</code>	
<code>font(self)</code>	
<code>fontInfo(self)</code>	
<code>fontMetrics(self)</code>	
<code>foregroundRole(self)</code>	
<code>forward(*args)</code>	Move forward in the view lim stack.
<code>frameGeometry(self)</code>	
<code>frameSize(self)</code>	
<code>geometry(self)</code>	
<code>getContentsMargins(self)</code>	
<code>grab(self, rectangle)</code>	
<code>grabGesture(self, Qt.GestureType, flags, ...)</code>	
<code>grabKeyboard(self)</code>	
<code>grabMouse(self)</code>	<code>grabMouse(self, Union[QCursor, Qt.CursorShape])</code>
<code>grabShortcut(self, Union[QKeySequence, ...])</code>	
<code>graphicsEffect(self)</code>	
<code>graphicsProxyWidget(self)</code>	
<code>hasFocus(self)</code>	
<code>hasHeightForWidth(self)</code>	
<code>hasMouseTracking(self)</code>	
<code>hasTabletTracking(self)</code>	
<code>height(self)</code>	
<code>heightForWidth(self, int)</code>	
<code>heightMM(self)</code>	
<code>hide(self)</code>	
<code>hideEvent(self, QHideEvent)</code>	
<code>home(*args)</code>	Restore the original view.
<code>iconSize(self)</code>	
<code>inherits(self, str)</code>	

Continued on next page

Table 7 – continued from previous page

<code>initPainter(self, QPainter)</code>
<code>initStyleOption(self, QStyleOptionToolBar)</code>
<code>inputMethodEvent(self, QInputMethodEvent)</code>
<code>inputMethodHints(self)</code>
<code>inputMethodQuery(self, Qt.InputMethodQuery)</code>
<code>insertAction(self, QAction, QAction)</code>
<code>insertActions(self, QAction, Iterable[QAction])</code>
<code>insertSeparator(self, QAction)</code>
<code>insertWidget(self, QAction, QWidget)</code>
<code>installEventFilter(self, QObject)</code>
<code>isActiveWindow(self)</code>
<code>isAncestorOf(self, QWidget)</code>
<code>isAreaAllowed(self, Qt.ToolBarArea)</code>
<code>isEnabled(self)</code>
<code>isEnabledTo(self, QWidget)</code>
<code>isFloatable(self)</code>
<code>isFloating(self)</code>
<code>isFullScreen(self)</code>
<code>isHidden(self)</code>
<code>isLeftToRight(self)</code>
<code>isMaximized(self)</code>
<code>isMinimized(self)</code>
<code>isModal(self)</code>
<code>isMovable(self)</code>
<code>isRightToLeft(self)</code>
<code>isSignalConnected(self, QMetaMethod)</code>
<code>isVisible(self)</code>
<code>isVisibleTo(self, QWidget)</code>
<code>isWidgetType(self)</code>
<code>isWindow(self)</code>
<code>isWindowModified(self)</code>
<code>isWindowType(self)</code>
<code>keyPressEvent(self, QKeyEvent)</code>
<code>keyReleaseEvent(self, QKeyEvent)</code>
<code>keyboardGrabber()</code>
<code>killTimer(self, int)</code>
<code>layout(self)</code>
<code>layoutDirection(self)</code>
<code>leaveEvent(self, QEvent)</code>
<code>locale(self)</code>
<code>logicalDpiX(self)</code>
<code>logicalDpiY(self)</code>
<code>lower(self)</code>
<code>mapFrom(self, QWidget, QPoint)</code>
<code>mapFromGlobal(self, QPoint)</code>
<code>mapFromParent(self, QPoint)</code>
<code>mapTo(self, QWidget, QPoint)</code>
<code>mapToGlobal(self, QPoint)</code>

Continued on next page

Table 7 – continued from previous page

<code>mapToParent(self, QPoint)</code>	
<code>mask(self)</code>	
<code>maximumHeight(self)</code>	
<code>maximumSize(self)</code>	
<code>maximumWidth(self)</code>	
<code>metaObject(self)</code>	
<code>metric(self, QPaintDevice.PaintDeviceMetric)</code>	
<code>minimumHeight(self)</code>	
<code>minimumSize(self)</code>	
<code>minimumSizeHint(self)</code>	
<code>minimumWidth(self)</code>	
<code>mouseDoubleClickEvent(self, QMouseEvent)</code>	
<code>mouseGrabber()</code>	
<code>mouseMoveEvent(self, QMouseEvent)</code>	
<code>mousePressEvent(self, QMouseEvent)</code>	
<code>mouseReleaseEvent(self, QMouseEvent)</code>	
<code>mouse_move(event)</code>	
<code>move(self, QPoint)</code>	<code>move(self, int, int)</code>
<code>moveEvent(self, QMoveEvent)</code>	
<code>moveToThread(self, QThread)</code>	
<code>nativeEvent(self, Union[QByteArray, bytes, ...])</code>	
<code>nativeParentWidget(self)</code>	
<code>nextInFocusChain(self)</code>	
<code>normalGeometry(self)</code>	
<code>objectName(self)</code>	
<code>orientation(self)</code>	
<code>overrideWindowFlags(self, ...)</code>	
<code>overrideWindowState(self, ...)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, QPaintEvent)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>pan(*args)</code>	Activate the pan/zoom tool.
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>press(event)</code>	Called whenever a mouse button is pressed.
<code>press_pan(event)</code>	Callback for mouse button press in pan/zoom mode.
<code>press_zoom(event)</code>	Callback for mouse button press in zoom to rect mode.
<code>previousInFocusChain(self)</code>	
<code>property(self, str)</code>	
<code>push_current()</code>	Push the current view limits and position onto the stack.
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	

Continued on next page

Table 7 – continued from previous page

<code>receivers(self, PYQT_SIGNAL)</code>	
<code>rect(self)</code>	
<code>release(event)</code>	Callback for mouse button release.
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	
<code>releaseShortcut(self, int)</code>	
<code>release_pan(event)</code>	Callback for mouse button release in pan/zoom mode.
<code>release_zoom(event)</code>	Callback for mouse button release in zoom to rect mode.
<code>removeAction(self, QAction)</code>	
<code>removeEventFilter(self, QObject)</code>	
<code>remove_rubberband()</code>	Remove the rubberband.
<code>render(self, QPaintDevice, targetOffset, ...)</code>	<code>render(self, QPainter, targetOffset: QPoint = QPoint(), sourceRegion: QRegion = QRegion(), flags: Union[QWidget.RenderFlags, QWidget.RenderFlag] = QWidget.RenderFlags(QWidget.DrawWindowBackground QWidget.DrawChildren))</code>
<code>repaint(self)</code>	<code>repaint(self, int, int, int, int) repaint(self, QRect) repaint(self, QRegion)</code>
<code>resize(self, QSize)</code>	<code>resize(self, int, int)</code>
<code>resizeEvent(self, QResizeEvent)</code>	
<code>restoreGeometry(self, Union[QByteArray, ...])</code>	
<code>saveGeometry(self)</code>	
<code>save_figure(*args)</code>	Save the current figure.
<code>scroll(self, int, int)</code>	<code>scroll(self, int, int, QRect)</code>
<code>sender(self)</code>	
<code>senderSignalIndex(self)</code>	
<code>setAcceptDrops(self, bool)</code>	
<code>setAccessibleDescription(self, str)</code>	
<code>setAccessibleName(self, str)</code>	
<code>setAllowedAreas(self, Union[Qt.ToolBarAreas, ...])</code>	
<code>setAttribute(self, Qt.WidgetAttribute, on)</code>	
<code>setAutoFillBackground(self, bool)</code>	
<code>setBackgroundRole(self, QPalette.ColorRole)</code>	
<code>setBaseSize(self, int, int)</code>	<code>setBaseSize(self, QSize)</code>
<code>setContentsMargins(self, int, int, int, int)</code>	<code>setContentsMargins(self, QMargins)</code>
<code>setContextMenuPolicy(self, Qt.ContextMenuPolicy)</code>	
<code>setCursor(self, Union[QCursor, Qt.CursorShape])</code>	
<code>setDisabled(self, bool)</code>	
<code>setEnabled(self, bool)</code>	
<code>setFixedHeight(self, int)</code>	
<code>setFixedSize(self, QSize)</code>	<code>setFixedSize(self, int, int)</code>
<code>setFixedWidth(self, int)</code>	
<code>setFloatable(self, bool)</code>	
<code>setFocus(self)</code>	<code>setFocus(self, Qt.FocusReason)</code>
<code>setFocusPolicy(self, Qt.FocusPolicy)</code>	
<code>setFocusProxy(self, QWidget)</code>	

Continued on next page

Table 7 – continued from previous page

setFont(self, QFont)	
setForegroundRole(self, QPalette.ColorRole)	
setGeometry(self, QRect)	setGeometry(self, int, int, int, int)
setGraphicsEffect(self, QGraphicsEffect)	
setHidden(self, bool)	
setIconSize(self, QSize)	
setInputMethodHints(self, ...)	
setLayout(self, QLayout)	
setLayoutDirection(self, Qt.LayoutDirection)	
setLocale(self, QLocale)	
setMask(self, QPixmap)	setMask(self, QRegion)
setMaximumHeight(self, int)	
setMaximumSize(self, int, int)	setMaximumSize(self, QSize)
setMaximumWidth(self, int)	
setMinimumHeight(self, int)	
setMinimumSize(self, int, int)	setMinimumSize(self, QSize)
setMinimumWidth(self, int)	
setMouseTracking(self, bool)	
setMovable(self, bool)	
setObjectName(self, str)	
setOrientation(self, Qt.Orientation)	
setPalette(self, QPalette)	
setParent(self, QWidget)	setParent(self, QWidget, Union[Qt.WindowFlags, Qt.WindowType])
setProperty(self, str, Any)	
setShortcutAutoRepeat(self, int, enabled)	
setShortcutEnabled(self, int, enabled)	
setSizeIncrement(self, int, int)	setSizeIncrement(self, QSize)
setSizePolicy(self, QSizePolicy)	setSizePolicy(self, QSizePolicy.Policy, QSizePolicy.Policy)
setStatusTip(self, str)	
setStyle(self, QStyle)	
setStyleSheet(self, str)	
setTabOrder(QWidget, QWidget)	
setTabletTracking(self, bool)	
setToolButtonStyle(self, Qt.ToolButtonStyle)	
setToolTip(self, str)	
setToolTipDuration(self, int)	
setUpdatesEnabled(self, bool)	
setVisible(self, bool)	
setWhatsThis(self, str)	
setWindowFilePath(self, str)	
setWindowFlag(self, Qt.WindowType, on)	
setWindowFlags(self, Union[Qt.WindowFlags, ...])	
setWindowIcon(self, QIcon)	
setWindowIconText(self, str)	

Continued on next page

Table 7 – continued from previous page

<code>setWindowModality(self, Qt.WindowModality)</code>	
<code>setWindowModified(self, bool)</code>	
<code>setWindowOpacity(self, float)</code>	
<code>setWindowRole(self, str)</code>	
<code>setWindowState(self, Union[Qt.WindowStates, ...])</code>	
<code>setWindowTitle(self, str)</code>	
<code>set_cursor(cursor)</code>	Set the current cursor to one of the <code>Cursors</code> enums values.
<code>set_history_buttons()</code>	Enable or disable the back/forward button.
<code>set_message(s)</code>	Display a message on toolbar or in status bar.
<code>sharedPainter(self)</code>	
<code>show(self)</code>	
<code>showEvent(self, QShowEvent)</code>	
<code>showFullScreen(self)</code>	
<code>showMaximized(self)</code>	
<code>showMinimized(self)</code>	
<code>showNormal(self)</code>	
<code>signalsBlocked(self)</code>	
<code>size(self)</code>	
<code>sizeHint(self)</code>	
<code>sizeIncrement(self)</code>	
<code>sizePolicy(self)</code>	
<code>stackUnder(self, QWidget)</code>	
<code>startTimer(self, int, timerType)</code>	
<code>statusTip(self)</code>	
<code>style(self)</code>	
<code>stylesheet(self)</code>	
<code>tabletEvent(self, QTabletEvent)</code>	
<code>testAttribute(self, Qt.WidgetAttribute)</code>	
<code>thread(self)</code>	
<code>timerEvent(self, QTimerEvent)</code>	
<code>toggleViewAction(self)</code>	
<code>toolButtonStyle(self)</code>	
<code>toolTip(self)</code>	
<code>toolTipDuration(self)</code>	
<code>tr(self, str, disambiguation, n)</code>	
<code>underMouse(self)</code>	
<code>ungrabGesture(self, Qt.GestureType)</code>	
<code>unsetCursor(self)</code>	
<code>unsetLayoutDirection(self)</code>	
<code>unsetLocale(self)</code>	
<code>update()</code>	Reset the axes stack.
<code>updateGeometry(self)</code>	
<code>updateMicroFocus(self)</code>	
<code>updatesEnabled(self)</code>	
<code>visibleRegion(self)</code>	
<code>whatsThis(self)</code>	
<code>wheelEvent(self, QWheelEvent)</code>	

Continued on next page

Table 7 – continued from previous page

<code>widgetForAction(self, QAction)</code>	
<code>width(self)</code>	
<code>widthMM(self)</code>	
<code>winId(self)</code>	
<code>window(self)</code>	
<code>windowFilePath(self)</code>	
<code>windowFlags(self)</code>	
<code>windowHandle(self)</code>	
<code>windowIcon(self)</code>	
<code>windowIconText(self)</code>	
<code>windowModality(self)</code>	
<code>windowOpacity(self)</code>	
<code>windowRole(self)</code>	
<code>windowState(self)</code>	
<code>windowTitle(self)</code>	
<code>windowType(self)</code>	
<code>x(self)</code>	
<code>y(self)</code>	
<code>zoom(*args)</code>	Activate zoom to rect mode.

Attributes

<code>DrawChildren</code>	
<code>DrawWindowBackground</code>	
<code>IgnoreMask</code>	
<code>PdmDepth</code>	
<code>PdmDevicePixelRatio</code>	
<code>PdmDevicePixelRatioScaled</code>	
<code>PdmDpiX</code>	
<code>PdmDpiY</code>	
<code>PdmHeight</code>	
<code>PdmHeightMM</code>	
<code>PdmNumColors</code>	
<code>PdmPhysicalDpiX</code>	
<code>PdmPhysicalDpiY</code>	
<code>PdmWidth</code>	
<code>PdmWidthMM</code>	
<code>actionTriggered</code>	<code>actionTriggered(self, QAction) [signal]</code>
<code>customContextMenuRequested</code>	<code>customContextMenuRequested(self, QPoint) [signal]</code>
<code>destroyed</code>	<code>destroyed(self, QObject = None) [signal]</code>
<code>iconSizeChanged</code>	<code>iconSizeChanged(self, QSize) [signal]</code>
<code>message</code>	
<code>movableChanged</code>	<code>movableChanged(self, bool) [signal]</code>
<code>objectNameChanged</code>	<code>objectNameChanged(self, str) [signal]</code>
<code>staticMetaObject</code>	
<code>toolitems</code>	
<code>topLevelChanged</code>	<code>topLevelChanged(self, bool) [signal]</code>
<code>visibilityChanged</code>	<code>visibilityChanged(self, bool) [signal]</code>

Continued on next page

Table 8 – continued from previous page

windowIconChanged	windowIconChanged(self, QIcon) [signal]
windowIconTextChanged	windowIconTextChanged(self, str) [signal]
windowTitleChanged	windowTitleChanged(self, str) [signal]

2.1.4 Widgets

```
SiQt.widgets.DebugInfoWidget(main_window[,
...])
```

```
SiQt.widgets.ProgressBarWidget(main_window)
```

SiQt.widgets.DebugInfoWidget

class SiQt.widgets.DebugInfoWidget (main_window, parent=None)

__init__ (main_window, parent=None)
Initialize self. See help(type(self)) for accurate signature.

Methods

__init__ (main_window[, parent])	Initialize self.
acceptDrops(self)	
accessibleDescription(self)	
accessibleName(self)	
actionEvent(self, QActionEvent)	
actions(self)	
activateWindow(self)	
addAction(self, QAction)	
addActions(self, object)	
addScrollBarWidget(self, QWidget, ...)	
addTopLevelItem(self, QTreeWidgetItem)	
addTopLevelItems(self, Iterable[QTreeWidgetItem])	Iter-
adjustSize(self)	
allColumnsShowFocus(self)	
alternatingRowColors(self)	
autoExpandDelay(self)	
autoFillBackground(self)	
autoScrollMargin(self)	
backgroundRole(self)	
baseSize(self)	
blockSignals(self, bool)	
changeEvent(self, QEvent)	
childAt(self, QPoint)	childAt(self, int, int) -> QWidget
childEvent(self, QChildEvent)	
children(self)	
childrenRect(self)	
childrenRegion(self)	

Continued on next page

Table 10 – continued from previous page

<code>clear(self)</code>
<code>clearFocus(self)</code>
<code>clearMask(self)</code>
<code>clearSelection(self)</code>
<code>close(self)</code>
<code>closeEditor(self, QWidget, ...)</code>
<code>closeEvent(self, QCloseEvent)</code>
<code>closePersistentEditor(self, QTreeWidgetItem, ...)</code>
<code>collapse(self, QModelIndex)</code>
<code>collapseAll(self)</code>
<code>collapseItem(self, QTreeWidgetItem)</code>
<code>colorCount(self)</code>
<code>columnAt(self, int)</code>
<code>columnCount(self)</code>
<code>columnCountChanged(self, int, int)</code>
<code>columnMoved(self)</code>
<code>columnResized(self, int, int, int)</code>
<code>columnViewportPosition(self, int)</code>
<code>columnWidth(self, int)</code>
<code>commitData(self, QWidget)</code>
<code>connectNotify(self, QMetaMethod)</code>
<code>contentsMargins(self)</code>
<code>contentsRect(self)</code>
<code>contextMenuEvent(self, QContextMenuEvent)</code>
<code>contextMenuPolicy(self)</code>
<code>cornerWidget(self)</code>
<code>create(self, window, initializeWindow, ...)</code>
<code>createWindowContainer(QWindow, parent, ...)</code>
<code>currentChanged(self, QModelIndex, QModelIndex)</code>
<code>currentColumn(self)</code>
<code>currentIndex(self)</code>
<code>currentItem(self)</code>
<code>cursor(self)</code>
<code>customEvent(self, QEvent)</code>
<code>dataChanged(self, QModelIndex, QModelIndex, ...)</code>
<code>defaultDropAction(self)</code>
<code>deleteLater(self)</code>
<code>depth(self)</code>
<code>destroy(self, destroyWindow, destroySubWindows)</code>
<code>devType(self)</code>
<code>devicePixelRatio(self)</code>
<code>devicePixelRatioF(self)</code>
<code>devicePixelRatioFScale()</code>
<code>dirtyRegionOffset(self)</code>
<code>disconnect(self)</code>

Continued on next page

Table 10 – continued from previous page

<code>disconnectNotify(self, QMetaMethod)</code>	
<code>dragDropMode(self)</code>	
<code>dragDropOverwriteMode(self)</code>	
<code>dragEnabled(self)</code>	
<code>dragEnterEvent(self, QDragEnterEvent)</code>	
<code>dragLeaveEvent(self, QDragLeaveEvent)</code>	
<code>dragMoveEvent(self, QDragMoveEvent)</code>	
<code>drawBranches(self, QPainter, QRect, QModelIndex)</code>	
<code>drawFrame(self, QPainter)</code>	
<code>drawRow(self, QPainter, ...)</code>	
<code>drawTree(self, QPainter, QRegion)</code>	
<code>dropEvent(self, QDropEvent)</code>	
<code>dropIndicatorPosition(self)</code>	
<code>dropMimeData(self, QTreeWidgetItem, int, ...)</code>	
<code>dumpObjectInfo(self)</code>	
<code>dumpObjectTree(self)</code>	
<code>dynamicPropertyNamees(self)</code>	
<code>edit(self, QModelIndex)</code>	<code>edit(self, QModelIndex, QAbstractItemView.EditTrigger, QEvent) -> bool</code>
<code>editItem(self, QTreeWidgetItem, column)</code>	
<code>editTriggers(self)</code>	
<code>editorDestroyed(self, QObject)</code>	
<code>effectiveWinId(self)</code>	
<code>ensurePolished(self)</code>	
<code>enterEvent(self, QEvent)</code>	
<code>event(self, QEvent)</code>	
<code>eventFilter(self, QObject, QEvent)</code>	
<code>executeDelayedItemsLayout(self)</code>	
<code>expand(self, QModelIndex)</code>	
<code>expandAll(self)</code>	
<code>expandItem(self, QTreeWidgetItem)</code>	
<code>expandToDepth(self, int)</code>	
<code>expandsOnDoubleClick(self)</code>	
<code>find(sip.voidptr)</code>	
<code>findChild(self, type, name, options, ...)</code>	<code>findChild(self, Tuple, name: str = "", options: Union[Qt.FindChildOptions, Qt.FindChildOption] = Qt.FindChildrenRecursively) -> QObject</code>

Continued on next page

Table 10 – continued from previous page

<code>findChildren(self, type, name, options, ...)</code>	<code>findChildren(self, Tuple, name: str = '', options: Union[Qt.FindChildOptions, Qt.FindChildOption] = Qt.FindChildrenRecursively) -> List[QObject]</code> <code>findChildren(self, type, QRegExp, options: Union[Qt.FindChildOptions, Qt.FindChildOption] = Qt.FindChildrenRecursively) -> List[QObject]</code> <code>findChildren(self, Tuple, QRegExp, options: Union[Qt.FindChildOptions, Qt.FindChildOption] = Qt.FindChildrenRecursively) -> List[QObject]</code> <code>findChildren(self, type, QRegularExpression, options: Union[Qt.FindChildOptions, Qt.FindChildOption] = Qt.FindChildrenRecursively) -> List[QObject]</code> <code>findChildren(self, Tuple, QRegularExpression, options: Union[Qt.FindChildOptions, Qt.FindChildOption] = Qt.FindChildrenRecursively) -> List[QObject]</code>
<code>findItems(self, str, Union[Qt.MatchFlags, ...])</code>	
<code>focusInEvent(self, QFocusEvent)</code>	
<code>focusNextChild(self)</code>	
<code>focusNextPrevChild(self, bool)</code>	
<code>focusOutEvent(self, QFocusEvent)</code>	
<code>focusPolicy(self)</code>	
<code>focusPreviousChild(self)</code>	
<code>focusProxy(self)</code>	
<code>focusWidget(self)</code>	
<code>font(self)</code>	
<code>fontInfo(self)</code>	
<code>fontMetrics(self)</code>	
<code>foregroundRole(self)</code>	
<code>frameGeometry(self)</code>	
<code>frameRect(self)</code>	
<code>frameShadow(self)</code>	
<code>frameShape(self)</code>	
<code>frameSize(self)</code>	
<code>frameStyle(self)</code>	
<code>frameWidth(self)</code>	
<code>geometry(self)</code>	
<code>getContentsMargins(self)</code>	
<code>grab(self, rectangle)</code>	
<code>grabGesture(self, Qt.GestureType, flags, ...)</code>	
<code>grabKeyboard(self)</code>	
<code>grabMouse(self)</code>	<code>grabMouse(self, Union[QCursor, Qt.CursorShape])</code>
<code>grabShortcut(self, Union[QKeySequence, ...])</code>	
<code>graphicsEffect(self)</code>	
<code>graphicsProxyWidget(self)</code>	
<code>hasAutoScroll(self)</code>	
<code>hasFocus(self)</code>	
<code>hasHeightForWidth(self)</code>	
<code>hasMouseTracking(self)</code>	
<code>hasTabletTracking(self)</code>	
<code>header(self)</code>	
<code>headerItem(self)</code>	

Continued on next page

Table 10 – continued from previous page

<code>height(self)</code>
<code>heightForWidth(self, int)</code>
<code>heightMM(self)</code>
<code>hide(self)</code>
<code>hideColumn(self, int)</code>
<code>hideEvent(self, QHideEvent)</code>
<code>horizontalOffset(self)</code>
<code>horizontalScrollBar(self)</code>
<code>horizontalScrollBarPolicy(self)</code>
<code>horizontalScrollMode(self)</code>
<code>horizontalScrollbarAction(self, int)</code>
<code>horizontalScrollbarValueChanged(self, int)</code>
<code>iconSize(self)</code>
<code>indentation(self)</code>
<code>indexAbove(self, QModelIndex)</code>
<code>indexAt(self, QPoint)</code>
<code>indexBelow(self, QModelIndex)</code>
<code>indexFromItem(self, QTreeWidgetItem, column)</code>
<code>indexOfTopLevelItem(self, QTreeWidgetItem)</code>
<code>indexRowSizeHint(self, QModelIndex)</code>
<code>indexWidget(self, QModelIndex)</code>
<code>inherits(self, str)</code>
<code>initPainter(self, QPainter)</code>
<code>initStyleOption(self, QStyleOptionFrame)</code>
<code>inputMethodEvent(self, QInputMethodEvent)</code>
<code>inputMethodHints(self)</code>
<code>inputMethodQuery(self, Qt.InputMethodQuery)</code>
<code>insertAction(self, QAction, QAction)</code>
<code>insertActions(self, QAction, Iterable[QAction])</code>
<code>insertTopLevelItem(self, int, QTreeWidgetItem)</code>
<code>insertTopLevelItems(self, int, ...)</code>
<code>installEventFilter(self, QObject)</code>
<code>invisibleRootItem(self)</code>
<code>isActiveWindow(self)</code>
<code>isAncestorOf(self, QWidget)</code>
<code>isAnimated(self)</code>
<code>isColumnHidden(self, int)</code>
<code>isEnabled(self)</code>
<code>isEnabledTo(self, QWidget)</code>
<code>isExpanded(self, QModelIndex)</code>
<code>isFirstColumnSpanned(self, int, QModelIndex)</code>
<code>isFirstItemColumnSpanned(self, QTreeWidgetItem)</code>
<code>isFullScreen(self)</code>

Continued on next page

Table 10 – continued from previous page

isHeaderHidden(self)	
isHidden(self)	
isIndexHidden(self, QModelIndex)	
isLeftToRight(self)	
isMaximized(self)	
isMinimized(self)	
isModal(self)	
isPersistentEditorOpen(self, ...)	
isRightToLeft(self)	
isRowHidden(self, int, QModelIndex)	
isSignalConnected(self, QMetaMethod)	
isSortingEnabled(self)	
isVisible(self)	
isVisibleTo(self, QWidget)	
isWidgetType(self)	
isWindow(self)	
isWindowModified(self)	
isWindowType(self)	
itemAbove(self, QTreeWidgetItem)	
itemAt(self, QPoint)	itemAt(self, int, int) -> QTreeWidgetItem
itemBelow(self, QTreeWidgetItem)	
itemDelegate(self)	itemDelegate(self, QModelIndex) -> QAbstractItemDelegate
itemDelegateForColumn(self, int)	
itemDelegateForRow(self, int)	
itemFromIndex(self, QModelIndex)	
itemWidget(self, QTreeWidgetItem, int)	
itemsExpandable(self)	
keyPressEvent(self, QKeyEvent)	
keyReleaseEvent(self, QKeyEvent)	
keyboardGrabber()	
keyboardSearch(self, str)	
killTimer(self, int)	
layout(self)	
layoutDirection(self)	
leaveEvent(self, QEvent)	
lineWidth(self)	
locale(self)	
logicalDpiX(self)	
logicalDpiY(self)	
lower(self)	
mapFrom(self, QWidget, QPoint)	
mapFromGlobal(self, QPoint)	
mapFromParent(self, QPoint)	
mapTo(self, QWidget, QPoint)	
mapToGlobal(self, QPoint)	
mapToParent(self, QPoint)	
mask(self)	
maximumHeight(self)	
maximumSize(self)	

Continued on next page

Table 10 – continued from previous page

<code>maximumViewportSize(self)</code>	
<code>maximumWidth(self)</code>	
<code>metaObject(self)</code>	
<code>metric(self, QPaintDevice.PaintDeviceMetric)</code>	
<code>midLineWidth(self)</code>	
<code>mimeData(self, Iterable[QTreeWidgetItem])</code>	
<code>mimeTypes(self)</code>	
<code>minimumHeight(self)</code>	
<code>minimumSize(self)</code>	
<code>minimumSizeHint(self)</code>	
<code>minimumWidth(self)</code>	
<code>model(self)</code>	
<code>mouseDoubleClickEvent(self, QMouseEvent)</code>	
<code>mouseGrabber()</code>	
<code>mouseMoveEvent(self, QMouseEvent)</code>	
<code>mousePressEvent(self, QMouseEvent)</code>	
<code>mouseReleaseEvent(self, QMouseEvent)</code>	
<code>move(self, QPoint)</code>	<code>move(self, int, int)</code>
<code>moveCursor(self, ...)</code>	
<code>moveEvent(self, QMoveEvent)</code>	
<code>moveToThread(self, QThread)</code>	
<code>nativeEvent(self, Union[QByteArray, bytes, ...])</code>	
<code>nativeParentWidget(self)</code>	
<code>nextInFocusChain(self)</code>	
<code>normalGeometry(self)</code>	
<code>objectName(self)</code>	
<code>openPersistentEditor(self, QTreeWidgetItem, ...)</code>	
<code>overrideWindowFlags(self, ...)</code>	
<code>overrideWindowState(self, ...)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, QPaintEvent)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>previousInFocusChain(self)</code>	
<code>property(self, str)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	
<code>receivers(self, PYQT_SIGNAL)</code>	
<code>rect(self)</code>	
<code>reexpand(self)</code>	
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	

Continued on next page

Table 10 – continued from previous page

<code>releaseShortcut(self, int)</code>	
<code>removeAction(self, QAction)</code>	
<code>removeEventFilter(self, QObject)</code>	
<code>removeItemWidget(self, QTreeWidgetItem, int)</code>	
<code>render(self, QPaintDevice, targetOffset, ...)</code>	<code>render(self, QPainter, targetOffset: QPoint = QPoint(), sourceRegion: QRegion = QRegion(), flags: Union[QWidget.RenderFlags, QWidget.RenderFlag] = QWidget.RenderFlags(QWidget.DrawWindowBackground QWidget.DrawChildren))</code>
<code>repaint(self)</code>	<code>repaint(self, int, int, int, int) repaint(self, QRect) repaint(self, QRegion)</code>
<code>reset(self)</code>	
<code>resetHorizontalScrollMode(self)</code>	
<code>resetIndentation(self)</code>	
<code>resetVerticalScrollMode(self)</code>	
<code>resize(self, QSize)</code>	<code>resize(self, int, int)</code>
<code>resizeColumnToContents(self, int)</code>	
<code>resizeEvent(self, QResizeEvent)</code>	
<code>restoreGeometry(self, Union[QByteArray, ...])</code>	
<code>rootIndex(self)</code>	
<code>rootIsDecorated(self)</code>	
<code>rowHeight(self, QModelIndex)</code>	
<code>rowsAboutToBeRemoved(self, QModelIndex, int, int)</code>	
<code>rowsInserted(self, QModelIndex, int, int)</code>	
<code>rowsRemoved(self, QModelIndex, int, int)</code>	
<code>saveGeometry(self)</code>	
<code>scheduleDelayedItemsLayout(self)</code>	
<code>scroll(self, int, int)</code>	<code>scroll(self, int, int, QRect)</code>
<code>scrollBarWidgets(self, Union[Qt.Alignment, ...])</code>	
<code>scrollContentsBy(self, int, int)</code>	
<code>scrollDirtyRegion(self, int, int)</code>	
<code>scrollTo(self, QModelIndex, hint)</code>	
<code>scrollToBottom(self)</code>	
<code>scrollToItem(self, QTreeWidgetItem, hint)</code>	
<code>scrollToTop(self)</code>	
<code>selectAll(self)</code>	
<code>selectedIndexes(self)</code>	
<code>selectedItems(self)</code>	
<code>selectionBehavior(self)</code>	
<code>selectionChanged(self, QItemSelection, ...)</code>	
<code>selectionCommand(self, QModelIndex, event)</code>	
<code>selectionMode(self)</code>	
<code>selectionModel(self)</code>	
<code>sender(self)</code>	
<code>senderSignalIndex(self)</code>	
<code>setAcceptDrops(self, bool)</code>	
<code>setAccessibleDescription(self, str)</code>	
<code>setAccessibleName(self, str)</code>	

Continued on next page

Table 10 – continued from previous page

setAllColumnsShowFocus(self, bool)	
setAlternatingRowColors(self, bool)	
setAnimated(self, bool)	
setAttribute(self, Qt.WidgetAttribute, on)	
setAutoExpandDelay(self, int)	
setAutoFillBackground(self, bool)	
setAutoScroll(self, bool)	
setAutoScrollMargin(self, int)	
setBackgroundRole(self, QPalette.ColorRole)	
setBaseSize(self, int, int)	setBaseSize(self, QSize)
setColumnCount(self, int)	
setColumnHidden(self, int, bool)	
setColumnWidth(self, int, int)	
setContentsMargins(self, int, int, int, int)	setContentsMargins(self, QMargins)
setContextMenuPolicy(self, Qt.ContextMenuPolicy)	
setCornerWidget(self, QWidget)	
setCurrentIndex(self, QModelIndex)	
setCurrentItem(self, QTreeWidgetItem)	setCurrentItem(self, QTreeWidgetItem, int) setCurrentItem(self, QTreeWidgetItem, int, Union[QItemSelectionModel.SelectionFlags, QItemSelectionModel.SelectionFlag])
setCursor(self, Union[QCursor, Qt.CursorShape])	
setDefaultDropAction(self, Qt.DropAction)	
setDirtyRegion(self, QRegion)	
setDisabled(self, bool)	
setDragDropMode(self, ...)	
setDragDropOverwriteMode(self, bool)	
setDragEnabled(self, bool)	
setDropIndicatorShown(self, bool)	
setEditTriggers(self, ...)	
setEnabled(self, bool)	
setExpanded(self, QModelIndex, bool)	
setExpandsOnDoubleClick(self, bool)	
setFirstColumnSpanned(self, int, ...)	
setFirstItemColumnSpanned(self, ...)	
setFixedHeight(self, int)	
setFixedSize(self, QSize)	setFixedSize(self, int, int)
setFixedWidth(self, int)	
setFocus(self)	setFocus(self, Qt.FocusReason)
setFocusPolicy(self, Qt.FocusPolicy)	
setFocusProxy(self, QWidget)	
setFont(self, QFont)	
setForegroundRole(self, QPalette.ColorRole)	
setFrameRect(self, QRect)	
setFrameShadow(self, QFrame.Shadow)	
setFrameShape(self, QFrame.Shape)	
setFrameStyle(self, int)	
setGeometry(self, QRect)	setGeometry(self, int, int, int, int)

Continued on next page

Table 10 – continued from previous page

setGraphicsEffect(self, QGraphicsEffect)	
setHeader(self, QHeaderView)	
setHeaderHidden(self, bool)	
setHeaderItem(self, QTreeWidgetItem)	
setHeaderLabel(self, str)	
setHeaderLabels(self, Iterable[str])	
setHidden(self, bool)	
setHorizontalScrollBar(self, QScrollBar)	
setHorizontalScrollBarPolicy(self, ...)	
setHorizontalScrollMode(self, ...)	
setIconSize(self, QSize)	
setIndentation(self, int)	
setIndexWidget(self, QModelIndex, QWidget)	
setInputMethodHints(self, ...)	
setItemDelegate(self, QAbstractItemDelegate)	
setItemDelegateForColumn(self, int, ...)	
setItemDelegateForRow(self, int, ...)	
setItemWidget(self, QTreeWidgetItem, int, ...)	
setItemsExpandable(self, bool)	
setLayout(self, QLayout)	
setLayoutDirection(self, Qt.LayoutDirection)	
setLineWidth(self, int)	
setLocale(self, QLocale)	
setMask(self, QBitmap)	setMask(self, QRegion)
setMaximumHeight(self, int)	
setMaximumSize(self, int, int)	setMaximumSize(self, QSize)
setMaximumWidth(self, int)	
setMidLineWidth(self, int)	
setMinimumHeight(self, int)	
setMinimumSize(self, int, int)	setMinimumSize(self, QSize)
setMinimumWidth(self, int)	
setModel	
setMouseTracking(self, bool)	
setObjectName(self, str)	
setPalette(self, QPalette)	
setParent(self, QWidget)	setParent(self, QWidget, Union[Qt.WindowFlags, Qt.WindowType])
setProperty(self, str, Any)	
setRootIndex(self, QModelIndex)	
setRootIsDecorated(self, bool)	
setRowHidden(self, int, QModelIndex, bool)	
setSelection(self, QRect, ...)	
setSelectionBehavior(self, ...)	
setSelectionMode(self, ...)	
setSelectionModel(self, QItemSelectionModel)	
setShortcutAutoRepeat(self, int, enabled)	
setShortcutEnabled(self, int, enabled)	

Continued on next page

Table 10 – continued from previous page

setSizeAdjustPolicy(self, ...)	
setSizeIncrement(self, int, int)	setSizeIncrement(self, QSize)
setSizePolicy(self, QSizePolicy)	setSizePolicy(self, QSizePolicy.Policy, QSizePolicy.Policy)
setSortingEnabled(self, bool)	
setState(self, QAbstractItemView.State)	
setStatusTip(self, str)	
setStyle(self, QStyle)	
setStyleSheet(self, str)	
setTabKeyNavigation(self, bool)	
setTabOrder(QWidget, QWidget)	
setTabletTracking(self, bool)	
setTextElideMode(self, Qt.TextElideMode)	
setToolTip(self, str)	
setToolTipDuration(self, int)	
setTreePosition(self, int)	
setUniformRowHeights(self, bool)	
setUpdatesEnabled(self, bool)	
setVerticalScrollBar(self, QScrollBar)	
setVerticalScrollBarPolicy(self, ...)	
setVerticalScrollMode(self, ...)	
setViewport(self, QWidget)	
setViewportMargins(self, int, int, int, int)	setViewportMargins(self, QMargins)
setVisible(self, bool)	
setWhatsThis(self, str)	
setWindowFilePath(self, str)	
setWindowFlag(self, Qt.WindowType, on)	
setWindowFlags(self, Union[Qt.WindowFlags, ...])	
setWindowIcon(self, QIcon)	
setWindowIconText(self, str)	
setWindowModality(self, Qt.WindowModality)	
setWindowModified(self, bool)	
setWindowOpacity(self, float)	
setWindowRole(self, str)	
setWindowState(self, Union[Qt.WindowStates, ...])	
setWindowTitle(self, str)	
setWordWrap(self, bool)	
setupViewport(self, QWidget)	
sharedPainter(self)	
show(self)	
showColumn(self, int)	
showDropIndicator(self)	
showEvent(self, QShowEvent)	
showFullScreen(self)	
showMaximized(self)	
showMinimized(self)	
showNormal(self)	

Continued on next page

Table 10 – continued from previous page

signalsBlocked(self)	
size(self)	
sizeAdjustPolicy(self)	
sizeHint(self)	
sizeHintForColumn(self, int)	
sizeHintForIndex(self, QModelIndex)	
sizeHintForRow(self, int)	
sizeIncrement(self)	
sizePolicy(self)	
sortByColumn(self, int, Qt.SortOrder)	
sortColumn(self)	
sortItems(self, int, Qt.SortOrder)	
stackUnder(self, QWidget)	
startDrag(self, Union[Qt.DropActions, ...])	
startTimer(self, int, timerType)	
state(self)	
statusTip(self)	
style(self)	
stylesheet(self)	
supportedDropActions(self)	
tabKeyNavigation(self)	
tabletEvent(self, QTabletEvent)	
takeTopLevelItem(self, int)	
testAttribute(self, Qt.WidgetAttribute)	
textElideMode(self)	
thread(self)	
timerEvent(self, QTimerEvent)	
toolTip(self)	
toolTipDuration(self)	
topLevelItem(self, int)	
topLevelItemCount(self)	
tr(self, str, disambiguation, n)	
treePosition(self)	
underMouse(self)	
ungrabGesture(self, Qt.GestureType)	
uniformRowHeights(self)	
unsetCursor(self)	
unsetLayoutDirection(self)	
unsetLocale(self)	
update(self)	update(self, QModelIndex)
updateEditorData(self)	
updateEditorGeometries(self)	
updateGeometries(self)	
updateGeometry(self)	
updateMicroFocus(self)	
updatesEnabled(self)	
verticalOffset(self)	
verticalScrollBar(self)	
verticalScrollBarPolicy(self)	
verticalScrollMode(self)	

Continued on next page

Table 10 – continued from previous page

<code>verticalScrollbarAction(self, int)</code>
<code>verticalScrollbarValueChanged(self, int)</code>
<code>viewOptions(self)</code>
<code>viewport(self)</code>
<code>viewportEvent(self, QEvent)</code>
<code>viewportMargins(self)</code>
<code>viewportSizeHint(self)</code>
<code>visibleRegion(self)</code>
<code>visualItemRect(self, QTreeWidgetItem)</code>
<code>visualRect(self, QModelIndex)</code>
<code>visualRegionForSelection(self, QItemSelection)</code>
<code>whatsThis(self)</code>
<code>wheelEvent(self, QWheelEvent)</code>
<code>width(self)</code>
<code>widthMM(self)</code>
<code>winId(self)</code>
<code>window(self)</code>
<code>windowFilePath(self)</code>
<code>windowFlags(self)</code>
<code>windowHandle(self)</code>
<code>windowIcon(self)</code>
<code>windowIconText(self)</code>
<code>windowModality(self)</code>
<code>windowOpacity(self)</code>
<code>windowRole(self)</code>
<code>windowState(self)</code>
<code>windowTitle(self)</code>
<code>windowType(self)</code>
<code>wordWrap(self)</code>
<code>x(self)</code>
<code>y(self)</code>

Attributes

<code>AboveItem</code>
<code>AdjustIgnored</code>
<code>AdjustToContents</code>
<code>AdjustToContentsOnFirstShow</code>
<code>AllEditTriggers</code>
<code>AnimatingState</code>
<code>AnyKeyPressed</code>
<code>BelowItem</code>
<code>Box</code>
<code>CollapsingState</code>
<code>ContiguousSelection</code>
<code>CurrentChanged</code>
<code>DoubleClicked</code>

Continued on next page

Table 11 – continued from previous page

DragDrop
DragOnly
DragSelectingState
DraggingState
DrawChildren
DrawWindowBackground
DropOnly
EditKeyPressed
EditingState
EnsureVisible
ExpandingState
ExtendedSelection
HLine
IgnoreMask
InternalMove
MoveDown
MoveEnd
MoveHome
MoveLeft
MoveNext
MovePageDown
MovePageUp
MovePrevious
MoveRight
MoveUp
MultiSelection
NoDragDrop
NoEditTriggers
NoFrame
NoSelection
NoState
OnItem
OnViewport
Panel
PdmDepth
PdmDevicePixelRatio
PdmDevicePixelRatioScaled
PdmDpiX
PdmDpiY
PdmHeight
PdmHeightMM
PdmNumColors
PdmPhysicalDpiX
PdmPhysicalDpiY
PdmWidth
PdmWidthMM
Plain
PositionAtBottom
PositionAtCenter
PositionAtTop

Continued on next page

Table 11 – continued from previous page

Raised	
ScrollPerItem	
ScrollPerPixel	
SelectColumns	
SelectItems	
SelectRows	
SelectedClicked	
Shadow_Mask	
Shape_Mask	
SingleSelection	
StyledPanel	
Sunken	
VLine	
WinPanel	
activated	activated(self, QModelIndex) [signal]
clicked	clicked(self, QModelIndex) [signal]
collapsed	collapsed(self, QModelIndex) [signal]
currentItemChanged	currentItemChanged(self, QTreeWidgetItem, QTreeWidgetItem) [signal]
customContextMenuRequested	customContextMenuRequested(self, QPoint) [signal]
destroyed	destroyed(self, QObject = None) [signal]
doubleClicked	doubleClicked(self, QModelIndex) [signal]
entered	entered(self, QModelIndex) [signal]
expanded	expanded(self, QModelIndex) [signal]
iconSizeChanged	iconSizeChanged(self, QSize) [signal]
itemActivated	itemActivated(self, QTreeWidgetItem, int) [signal]
itemChanged	itemChanged(self, QTreeWidgetItem, int) [signal]
itemClicked	itemClicked(self, QTreeWidgetItem, int) [signal]
itemCollapsed	itemCollapsed(self, QTreeWidgetItem) [signal]
itemDoubleClicked	itemDoubleClicked(self, QTreeWidgetItem, int) [signal]
itemEntered	itemEntered(self, QTreeWidgetItem, int) [signal]
itemExpanded	itemExpanded(self, QTreeWidgetItem) [signal]
itemPressed	itemPressed(self, QTreeWidgetItem, int) [signal]
itemSelectionChanged	itemSelectionChanged(self) [signal]
objectNameChanged	objectNameChanged(self, str) [signal]
pressed	pressed(self, QModelIndex) [signal]
staticMetaObject	
viewportEntered	viewportEntered(self) [signal]
windowIconChanged	windowIconChanged(self, QIcon) [signal]
windowIconTextChanged	windowIconTextChanged(self, str) [signal]
windowTitleChanged	windowTitleChanged(self, str) [signal]

closeEvent (*self*, *QCloseEvent*)

SiQt.widgets.ProgressBarWidget

class SiQt.widgets.ProgressBarWidget (*main_window*, *parent=None*)

`__init__(main_window, parent=None)`
Initialize self. See `help(type(self))` for accurate signature.

Methods

<code>__init__(main_window[, parent])</code>	Initialize self.
<code>acceptDrops(self)</code>	
<code>accessibleDescription(self)</code>	
<code>accessibleName(self)</code>	
<code>actionEvent(self, QActionEvent)</code>	
<code>actions(self)</code>	
<code>activateWindow(self)</code>	
<code>addAction(self, QAction)</code>	
<code>addActions(self, object)</code>	
<code>adjustSize(self)</code>	
<code>autoFillBackground(self)</code>	
<code>backgroundRole(self)</code>	
<code>baseSize(self)</code>	
<code>blockSignals(self, bool)</code>	
<code>changeEvent(self, QEvent)</code>	
<code>childAt(self, QPoint)</code>	<code>childAt(self, int, int) -> QWidget</code>
<code>childEvent(self, QChildEvent)</code>	
<code>children(self)</code>	
<code>childrenRect(self)</code>	
<code>childrenRegion(self)</code>	
<code>clearFocus(self)</code>	
<code>clearMask(self)</code>	
<code>close(self)</code>	
<code>closeEvent(self, QCloseEvent)</code>	
<code>colorCount(self)</code>	
<code>connectNotify(self, QMetaMethod)</code>	
<code>contentsMargins(self)</code>	
<code>contentsRect(self)</code>	
<code>contextMenuEvent(self, QContextMenuEvent)</code>	
<code>contextMenuPolicy(self)</code>	
<code>create(self, window, initializeWindow, ...)</code>	
<code>createWindowContainer(QWindow, parent, ...)</code>	
<code>cursor(self)</code>	
<code>customEvent(self, QEvent)</code>	
<code>deleteLater(self)</code>	
<code>depth(self)</code>	
<code>destroy(self, destroyWindow, destroySubWindows)</code>	
<code>devType(self)</code>	
<code>devicePixelRatio(self)</code>	
<code>devicePixelRatioF(self)</code>	
<code>devicePixelRatioFScale()</code>	
<code>disconnect(self)</code>	
<code>disconnectNotify(self, QMetaMethod)</code>	

Continued on next page

Table 12 – continued from previous page

<code>dragEnterEvent(self, QDragEnterEvent)</code>	
<code>dragLeaveEvent(self, QDragLeaveEvent)</code>	
<code>dragMoveEvent(self, QDragMoveEvent)</code>	
<code>dropEvent(self, QDropEvent)</code>	
<code>dumpObjectInfo(self)</code>	
<code>dumpObjectTree(self)</code>	
<code>dynamicPropertyNames(self)</code>	
<code>effectiveWinId(self)</code>	
<code>ensurePolished(self)</code>	
<code>enterEvent(self, QEvent)</code>	
<code>event(self, QEvent)</code>	
<code>eventFilter(self, QObject, QEvent)</code>	
<code>find(sip.voidptr)</code>	
<code>findChild(self, type, name, options, ...)</code>	<code>findChild(self, Tuple, name: str = "", options: Union[Qt.FindChildOptions, Qt.FindChildOption] = Qt.FindChildrenRecursively) -> QObject</code>
<code>findChildren(self, type, name, options, ...)</code>	<code>findChildren(self, Tuple, name: str = "", options: Union[Qt.FindChildOptions, Qt.FindChildOption] = Qt.FindChildrenRecursively) -> List[QObject]</code> <code>findChildren(self, type, QRegExp, options: Union[Qt.FindChildOptions, Qt.FindChildOption] = Qt.FindChildrenRecursively) -> List[QObject]</code> <code>findChildren(self, Tuple, QRegExp, options: Union[Qt.FindChildOptions, Qt.FindChildOption] = Qt.FindChildrenRecursively) -> List[QObject]</code> <code>findChildren(self, type, QRegularExpression, options: Union[Qt.FindChildOptions, Qt.FindChildOption] = Qt.FindChildrenRecursively) -> List[QObject]</code> <code>findChildren(self, Tuple, QRegularExpression, options: Union[Qt.FindChildOptions, Qt.FindChildOption] = Qt.FindChildrenRecursively) -> List[QObject]</code>
<code>focusInEvent(self, QFocusEvent)</code>	
<code>focusNextChild(self)</code>	
<code>focusNextPrevChild(self, bool)</code>	
<code>focusOutEvent(self, QFocusEvent)</code>	
<code>focusPolicy(self)</code>	
<code>focusPreviousChild(self)</code>	
<code>focusProxy(self)</code>	
<code>focusWidget(self)</code>	
<code>font(self)</code>	
<code>fontInfo(self)</code>	
<code>fontMetrics(self)</code>	
<code>foregroundRole(self)</code>	
<code>frameGeometry(self)</code>	
<code>frameSize(self)</code>	
<code>geometry(self)</code>	
<code>getContentsMargins(self)</code>	
<code>grab(self, rectangle)</code>	
<code>grabGesture(self, Qt.GestureType, flags, ...)</code>	
<code>grabKeyboard(self)</code>	
<code>grabMouse(self)</code>	<code>grabMouse(self, Union[QCursor, Qt.CursorShape])</code>

Continued on next page

Table 12 – continued from previous page

<code>grabShortcut(self, Union[QKeySequence, ...])</code>
<code>graphicsEffect(self)</code>
<code>graphicsProxyWidget(self)</code>
<code>hasFocus(self)</code>
<code>hasHeightForWidth(self)</code>
<code>hasMouseTracking(self)</code>
<code>hasTabletTracking(self)</code>
<code>height(self)</code>
<code>heightForWidth(self, int)</code>
<code>heightMM(self)</code>
<code>hide(self)</code>
<code>hideEvent(self, QHideEvent)</code>
<code>inherits(self, str)</code>
<code>initPainter(self, QPainter)</code>
<code>inputMethodEvent(self, QInputMethodEvent)</code>
<code>inputMethodHints(self)</code>
<code>inputMethodQuery(self, Qt.InputMethodQuery)</code>
<code>insertAction(self, QAction, QAction)</code>
<code>insertActions(self, QAction, Iterable[QAction])</code>
<code>installEventFilter(self, QObject)</code>
<code>isActiveWindow(self)</code>
<code>isAncestorOf(self, QWidget)</code>
<code>isEnabled(self)</code>
<code>isEnabledTo(self, QWidget)</code>
<code>isFullScreen(self)</code>
<code>isHidden(self)</code>
<code>isLeftToRight(self)</code>
<code>isMaximized(self)</code>
<code>isMinimized(self)</code>
<code>isModal(self)</code>
<code>isRightToLeft(self)</code>
<code>isSignalConnected(self, QMetaMethod)</code>
<code>isVisible(self)</code>
<code>isVisibleTo(self, QWidget)</code>
<code>isWidgetType(self)</code>
<code>isWindow(self)</code>
<code>isWindowModified(self)</code>
<code>isWindowType(self)</code>
<code>keyPressEvent(self, QKeyEvent)</code>
<code>keyReleaseEvent(self, QKeyEvent)</code>
<code>keyboardGrabber()</code>
<code>killTimer(self, int)</code>
<code>layout(self)</code>
<code>layoutDirection(self)</code>
<code>leaveEvent(self, QEvent)</code>
<code>locale(self)</code>
<code>logicalDpiX(self)</code>
<code>logicalDpiY(self)</code>

Continued on next page

Table 12 – continued from previous page

<code>lower(self)</code>	
<code>mapFrom(self, QWidget, QPoint)</code>	
<code>mapFromGlobal(self, QPoint)</code>	
<code>mapFromParent(self, QPoint)</code>	
<code>mapTo(self, QWidget, QPoint)</code>	
<code>mapToGlobal(self, QPoint)</code>	
<code>mapToParent(self, QPoint)</code>	
<code>mask(self)</code>	
<code>maximumHeight(self)</code>	
<code>maximumSize(self)</code>	
<code>maximumWidth(self)</code>	
<code>metaObject(self)</code>	
<code>metric(self, QPaintDevice.PaintDeviceMetric)</code>	
<code>minimumHeight(self)</code>	
<code>minimumSize(self)</code>	
<code>minimumSizeHint(self)</code>	
<code>minimumWidth(self)</code>	
<code>mouseDoubleClickEvent(self, QMouseEvent)</code>	
<code>mouseGrabber()</code>	
<code>mouseMoveEvent(self, QMouseEvent)</code>	
<code>mousePressEvent(self, QMouseEvent)</code>	
<code>mouseReleaseEvent(self, QMouseEvent)</code>	
<code>move(self, QPoint)</code>	<code>move(self, int, int)</code>
<code>moveEvent(self, QMoveEvent)</code>	
<code>moveToThread(self, QThread)</code>	
<code>nativeEvent(self, Union[QByteArray, bytes, ...])</code>	
<code>nativeParentWidget(self)</code>	
<code>nextInFocusChain(self)</code>	
<code>normalGeometry(self)</code>	
<code>objectName(self)</code>	
<code>onFinished()</code>	
<code>onProgress(i)</code>	
<code>overrideWindowFlags(self, ...)</code>	
<code>overrideWindowState(self, ...)</code>	
<code>paintEngine(self)</code>	
<code>paintEvent(self, QPaintEvent)</code>	
<code>paintingActive(self)</code>	
<code>palette(self)</code>	
<code>parent(self)</code>	
<code>parentWidget(self)</code>	
<code>physicalDpiX(self)</code>	
<code>physicalDpiY(self)</code>	
<code>pos(self)</code>	
<code>previousInFocusChain(self)</code>	
<code>property(self, str)</code>	
<code>pyqtConfigure(...)</code>	Each keyword argument is either the name of a Qt property or a Qt signal.
<code>raise_(self)</code>	
<code>receivers(self, PYQT_SIGNAL)</code>	

Continued on next page

Table 12 – continued from previous page

<code>rect(self)</code>	
<code>releaseKeyboard(self)</code>	
<code>releaseMouse(self)</code>	
<code>releaseShortcut(self, int)</code>	
<code>removeAction(self, QAction)</code>	
<code>removeEventFilter(self, QObject)</code>	
<code>render(self, QPaintDevice, targetOffset, ...)</code>	<code>render(self, QPainter, targetOffset: QPoint = QPoint(), sourceRegion: QRegion = QRegion(), flags: Union[QWidget.RenderFlags, QWidget.RenderFlag] = QWidget.RenderFlags(QWidget.DrawWindowBackground QWidget.DrawChildren))</code>
<code>repaint(self)</code>	<code>repaint(self, int, int, int, int) repaint(self, QRect) repaint(self, QRegion)</code>
<code>resize(self, QSize)</code>	<code>resize(self, int, int)</code>
<code>resizeEvent(self, QResizeEvent)</code>	
<code>restoreGeometry(self, Union[QByteArray, ...])</code>	
<code>saveGeometry(self)</code>	
<code>scroll(self, int, int)</code>	<code>scroll(self, int, int, QRect)</code>
<code>sender(self)</code>	
<code>senderSignalIndex(self)</code>	
<code>setAcceptDrops(self, bool)</code>	
<code>setAccessibleDescription(self, str)</code>	
<code>setAccessibleName(self, str)</code>	
<code>setAttribute(self, Qt.WidgetAttribute, on)</code>	
<code>setAutoFillBackground(self, bool)</code>	
<code>setBackgroundRole(self, QPalette.ColorRole)</code>	
<code>setBaseSize(self, int, int)</code>	<code>setBaseSize(self, QSize)</code>
<code>setContentsMargins(self, int, int, int, int)</code>	<code>setContentsMargins(self, QMargins)</code>
<code>setContextMenuPolicy(self, Qt.ContextMenuPolicy)</code>	
<code>setCursor(self, Union[QCursor, Qt.CursorShape])</code>	
<code>setDisabled(self, bool)</code>	
<code>setEnabled(self, bool)</code>	
<code>setFixedHeight(self, int)</code>	
<code>setFixedSize(self, QSize)</code>	<code>setFixedSize(self, int, int)</code>
<code>setFixedWidth(self, int)</code>	
<code>setFocus(self)</code>	<code>setFocus(self, Qt.FocusReason)</code>
<code>setFocusPolicy(self, Qt.FocusPolicy)</code>	
<code>setFocusProxy(self, QWidget)</code>	
<code>setFont(self, QFont)</code>	
<code>setForegroundRole(self, QPalette.ColorRole)</code>	
<code>setGeometry(self, QRect)</code>	<code>setGeometry(self, int, int, int, int)</code>
<code>setGraphicsEffect(self, QGraphicsEffect)</code>	
<code>setHidden(self, bool)</code>	
<code>setInputMethodHints(self, ...)</code>	
<code>setLayout(self, QLayout)</code>	
<code>setLayoutDirection(self, Qt.LayoutDirection)</code>	
<code>setLocale(self, QLocale)</code>	
<code>setMask(self, QBitmap)</code>	<code>setMask(self, QRegion)</code>

Continued on next page

Table 12 – continued from previous page

setMaximumHeight(self, int)	
setMaximumSize(self, int, int)	setMaximumSize(self, QSize)
setMaximumWidth(self, int)	
setMinimumHeight(self, int)	
setMinimumSize(self, int, int)	setMinimumSize(self, QSize)
setMinimumWidth(self, int)	
setMouseTracking(self, bool)	
setObjectName(self, str)	
setPalette(self, QPalette)	
setParent(self, QWidget)	setParent(self, QWidget, Union[Qt.WindowFlags, Qt.WindowType])
setProperty(self, str, Any)	
setShortcutAutoRepeat(self, int, enabled)	
setShortcutEnabled(self, int, enabled)	
setSizeIncrement(self, int, int)	setSizeIncrement(self, QSize)
setSizePolicy(self, QSizePolicy)	setSizePolicy(self, QSizePolicy.Policy, QSizePolicy.Policy)
setStatusTip(self, str)	
setStyle(self, QStyle)	
setStyleSheet(self, str)	
setTabOrder(QWidget, QWidget)	
setTabletTracking(self, bool)	
setToolTip(self, str)	
setToolTipDuration(self, int)	
setUpdatesEnabled(self, bool)	
setVisible(self, bool)	
setWhatsThis(self, str)	
setWindowFilePath(self, str)	
setWindowFlag(self, Qt.WindowType, on)	
setWindowFlags(self, Union[Qt.WindowFlags, ...])	
setWindowIcon(self, QIcon)	
setWindowIconText(self, str)	
setWindowModality(self, Qt.WindowModality)	
setWindowModified(self, bool)	
setWindowOpacity(self, float)	
setWindowRole(self, str)	
setWindowState(self, Union[Qt.WindowStates, ...])	
setWindowTitle(self, str)	
sharedPainter(self)	
show(self)	
showEvent(self, QShowEvent)	
showFullScreen(self)	
showMaximized(self)	
showMinimized(self)	
showNormal(self)	
signalsBlocked(self)	
size(self)	

Continued on next page

Table 12 – continued from previous page

sizeHint(self)	
sizeIncrement(self)	
sizePolicy(self)	
stackUnder(self, QWidget)	
startTimer(self, int, timerType)	
statusTip(self)	
style(self)	
styleSheet(self)	
tabletEvent(self, QTabletEvent)	
testAttribute(self, Qt.WidgetAttribute)	
thread(self)	
timerEvent(self, QTimerEvent)	
toolTip(self)	
toolTipDuration(self)	
tr(self, str, disambiguation, n)	
underMouse(self)	
ungrabGesture(self, Qt.GestureType)	
unsetCursor(self)	
unsetLayoutDirection(self)	
unsetLocale(self)	
update(self)	update(self, QRect) update(self, QRegion) update(self, int, int, int, int)
updateGeometry(self)	
updateMicroFocus(self)	
updatesEnabled(self)	
visibleRegion(self)	
whatsThis(self)	
wheelEvent(self, QWheelEvent)	
width(self)	
widthMM(self)	
winId(self)	
window(self)	
windowFilePath(self)	
windowFlags(self)	
windowHandle(self)	
windowIcon(self)	
windowIconText(self)	
windowModality(self)	
windowOpacity(self)	
windowRole(self)	
windowState(self)	
windowTitle(self)	
windowType(self)	
x(self)	
y(self)	

Attributes

DrawChildren

Continued on next page

Table 13 – continued from previous page

DrawWindowBackground	
IgnoreMask	
PdmDepth	
PdmDevicePixelRatio	
PdmDevicePixelRatioScaled	
PdmDpiX	
PdmDpiY	
PdmHeight	
PdmHeightMM	
PdmNumColors	
PdmPhysicalDpiX	
PdmPhysicalDpiY	
PdmWidth	
PdmWidthMM	
customContextMenuRequested	customContextMenuRequested(self, QPoint) [signal]
destroyed	destroyed(self, QObject = None) [signal]
objectNameChanged	objectNameChanged(self, str) [signal]
staticMetaObject	
windowIconChanged	windowIconChanged(self, QIcon) [signal]
windowIconTextChanged	windowIconTextChanged(self, str) [signal]
windowTitleChanged	windowTitleChanged(self, str) [signal]

Symbols

`__init__()` (SiQt.definitions.SiQtMixin method), 5
`__init__()` (SiQt.definitions.SiqtItem method), 6
`__init__()` (SiQt.matplotlib.NavigationToolbar method), 8
`__init__()` (SiQt.widgets.DebugInfoWidget method), 17
`__init__()` (SiQt.widgets.ProgressBarWidget method), 31
`_resource_path()` (in module SiQt.deployment), 8

C

`calculate_dependencies()` (in module SiQt.dep_resolv), 7
`check_depflags()` (in module SiQt.dep_resolv), 7
`closeEvent()` (SiQt.widgets.DebugInfoWidget method),
31

D

`DebugInfoWidget` (class in SiQt.widgets), 17

M

`menu_generator()` (SiQt.definitions.SiQtMixin method),
6

N

`NavigationToolbar` (class in SiQt.matplotlib), 8

P

`ProgressBarWidget` (class in SiQt.widgets), 31

S

`set_dep_flag_recursive()` (SiQt.definitions.SiQtMixin
method), 6
`show_qt_control_element()` (in module SiQt.dep_resolv),
7
`SiqtItem` (class in SiQt.definitions), 6
`SiQtMixin` (class in SiQt.definitions), 5
`sync_gui()` (in module SiQt.dep_resolv), 7

V

`value` (SiQt.definitions.SiqtItem attribute), 7

W

`with_layout()` (SiQt.definitions.SiqtItem class method), 7