



个人资料

[管理]



一去、二三里



博客等级: 18

博客积分: 78

博客访问: **407,762**

关注人气: **378**

获赠金笔：**12**

赠出金笔：

荣誉徽章:

相关博文

- 文件监控之FileSystemWatcher(C++ 一去、二三里
- Qt之自定义界面（二）添加最小化 一去、二三里
- Qt中如何启动其他应用程序（QProcess） 一去、二三里
- Qt之多语化 一去、二三里
- Qt之系统托盘（QSystemTrayIcon） 一去、二三里
- Qt之获取本机网络信息 一去、二三里
- UAC及其设置 一去、二三里
- Qt之拖放 一去、二三里
- Qt之模型/视图（自定义风格） 一去、二三里
- Qt解析XML文件（QDomDocument） 一去、二三里

更多>>

正文

字体大小: 大 中 小

QwtPlot之绘制统计图 (三) (2014-11-23 09:07:27) [编辑][删除]

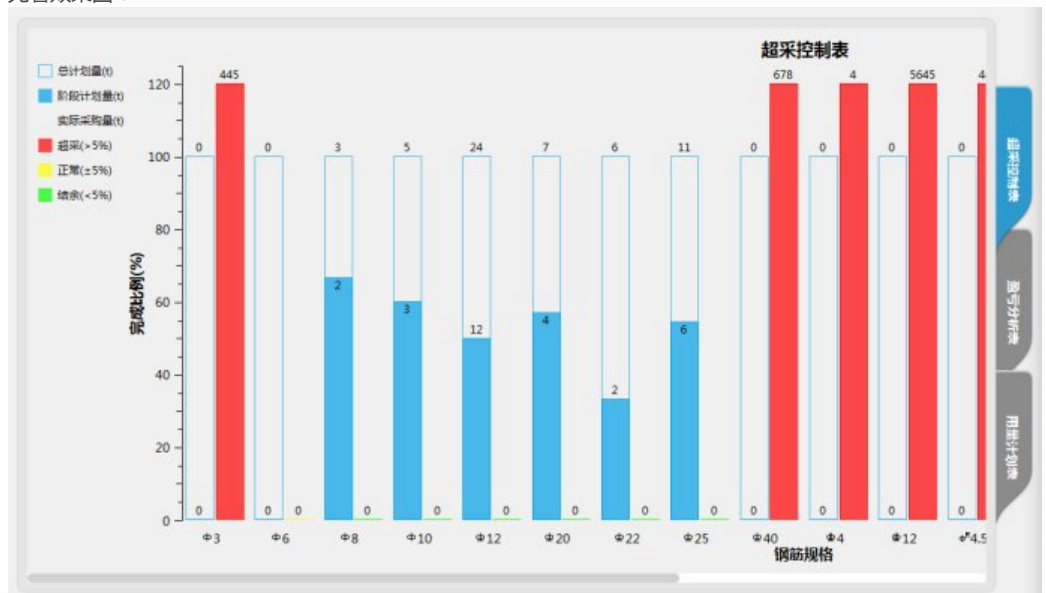
(2014-11-23 09:07:27) [编辑][删除]

[编辑] [删除]

+ 转载 ▼

标签: qt qwt qwtplot qwtplotrenderer 分类: Qt

这几天一直忙于做项目，也没时间写博客，一大早起来先继续上节的内容讲述QWT绘制统计图。先看效果图：



QWT中使用现有的东西一般可以满足比较简单的需求，但绝大多数都需要自己进行比较复杂的处理才可已达到想要的效果！

在这里，讲述一下上图效果的实现方式。

一：X轴需要实现效果：图标+文本。

QwtScaleDraw现有的方式无法满足，所以就自行处理。这里为了方便显示，每个文本均先显示为20个空格，实际显示的“图标+文本”在绘制事件中处理！

第1209篇 • 图腾

评论 |

北美崔哥：美国正称霸世界，春晚

第1207篇 • 日货

某些国家为何拿高铁来戏弄中国？

肖鹰：柴静的意义——她为何值得

第1202篇 • 冤死

北美崔哥：中国式上访，已正式输

蝗虫之日的启示

关于黄圣依奥斯卡之行遇网络攻击



颐和园暮色



早春二月烟雨江南如画



雪国童话里的欢乐时光



斯里兰卡尼甘布匆匆日子



探秘红崖峪口外大探口堡



实拍荷兰星形要塞纳尔登

查看更多>>

谁看过这篇博文	
	3月2日
	3月1日
	3月1日
	2月28日
	2月27日
	2月25日
	2月25日
	2月24日
	2月10日
	2月6日
	2月5日
	2月5日

```
if (m_scaleDraw == BreakevenAnalysePlot)
{
    //20空格
    text = QwtText("      ");
}
return text;
}

//绘制钢筋规格
void DistroScaleDraw::drawLabel(QPainter *painter, double value) const
{
    QFont font = this->label(value).font();
    QRectF rect = this->labelRect(font, value);

    QString text("");
    int index = static_cast<int>(value);
    if (index >= 0 && index < d_labels.size())
        text = d_labels[index];

    QString strBarLevel("");
    QString strDiameter("");
    if (!text.isEmpty())
    {
        QStringList barList = text.split(";");
        if (barList.count() > 1)
        {
            strBarLevel = barList.at(0);
            strDiameter = barList.at(1);
        }
    }

    //获取钢筋直径文本的宽度
    QFontMetrics fm(font);
    int width = fm.width(text);

    //获取左边的坐标
    int x = (rect.width() - width - 13) / 2;
    QPointF point = this->labelPosition(value);
    int left = point.x() - rect.width() / 2 + x;

    painter->drawPixmap(QPointF(left, point.y()), QPixmap(QString(":/Bar/barLevel%1").arg(strBarLevel)));
    painter->drawText(QRect(left + 13, point.y(), rect.width() - x - 13, rect.height() + 5),
        Qt::AlignLeft | Qt::AlignVCenter, strDiameter);
}
```

二：柱状图的绘制

这里为了避免柱状图在绘制的时候文字叠加影像显示效果，会有特殊处理（超过50%时阶段计划量会在下方显示，否则会在上方显示）。

```
GJHOverdraftColumnSymbol::GJHOverdraftColumnSymbol(double amount, double planAmount, double purchaseAmount,
EnAmount enAmount)
: QwtColumnSymbol(),
m_amount(amount),
m_planAmount(planAmount),
m_purchaseAmount(purchaseAmount),
m_enAmount(enAmount),
m_textHeight(18),
m_textSpacing(10),
m_width(30)
{
    //数据四舍五入之后再比较，如果使用真实值比较，则可能存在显示相同，但高度、颜色不一致
    m_amount = fRound(m_amount, 0);
    m_planAmount = fRound(m_planAmount, 0);
    m_purchaseAmount = fRound(m_purchaseAmount, 0);

    m_color = QColor(0, 160, 230);
    if (m_enAmount == PurchaseAmount)
    {
        if (sameValue(m_planAmount, 0))
        {
            if (sameValue(m_purchaseAmount, 0))
            {
                m_color = Qt::yellow;
            }
            else if (m_purchaseAmount > 0)
            {
                m_color = Qt::red;
            }
        }
    }
}
```

```
    }
}
else
{
    if (sameValue(m_planAmount, m_purchaseAmount))
    {
        m_color = Qt::yellow;
    }
    else if (qAbs(m_planAmount - m_purchaseAmount) / m_planAmount <= 0.05)
    {
        m_color = Qt::yellow;
    }
    else
    {
        if (m_planAmount > m_purchaseAmount)
        {
            m_color = Qt::green;
        }
        else
        {
            m_color = Qt::red;
        }
    }
}

m_color.setAlpha(180);
}

//绘制柱形图
void GJH0verdrafftColumnSymbol::draw(QPainter *painter, const QwtColumnRect &rect) const
{
    QRectF rectF = rect.toRect();
    if (!rectF.isValid())
    {
        return;
    }

    Qt::Alignment alignment = Qt::AlignTop | Qt::AlignHCenter;
    int nHeight = rectF.height();

    //设置绘制区域，宽固定为30
    if (rectF.width() != m_width)
    {
        int nLeft = (rectF.width() - m_width) / 2;
        rectF = QRect(rectF.left() + nLeft, rectF.top(), m_width, rectF.height());
    }

    if (m_enAmount == PurchaseAmount)
    {
        int nPurchaseHeight = nHeight;
        if (sameValue(m_purchaseAmount, 0))
        {
            nPurchaseHeight = 0;
        }
        else if (m_purchaseAmount < m_amount * 1.2) //小于120%时，120所对应的重量
        {
            nPurchaseHeight = m_purchaseAmount / (m_amount * 1.2 / nHeight);
        }
        int nSpacing = nHeight - nPurchaseHeight;
        QRectF planRect = QRect(rectF.left(), rectF.top() + nSpacing, rectF.width(), nPurchaseHeight);
        //+5为了显示全数字
        QRectF planTextRect = QRect(rectF.left() - m_textSpacing, rectF.top() + nSpacing - m_textHeight,
            rectF.width() + 2 * m_textSpacing, nPurchaseHeight + m_textHeight);
        painter->setPen(m_color);
        painter->setBrush(m_color);
        painter->drawRect(planRect);
        painter->setPen(Qt::black);
        painter->drawText(planTextRect, alignment, QString("%1").arg(m_purchaseAmount));
    }
    else
    {
        painter->setBrush(Qt::NoBrush);
        painter->setPen(m_color);
        painter->drawRect(rectF);

        int nPlanHeight = 0;
        if (!sameValue(m_amount, 0))
```

```

{
    nPlanHeight = m_planAmount / (m_amount / nHeight);
}
int nSpacing = nHeight - nPlanHeight;
QRect amountTextRect(rectF.left() - m_textSpacing, rectF.top() - m_textHeight, rectF.width() + 2 *
m_textSpacing, nHeight + m_textHeight);
QRect planRect = QRect(rectF.left(), rectF.top() + nSpacing, rectF.width(), nHeight - nSpacing);
QRect planTextRect = QRect(rectF.left() - m_textSpacing, rectF.top() + nSpacing - m_textHeight,
rectF.width() + 2 * m_textSpacing, nPlanHeight + m_textHeight);
painter->setBrush(m_color);
painter->drawRect(planRect);
painter->setPen(Qt::black);
painter->drawText(amountTextRect, alignment, QString("%1").arg(m_amount));

//当超过50%时候文字显示在区域内，否则显示在区域上
if (m_planAmount > m_amount / 2)
{
    planTextRect = planRect;
}
painter->drawText(planTextRect, alignment, QString("%1").arg(m_planAmount));
}
}

```

三：柱状图的显示

这里主要将第二部实现的柱状图显示出来，QwtPlotMultiBarChart顾名思义，即多个柱状图为一组显示。

```

GJHPlotMultiBarChart::GJHPlotMultiBarChart()
: QwtPlotMultiBarChart()
{
}

void GJHPlotMultiBarChart::setList(QList barAmountList)
{
    m_barAmountList = barAmountList;
}

QwtColumnSymbol *GJHPlotMultiBarChart::specialSymbol(int sampleIndex, int valueIndex) const
{
    BarAmount barAmount = m_barAmountList.at(sampleIndex);
    EnAmount enAmount = (valueIndex == 1) ? PurchaseAmount : PlanAmount;
    double dValue = barAmount.amount;
    double dPlanValue = barAmount.planAmount;
    double dPurchaseAmount = barAmount.purchaseAmount;
    GJHOverdraftColumnSymbol *pSymbol = new GJHOverdraftColumnSymbol(dValue, dPlanValue, dPurchaseAmount,
enAmount);
    return pSymbol;
}

```

四：图例的显示

这里比较简单，主要实现自定义的样式（填充画笔+画刷颜色）即可！

```

GJHColumnSymbol::GJHColumnSymbol(QPen pen, QBrush brush)
: QwtColumnSymbol(),
m_pen(pen),
m_brush(brush)
{
}

void GJHColumnSymbol::draw(QPainter *painter, const QwtColumnRect &rect) const
{
    painter->setPen(m_pen);
    painter->setBrush(m_brush);
    painter->drawRect(rect.toRect());
}

```

五：组合显示

完成以上步骤之后，即可已将他们组合起来，实现效果。

```

GJHOverdraftControlChart::GJHOverdraftControlChart(QWidget *parent):
QwtPlot(parent)
{
    setTitle(TRANS_STRING("超采控制表"));
    setAxisTitle(QwtPlot::yLeft, TRANS_STRING("完成比例(%)^"));
    setAxisTitle(QwtPlot::xBottom, TRANS_STRING("钢筋规格"));
}

```

```
QwtPlotCanvas *canvas = new QwtPlotCanvas();
canvas->setFrameStyle(QwtColumnSymbol::NoFrame);
canvas->setBorderRadius(5);
canvas->setCursor(Qt::ArrowCursor);
setCanvas(canvas);

m_barChartItem = new GJHPlotMultiBarChart();
m_barChartItem->setLayoutPolicy(QwtPlotMultiBarChart::AutoAdjustSamples);
m_barChartItem->setSpacing(10);
m_barChartItem->setMargin(0);
m_barChartItem->attach(this);

insertLegend(new QwtLegend());
initColumnSymbol();
}

void GJHOverdraftControlChart::initColumnSymbol()
{
    QList titles;
    titles += TRANS_STRING("总计划量(t)");
    titles += TRANS_STRING("阶段计划量(t)");

    m_barChartItem->setBarTitles(titles);
    m_barChartItem->setLegendIconSize(QSize(15, 15));

    for (int i = 0; i < 2; ++i)
    {
        QColor color = QColor(0, 160, 230);
        color.setAlpha(180);
        QwtColumnSymbol *symbol;
        if (i == 0)
        {
            symbol = new GJHColumnSymbol(QPen(color), QBrush(Qt::NoBrush));
        }
        else
        {
            symbol = new QwtColumnSymbol(QwtColumnSymbol::Box);
            symbol->setFrameStyle(QwtColumnSymbol::NoFrame);
            symbol->setPalette(QPalette(color));
        }
        m_barChartItem->setSymbol(i, symbol);
    }

    QwtPlotCurve *pPlotCurve = new QwtPlotCurve(TRANS_STRING("实际采购量(t)"));
    pPlotCurve->setBrush(Qt::transparent);
    pPlotCurve->setSymbol(NULL);
    pPlotCurve->setLegendIconSize(QSize(15, 15));
    pPlotCurve->attach(this);

    QList strList;
    strList << TRANS_STRING("超采(>5%)") << TRANS_STRING("正常(±5%)") << TRANS_STRING("结余(<5%)");

    QwtPlotMultiBarChart *pMultiBarChart = new QwtPlotMultiBarChart();
    pMultiBarChart->setBarTitles(strList);
    pMultiBarChart->setLegendIconSize(QSize(15, 15));
    pMultiBarChart->attach(this);

    QList overColorList;
    overColorList << Qt::red << Qt::yellow << Qt::green;
    for (int i = 0; i < strList.count(); ++i)
    {
        QColor color = overColorList.at(i);
        color.setAlpha(180);
        QwtColumnSymbol *symbol = new QwtColumnSymbol(QwtColumnSymbol::Box);
        symbol->setFrameStyle(QwtColumnSymbol::NoFrame);
        symbol->setPalette(QPalette(color));
        pMultiBarChart->setSymbol(i, symbol);
    }
}

void GJHOverdraftControlChart::updateData(QList &barAmountList)
{
    m_barFormat.clear();

    QVector< series>
    for (int i = 0; i < barAmountList.count(); ++i)
```

```

{
    BarAmount barAmount = barAmountList.at(i);
    m_barFormat += QString("%1;%2").arg(barAmount.barLevel).arg(barAmount.diameter);

    QVector values;
    for (int j = 0; j < 2; ++j)
    {
        //总量显示到100%处
        if (j == 0)
        {
            values += 100;
        }
        else
        {
            values += 120;
        }
    }
    series += values;
}

m_barChartItem->setList(barAmountList);
m_barChartItem->setSamples(series);

setOrientation();
}

void GJH0verdraftControlChart::setOrientation()
{
    QwtPlot::Axis axis1 = QwtPlot::xBottom;
    QwtPlot::Axis axis2 = QwtPlot::yLeft;
    m_barChartItem->setOrientation(Qt::Vertical);

    //设置后一条数据也可以正常显示
    int nMax = 1;
    if (m_barChartItem->dataSize() > 1)
    {
        nMax = m_barChartItem->dataSize() - 1;
    }
    setAxisScale(axis1, 0, nMax, 1.0);
    DistroScaleDraw *pScaleDraw = new DistroScaleDraw(m_barFormat);
    setAxisScaleDraw(axis1, pScaleDraw);
    setAxisScale(axis2, 0, 125, 20);

    plotLayout()->setCanvasMargin(5);
    plotLayout()->setLegendPosition(QwtPlot::LeftLegend);
    plotLayout()->setSpacing(0);
    setContentsMargins(10, 10, 10, 10);

    replot();
}

```

五：导出

实现功能之后，QWT也提供了导出的功能，导出格式主要包括：*.pdf、*.svg、*.ps、Images(*.bmp、*.ico、*.jpeg、*.png.....)等。

```

void GJH0verdraftControlChart::exportChart()
{
    QString strPath = recentDir();
    if (!strPath.endsWith("\\") || !strPath.endsWith("/"))
    {
        strPath.append("\\");
        strPath = QDir::toNativeSeparators(strPath);
    }
    QwtPlotRenderer renderer;
    renderer.exportTo(this, strPath + TRANS_STRING("超采控制表.pdf"));
}

```

```

bool exportTo( QwtPlot *, const QString &documentName,
               const QSizeF &sizeMM = QSizeF( 300, 200 ), int resolution = 85 );

```

这里会存在文件选择框出现英文的情况，包括图片过大显示不全的情况，因为默认大小为300*200，所以需要处理一下。

如下所示：

```

void GJH0verdraftControlChart::exportChart(int width, int height)
{

```

```
QString strPath = recentDir();
if (!strPath.endsWith("\\") || !strPath.endsWith("/"))
{
    strPath.append("\\");
    strPath = QDir::toNativeSeparators(strPath);
}

QString strFileName = TRANS_STRING("超采控制表");
QString strFilePath = strPath + strFileName;

const QList imageFormats = QImageWriter::supportedImageFormats();

QStringList filter;
filter += TRANS_STRING("PDF文档(*.pdf)");
filter += TRANS_STRING("SVG文档(*.svg)");
filter += TRANS_STRING("Postscript文档(*.ps)");

if (imageFormats.size() > 0)
{
    QString imageFilter = TRANS_STRING("Images图片");
    for (int i = 0; i < imageFormats.size(); i++)
    {
        if (i > 0)
            imageFilter += " ";
        imageFilter += "*. ";
        imageFilter += imageFormats[i];
    }
    imageFilter += ".");

    filter += imageFilter;
}

strFilePath = QFileDialog::getSaveFileName(
    this,
    TRANS_STRING("打印超采控制表"),
    strFilePath,
    filter.join(";"),
    NULL,
    QFileDialog::DontConfirmOverwrite);

if (!strFilePath.isEmpty())
{
    setRecentDir(extractFileDir(strFilePath));

    QwtPlotRenderer renderer;
    renderer.setDiscardFlag(QwtPlotRenderer::DiscardBackground, false);
    renderer.renderDocument(this, strFilePath, QSizeF(width, height));
}
}
```

注：

技术在于交流、沟通，转载请注明出处并保持作品的完整性。

作者：`☆一去♥二三里` 原文：http://blog.sina.com.cn/s/blog_a6fb6cc90102v61z.html。

3

喜欢

分享：         

阅读(579) | 评论(2) | 收藏(0) | 已有4人转载 | 喜欢 | 打印

已投稿到： 排行榜

前一篇：Qt之qSort

后一篇：Qt之QTreeView（一）

评论

重要提示：警惕虚假中奖信息

[发评论]




Rev01

刚好在找绘图库，除了这个还发现一个叫qcustomplot的

2014-11-23 10:11 来自  Rev01 的评论 回复(1)

发评论

一去、二三里：



☐  分享到微博 

☐ 匿名评论

验证码： [请点击后输入验证码](#) [收听验证码](#)

发评论

以上网友发言只代表其个人观点，不代表新浪网的观点或立场。

[< 前一篇](#)
[Qt之qSort](#)

[后一篇 >](#)
[Qt之QTreeView（一）](#)