**ProjectManager新增控件**

修订历史记录

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **日期** | **版本** | **作者** | **审核者** | **说明** |
| 2022-11-13 | V1.0 | 王茂房 |  | 初稿 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

**目 录**

目录

[1.1 实现控件 3](#_Toc8676)

[1.2 实现designer插件编辑控件需要的Host 7](#_Toc8726)

[1.3 注册控件对象创建的工厂函数 9](#_Toc7197)

本文以图片控件QImageBox为例说明如何为组态编辑端的designer插件新增控件。本文仅供参考，具体请看实现源码。

designer插件的控件具体实现位于共享库{HmiFuncDesigner.pro所在目录}\libs\shared\widgets目录

## 实现控件

以图片控件QImageBox为例源码为qimagebox.cpp，qimagebox.h。编辑端所有需要属性全部使用宏：

**Q\_PROPERTY(type name**

**(READ getFunction [WRITE setFunction] |**

**MEMBER memberName [(READ getFunction | WRITE setFunction)])**

**[RESET resetFunction]**

**[NOTIFY notifySignal]**

**[REVISION int]**

**[DESIGNABLE bool]**

**[SCRIPTABLE bool]**

**[STORED bool]**

**[USER bool]**

**[CONSTANT]**

**[FINAL])**

封装。

如属性图片属性：

Q\_PROPERTY(QString ImageFile READ getImageFile WRITE setImageFile)

|  |  |
| --- | --- |
| 字符 | 功能描述 |
| ImageFile | 属性名称 |
| getImageFile | 获取属性ImageFile的方法 |
| setImageFile | 设置属性ImageFile的方法 |

**qimagebox.h具体实现源码如下：**

#ifndef QIMAGEBOX\_H

#define QIMAGEBOX\_H

#include <QLabel>

#include <QPaintEvent>

class **QImageBox** : public QLabel

{

Q\_OBJECT

Q\_PROPERTY(QString ImageFile READ getImageFile WRITE setImageFile)

Q\_PROPERTY(bool NoScale READ isNoScale WRITE setNoScale)

Q\_PROPERTY(int BoardWidth READ getBoardWidth WRITE setBoardWidth)

Q\_PROPERTY(QColor BoardColor READ getBoardColor WRITE setBoardColor)

public:

explicit **QImageBox**(QWidget \*parent = nullptr);

QString **getImageFile**();

void **setImageFile**(const QString szName);

bool **isNoScale**();

void **setNoScale**(bool bNoScale);

int **getBoardWidth**();

void **setBoardWidth**(int iWidth);

QColor **getBoardColor**();

void **setBoardColor**(QColor color);

protected:

void ***resizeEvent***(QResizeEvent \*event);

void ***paintEvent***(QPaintEvent \*event);

private:

QString m\_szImageFile;

QImage m\_imageObj;

bool m\_bNoScale;

int m\_iBoardWidth;

QColor m\_boardColorObj;

};

#endif // QIMAGEBOX\_H

**qimagebox.cpp具体实现源码如下：**

#include "qimagebox.h"

#include "../qprojectcore.h"

#include <QFileInfo>

#include <QPainter>

QImageBox::**QImageBox**(QWidget \*parent) : QLabel(parent)

{

m\_szImageFile = "";

m\_bNoScale = false;

m\_iBoardWidth = 1;

m\_boardColorObj = Qt::black;

this->setAlignment(Qt::AlignCenter);

this->update();

}

QString QImageBox::**getImageFile**()

{

return m\_szImageFile;

}

void QImageBox::**setImageFile**(const QString szName)

{

QStringList szListInfo = szName.split("|"); // 图片名称|图片BASE64数据

if(szListInfo.size() == 2) {

QFileInfo info(szListInfo.at(0));

m\_imageObj = PictureResourceManager::base64ToImage(szListInfo.at(1).toLocal8Bit(), info.suffix());

this->update();

}

m\_szImageFile = szName;

}

bool QImageBox::**isNoScale**()

{

return m\_bNoScale;

}

void QImageBox::**setNoScale**(bool bNoScale)

{

m\_bNoScale = bNoScale;

this->update();

}

int QImageBox::**getBoardWidth**()

{

return m\_iBoardWidth;

}

void QImageBox::**setBoardWidth**(int iWidth)

{

m\_iBoardWidth = iWidth;

this->update();

}

QColor QImageBox::**getBoardColor**()

{

return m\_boardColorObj;

}

void QImageBox::**setBoardColor**(QColor color)

{

m\_boardColorObj = color;

this->update();

}

void QImageBox::***resizeEvent***(QResizeEvent \*event)

{

QLabel::resizeEvent(event);

}

void QImageBox::***paintEvent***(QPaintEvent \*event)

{

QPainter painter(this);

painter.setRenderHints(QPainter::HighQualityAntialiasing | QPainter::TextAntialiasing);

painter.setPen(QPen(m\_boardColorObj, m\_iBoardWidth, Qt::SolidLine));

QRect rect(0, 0, this->width()-1, this->height()-1);

painter.drawRect(rect);

if(!m\_imageObj.isNull()) {

if(m\_bNoScale) {

QRect imgRect((this->width()-m\_imageObj.width())/2,

(this->height()-m\_imageObj.height())/2,

m\_imageObj.width(),

m\_imageObj.height());

painter.drawImage(imgRect, m\_imageObj);

} else {

painter.drawImage(rect, m\_imageObj.scaled(this->width()-1, this->height()-1));

}

}

QLabel::paintEvent(event);

}

## 实现designer插件编辑控件需要的Host

具体实现位于共享库{HmiFuncDesigner.pro所在目录}\libs\shared\host目录qimagehost.cpp, qimagehost.h

**qimagehost.h具体实现源码如下：**

#ifndef QIMAGEHOST\_H

#define QIMAGEHOST\_H

#include "qframehost.h"

class **QImageHost** : public QFrameHost

{

Q\_OBJECT

public:

Q\_INVOKABLE **QImageHost**(QAbstractHost \*parent = 0);

static QString **getShowName**();//designer插件工具箱内控件显示的名称

static QString **getShowIcon**();//designer插件工具箱内控件显示的图标

static QString **getShowGroup**();//designer插件工具箱内控件所在的分组名称

protected:

void ***initProperty***() override; //初始化控件属性

protected:

void ***createObject***() override; //创建图片控件对象

};

#endif // QIMAGEHOST\_H

**qimagehost.cpp具体实现源码如下：**

#include "qimagehost.h"

#include "../property/qabstractproperty.h"

#include "../qpropertyfactory.h"

#include "../qcommonstruct.h"

#include "../widgets/qimagebox.h"

QImageHost::**QImageHost**(QAbstractHost \*parent) : QFrameHost(parent)

{

setProperty("need\_frame", false);

setProperty("accept\_drop", false);

setProperty("scaledContents", true);

}

//designer插件工具箱内控件显示的名称

QString QImageHost::**getShowName**()

{

return tr("图片");//tr("Image");

}

//designer插件工具箱内控件显示的图标

QString QImageHost::**getShowGroup**()

{

return tr("显示部件");//tr("Display Widgets");

}

//designer插件工具箱内控件所在的分组名称

QString QImageHost::**getShowIcon**()

{

return ":/images/image.png";

}

//创建图片控件对象

void QImageHost::***createObject***()

{

m\_object = new QImageBox();

m\_object->setObjectName("image");

}

//初始化控件属性

void QImageHost::***initProperty***()

{

QFrameHost::initProperty();

QAbstractProperty\* pObj = QPropertyFactory::create\_property("Image");

if(pObj != NULL) {

pObj->setObjectProperty("name", "ImageFile");

pObj->setAttribute("show\_name", tr("选择图片"));

pObj->setAttribute("filters", "Images (\*.png \*.xpm \*.jpg)");

pObj->setAttribute("group", "Attributes");

//pObj->setAttribute(ATTR\_CAN\_SAME, true);

insertProperty(pObj);

}

pObj = QPropertyFactory::create\_property("Bool");

if(pObj != NULL) {

pObj->setObjectProperty("name", "NoScale");

pObj->setAttribute("show\_name", tr("原尺寸显示"));

pObj->setAttribute("group", "Attributes");

insertProperty(pObj);

}

pObj = QPropertyFactory::create\_property("Number");

if(pObj != NULL) {

pObj->setObjectProperty("name", "BoardWidth");

pObj->setAttribute("show\_name", tr("边框宽度"));

pObj->setAttribute("group", "Attributes");

insertProperty(pObj);

}

pObj = QPropertyFactory::create\_property("Color");

if(pObj != NULL) {

pObj->setObjectProperty("name", "BoardColor");

pObj->setAttribute("show\_name", tr("边框颜色"));

pObj->setAttribute("group", "Attributes");

insertProperty(pObj);

}

setPropertyValue("geometry", QRect(0, 0, 80, 80));

removeProperty("frameShape");

removeProperty("frameShadow");

}

## 注册控件对象创建的工厂函数

具体实现位于共享库{HmiFuncDesigner.pro所在目录}\libs\shared\qbaseinit.cpp

具体实现函数如下：

void **qRegisterBaseHost**()

{

qRegisterHost("containners.frame", QFrameHost);

qRegisterHost("display\_widgets.label", QLabelHost);

qRegisterHost("display\_widgets.image", QImageHost);

qRegisterHost("display\_widgets.indicationlamp", QIndicationLampHost);

qRegisterHost("display\_widgets.qjarshape", QJarShapeHost);

qRegisterHost("display\_widgets.qvaluestick", QValueStickHost);

qRegisterHost("buttons.pushbutton", QPushButtonHost);

qRegisterHost("buttons.switchbutton", QSwitchButtonHost);

qRegisterHost("inputwidgets.lineedit", QLineEditHost);

}