Subject: GridCtrl + rich texts
Posted by idkfa46 on Thu, 02 Aug 2012 12:29:21 GMT
View Forum Message <> Reply to Message

Hello guys,
is it possibile to active QTF format inside GridCtrl grid?
I'd like to use Superscript['] and Subscript [,] that's working fine on layouts but not inside grid...

[` superscript]
[, subscript]

any suggestion?

thanks,
Matteo

Subject: Re: GridCtrl + rich texts
Posted by Didier on Fri, 03 Aug 2012 16:19:45 GMT
View Forum Message <> Reply to Message

Hi Matteo,

Try to use 'AttrText' (look at 'Display' in U++ help) it will probably work (although I nerver tried)

Subject: Re: GridCtrl + rich texts
Posted by Sender Ghost on Fri, 03 Aug 2012 19:39:47 GMT
View Forum Message <> Reply to Message

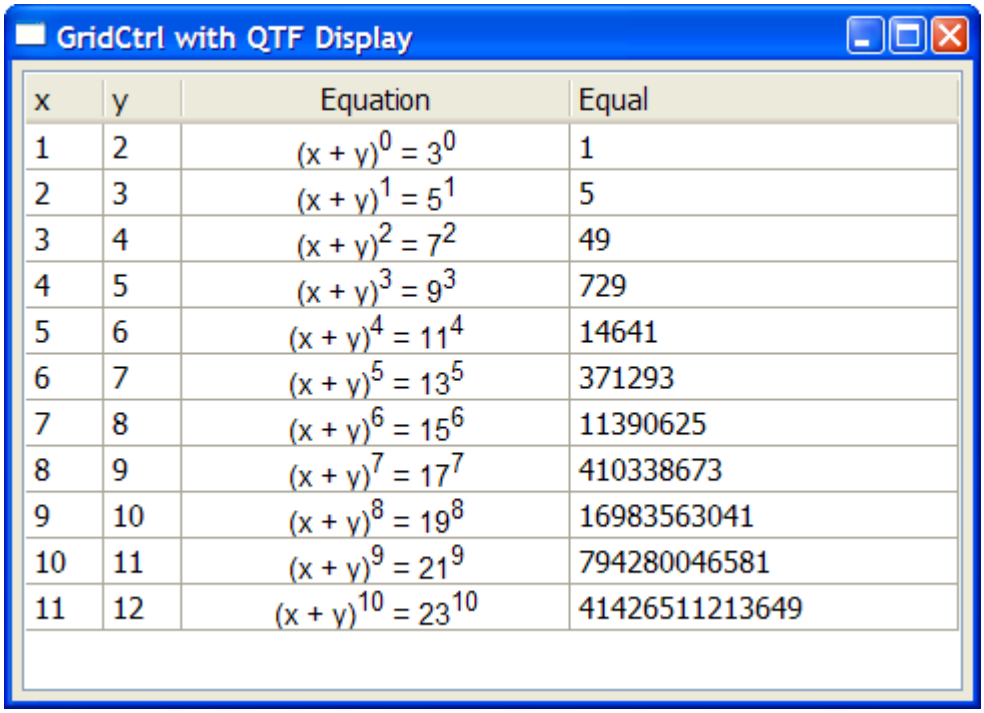
Hello, Matteo.
idkfa46 wrote on Thu, 02 August 2012 14:29Is it possibile to active QTF format inside GridCtrl grid?
Yes, it's possible to create special GridQTFDisplay class (inherited from GridDisplay), which uses QTFDisplay Paint function.

#include <GridCtrl/GridCtrl.h>
#include <RichText/RichText.h>

```
using namespace Upp;

class GridQTFDisplay : public GridDisplay {
public:
 virtual void Paint(Draw& w, int x, int y, int cx, int cy, const Value& val, dword style,
  Color& fg, Color& bg, Font& fnt, bool found, int fs, int fe)
 {
  QTFDisplay().Paint(w, RectC(x, y, cx, cy), val, fg, bg, style);
 }
};

class App : public TopWindow {
public:
 typedef App CLASSNAME;
 App();

 GridCtrl list;
};

App::App()
{
 Title("GridCtrl with QTF Display");
 Sizeable().Zoomable();
 const Size sz(480, 320);
 SetRect(sz); SetMinSize(sz);

 list.Chameleon();
 list.AddColumn("x", 10);
 list.AddColumn("y", 10);
 list.AddColumn("Equation", 50).SetDisplay(Single<GridQTFDisplay>()).HeaderAlignCenter();
 list.AddColumn("Equal");

 for (int i = 0, x = 1, y = 2; i <= 10; ++i, ++x, ++y)
  list.Add(x, y, Format("[= [1 (x + y)][` %d][1  = %d][` %d]]", i, x + y, i),
   pow(double(x + y), i));

 Add(list.VSizePosZ(4, 4).HSizePosZ(4, 4));
}

GUI_APP_MAIN
{
 App app;
 app.Run();
}
```

File Attachments

1) GridQTFDisplay.png, downloaded 1285 times



Subject: Re: GridCtrl + rich texts
Posted by koldo on Sat, 04 Aug 2012 13:38:14 GMT
View Forum Message <> Reply to Message

Nice example.

I did not know it was possible.

Subject: Re: GridCtrl + rich texts
Posted by idkfa46 on Sat, 04 Aug 2012 22:38:46 GMT
View Forum Message <> Reply to Message

Oh... Thanksssssss!!!! GridQTFDisplay class and example are fantastic !!!!
Just one more stupid question....

I was trying to change column alignment (with QTFDisplay) by .alignLeft() but nothing happen ...
It's always at center

regards,
Matteo

Subject: Re: GridCtrl + rich texts

Posted by Sender Ghost on Sat, 04 Aug 2012 23:53:45 GMT
View Forum Message <> Reply to Message

idkfa46 wrote on Sun, 05 August 2012 00:38I was trying to change column alignment (with QTFDisplay) by .AlignLeft() but nothing happen ... It's always at center
Technically, the GridQTFDisplay class forwarding GridDisplay Paint function through QTFDisplay Paint function, which doesn't use information about style variable, therefore GridCtrl align functions not applicable here.

Just use QTF paragraph formatting, e.g. "[< Left]", "[= Center]", "[> Right]", but "<" is by default (and not necessary for this case).
Now, you could understand why it was on the center.

Subject: Re: GridCtrl + rich texts
Posted by forlano on Sun, 05 Aug 2012 10:32:11 GMT
View Forum Message <> Reply to Message

Hello,

I guess it should work for ArrayCtrl too.
I vote to move this example in the reference or examples assembly.

Luigi

Subject: Re: GridCtrl + rich texts
Posted by Sender Ghost on Sun, 05 Aug 2012 13:51:24 GMT
View Forum Message <> Reply to Message

Hello, Luigi.
forlano wrote on Sun, 05 August 2012 12:32I guess it should work for ArrayCtrl too.
Yes, QTFDisplayCls, which inherited from Display, could be used inside many Ctrls, which supports SetDisplay function with Display argument, e.g. ArrayCtrl, ColumnList, DropList, TreeCtrl, etc.
Toggle Spoiler

```
#include <CtrlLib/CtrlLib.h>
#include <RichText/RichText.h>

using namespace Upp;

class App : public TopWindow {
public:
 typedef App CLASSNAME;
 App();

 ArrayCtrl list;
```

```
};

App::App()
{
 Title("ArrayCtrl with QTF Display");
 Sizeable().Zoomable();
 const Size sz(480, 340);
 SetRect(sz); SetMinSize(sz);

 list.AutoHideSb();
 list.SetLineCy(Draw::GetStdFont().GetCy() + 8);
 list.AddColumn("QTF");
 list.AddColumn("QTF Display").SetDisplay(QTFDisplay());

 const String qtf("[1 [* (]x + y[* )]][` %d]");
 for (int i = 0; i <= 10; ++i) {
  const String text(Format(qtf, i));
  list.Add(text, text);
 }

 Add(list.VSizePosZ(4, 4).HSizePosZ(4, 4));
}

GUI_APP_MAIN
{
 App app;
 app.Run();
}
```

forlano wrote on Sun, 05 August 2012 12:32I vote to move this example in the reference or examples assembly.
I agree (with some variation of contents). But this is undocumented functionality and question to developer(s) of RichText U++ package.

Edit: Fixed QTF formatted text.

Subject: Re: GridCtrl + rich texts
Posted by idkfa46 on Sun, 05 Aug 2012 14:23:06 GMT
View Forum Message <> Reply to Message

Thank you! Working good...

Subject: Re: GridCtrl + rich texts
Posted by Didier on Sun, 05 Aug 2012 18:49:10 GMT

I also searched for the same function some time ago.

It sure would have saved me some precious time !!

Subject: Re: GridCtrl + rich texts
Posted by Sender Ghost on Tue, 07 Aug 2012 00:35:43 GMT

I thought about generalized way to access Display through GridDisplay and made following example with templates:

Toggle Spoiler

```
#include <GridCtrl/GridCtrl.h>
#include <RichText/RichText.h>

using namespace Upp;

template <const Display& (*DisplayFn)()>
class DisplayToGrid : public GridDisplay {
protected:
 const Display& display;
public:
 DisplayToGrid() : display(DisplayFn()) {}

 virtual void Paint(Draw& w, int x, int y, int cx, int cy, const Value& val, dword style,
  Color& fg, Color& bg, Font& fnt, bool found, int fs, int fe)
 {
  display.Paint(w, RectC(x, y, cx, cy), val, fg, bg, style);
 }
};

template <const Display& (*DisplayFn)()>
GridDisplay& SingleDisplayToGrid()
{
 return Single<DisplayToGrid<DisplayFn> >();
}

class DisplayGrid : public GridDisplay {
protected:
 const Display *display;
public:
 DisplayGrid() : display(NULL)             {}
```

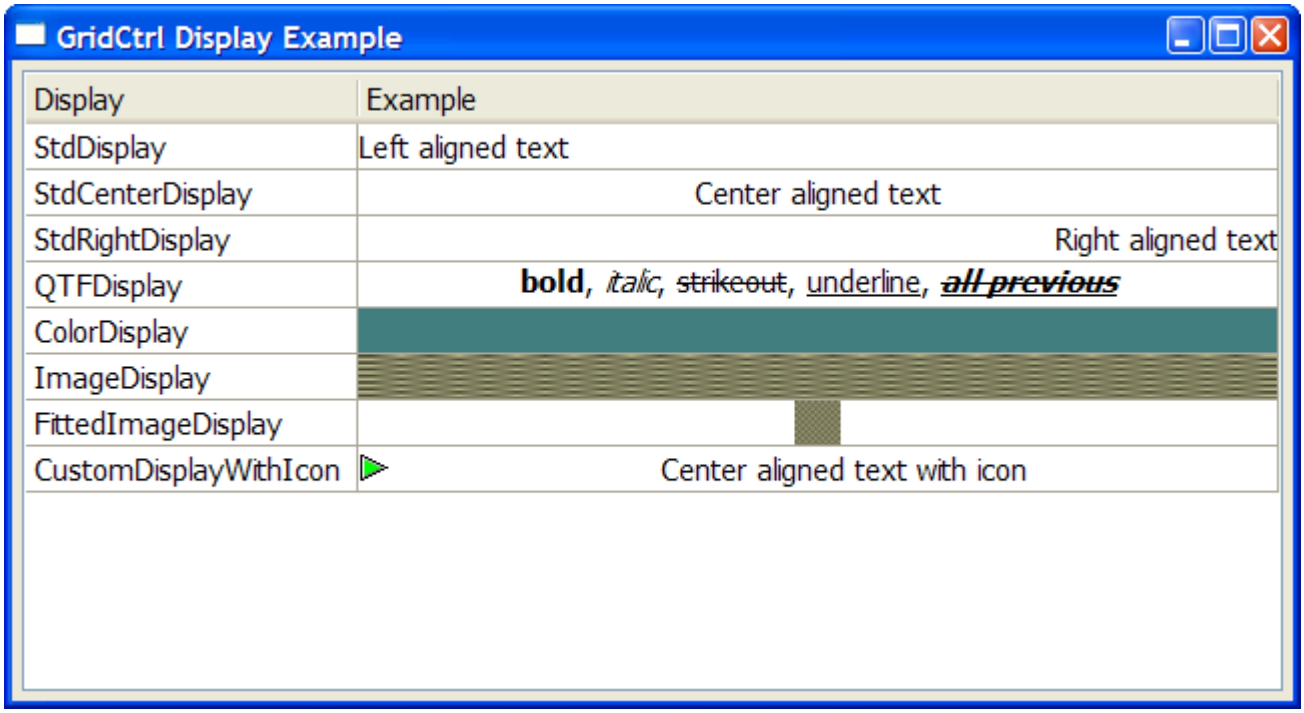
```
 DisplayGrid(const Display& d) : display(&d) {}
 void SetDisplay(const Display& d)          { display = &d; }
 const Display& GetDisplay()               { return *display; }

 virtual void Paint(Draw& w, int x, int y, int cx, int cy, const Value& val, dword style,
  Color& fg, Color& bg, Font& fnt, bool found, int fs, int fe)
 {
  if (display)
   display->Paint(w, RectC(x, y, cx, cy), val, fg, bg, style);
 }
};

class CustomDisplayWithIcon : public DisplayWithIcon {
public:
 virtual void Paint(Draw& w, const Rect& r, const Value& q, Color ink, Color paper, dword style)
const
 {
  w.DrawRect(r, paper);
  DisplayWithIcon::Paint(w, r, q, ink, paper, style);
 }
};

typedef GridDisplay& (*SingleDisplay)();
Tuple3<String, String, SingleDisplay> display[] = {
 "StdDisplay", "Left aligned text", SingleDisplayToGrid<StdDisplay>,
 "StdCenterDisplay", "Center aligned text", SingleDisplayToGrid<StdCenterDisplay>,
 "StdRightDisplay", "Right aligned text", SingleDisplayToGrid<StdRightDisplay>,
 "QTFDisplay", "[G1= [* bold], [/ italic], [- strikeout], [_ underline], [_-*/ all previous]]",
 SingleDisplayToGrid<QTFDisplay>
};

const int displayCount = __countof(display);

class App : public TopWindow {
public:
 typedef App CLASSNAME;
 App();

 GridCtrl list;
};

App::App()
{
 Title("GridCtrl Display Example");
 Sizeable().Zoomable();
 const Size sz(640, 320);
 SetRect(sz); SetMinSize(sz);
```

```
 list.Chameleon();
 list.AddColumn("Display").Fixed(120);
 list.AddColumn("Example");

 for (int i = 0; i < displayCount; ++i) {
  list.Add(display[i].a, display[i].b);
  list.GetCell(i, 1).SetDisplay(display[i].c());
 }

 int count = displayCount;
 list.Add("ColorDisplay", Blend(LtGreen(), Magenta())).GetCell(count++,
1).SetDisplay(SingleDisplayToGrid<ColorDisplay>());
 list.Add("ImageDisplay", CtrlImg::checkers()).GetCell(count++,
1).SetDisplay(SingleDisplayToGrid<ImageDisplay>());
 list.Add("FittedImageDisplay", CtrlImg::checkers()).GetCell(count++,
1).SetDisplay(SingleDisplayToGrid<FittedImageDisplay>());
 // And complex example about how to integrate GridDisplay with Display, while using their custom
functions.
 CustomDisplayWithIcon& dwi = Single<CustomDisplayWithIcon>();
 dwi.SetIcon(CtrlImg::go_forward());
 dwi.SetDisplay(StdCenterDisplay());
 DisplayGrid& dg = Single<DisplayGrid>();
 dg.SetDisplay(dwi);
 list.Add("CustomDisplayWithIcon", "Center aligned text with icon").GetCell(count,
1).SetDisplay(dg);

 Add(list.VSizePosZ(4, 4).HSizePosZ(4, 4));
}

GUI_APP_MAIN
{
 App app;
 app.Run();
}
```

Edit: Replaced DisplayWithIcon with CustomDisplayWithIcon class, which draws background rectangle to fix icon drawing issue (and possibly other Display without background rectangle) on GridCtrl cell. Optimized DisplayToGrid initialization.

File Attachments

1) DisplayGridExample.png, downloaded 1128 times



Subject: Re: GridCtrl + rich texts
Posted by forlano on Thu, 09 Aug 2012 11:48:33 GMT
View Forum Message <> Reply to Message

Hello,

unfortunately I can't now test it myself, but I wonder how this class interact with the AsQtf() method of arrayCtrl.
I hope the qtf code sent in the cells will survive in qtf file to produce a nice output.

Luigi

Subject: Re: GridCtrl + rich texts
Posted by Sender Ghost on Thu, 09 Aug 2012 13:14:18 GMT
View Forum Message <> Reply to Message

forlano wrote on Thu, 09 August 2012 13:48I wonder how this class interact with the AsQtf() method of ArrayCtrl.
If you meant GridDisplay based classes, then they not related to ArrayCtrl, because it uses Display based classes. Moreover, Display and GridDisplay classes used to render input Value. Therefore, they not interact with AsQtf method of ArrayCtrl.

I tested AsQtf method of ArrayCtrl with QTF formatted texts and its (correct) output works inside QTF designer of TheIDE.

Further, I tested QTF formatted texts for examples inside this topic. And I found one mistake:
[1 [* ()x + y[*)]]][` %d]
which I fixed to:
[1 [* ()x + y[*)]][` %d]

Subject: Re: GridCtrl + rich texts
Posted by forlano on Thu, 09 Aug 2012 17:17:40 GMT
View Forum Message <> Reply to Message

Sender Ghost wrote on Thu, 09 August 2012 15:14forlano wrote on Thu, 09 August 2012 13:48I wonder how this class interact with the AsQtf() method of ArrayCtrl.
If you meant GridDisplay based classes, then they not related to ArrayCtrl, because it uses Display based classes. Moreover, Display and GridDisplay classes used to render input Value. Therefore, they not interact with AsQtf method of ArrayCtrl.

I tested AsQtf method of ArrayCtrl with QTF formatted texts and its (correct) output works inside QTF designer of TheIDE.

Further, I tested QTF formatted texts for examples inside this topic. And I found one mistake:
[1 [* ()x + y[*)]]][` %d]
which I fixed to:
[1 [* ()x + y[*)]][` %d]

I meant the class equivalent of GridDisplay but for ArrayCtrl. There we have for free the AsQtf method. So what we will see on the screen will be rendered in a qtf document as well. If I have understood it should work.

Thanks,
Luigi

Subject: Re: GridCtrl + rich texts
Posted by Sender Ghost on Thu, 09 Aug 2012 18:42:01 GMT
View Forum Message <> Reply to Message

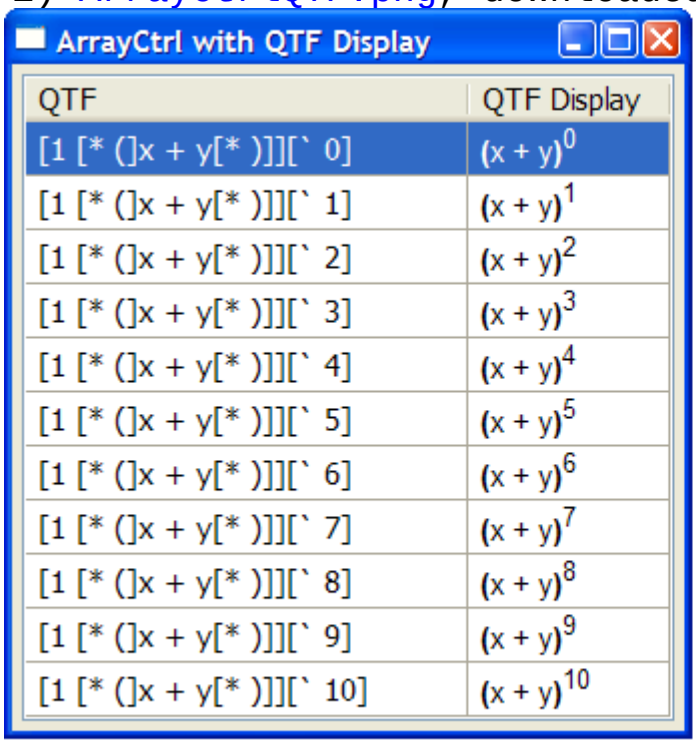
forlano wrote on Thu, 09 August 2012 19:17I meant the class equivalent of GridDisplay but for ArrayCtrl. There we have for free the AsQtf method. So what we will see on the screen will be rendered in a qtf document as well.
Sorry about my wrong guess. Seems like, I thought about reply message instead of this one first.
forlano wrote on Thu, 09 August 2012 19:17If I have understood it should work.
Paraphrasing, yes, the QTFDisplayCls class (also) works inside ArrayCtrl and its correct input Value creates correct output inside AsQtf method.
By input Value I meant QTF formatted text. Without QTFDisplayCls you will see not preprocessed QTF formatted text, but inside QTF designer you will see preprocessed output without reference

to applied (or not applied) QTFDisplayCls class.

I fixed some issues after your message.
Thank you too.

File Attachments

1) ArrayCtrlQTF.png, downloaded 1032 times



QTF	QTF Display
$(x + y)^0$	$(x + y)^0$
$(x + y)^1$	$(x + y)^1$
$(x + y)^2$	$(x + y)^2$
$(x + y)^3$	$(x + y)^3$
$(x + y)^4$	$(x + y)^4$
$(x + y)^5$	$(x + y)^5$
$(x + y)^6$	$(x + y)^6$
$(x + y)^7$	$(x + y)^7$
$(x + y)^8$	$(x + y)^8$
$(x + y)^9$	$(x + y)^9$
$(x + y)^{10}$	$(x + y)^{10}$

Subject: Re: GridCtrl + rich texts
Posted by forlano on Thu, 09 Aug 2012 20:09:18 GMT
View Forum Message <> Reply to Message

Nice job Sender!

I hope to include it in my app very soon.

Thanks,
Luigi

Subject: Re: GridCtrl + rich texts -- Array Control
Posted by deep on Sun, 14 Oct 2018 08:19:49 GMT
View Forum Message <> Reply to Message

Hi,

I am using this very old thread.

I want to have colors in grid and array controls. Want to use QTFDisplay.

ArrayCtrl example

```
#include <CtrlLib/CtrlLib.h>
#include <RichText/RichText.h>

using namespace Upp;

class App : public TopWindow {
public:
 typedef App CLASSNAME;
 App();

 ArrayCtrl list;
};

App::App()
{
 Title("ArrayCtrl with QTF Display");
 Sizeable().Zoomable();
 const Size sz(480, 340);
 SetRect(sz); SetMinSize(sz);

 list.AutoHideSb();
 list.SetLineCy(Draw::GetStdFont().GetCy() + 8);
 list.AddColumn("QTF");
 list.AddColumn("QTF Display").SetDisplay(QTFDisplay());

 const String qtf("[@G [1 [* ]x + y[* )]][` %d]]");
 for (int i = 0; i <= 10; ++i) {
  const String text(Format(qtf, i));
  list.Add(text, text);
  DUMP(text);
 }

 Add(list.VSizePosZ(4, 4).HSizePosZ(4, 4));
}

GUI_APP_MAIN
{
 App app;
 app.Run();
}
```

const String qtf("[@G [1 [* (]x + y[*)]][` %d]]"); "[@G ...]" Gives correct forground color.
"[$G ...] this will not render background Color

File Attachments

1) QTFDisplay1.png, downloaded 763 times

```
list.AutoHideSb();
list.SetLineCy(Draw::GetStdFont().GetCy() + 8);
list.AddColumn("QTF");
list.AddColumn("QTF Display").SetDisplay(QTFDispla

const String qtf("[@G [1 [* (]x + y[* )]][` %d]]")
for (int i = 0; i <= 10; ++i) {
    const String text(Format(qtf, i));
    list.Add(text, text);
    DUMP(text);
}
```

```
list.AutoHideSb();
list.SetLineCy(Draw::GetStdFont().GetCy() + 8
list.AddColumn("QTF");
list.AddColumn("QTF Display").SetDisplay(QTFD

const String qtf("[$G [1 [* (]x + y[* )]][` %
for (int i = 0; i <= 10; ++i) {
```

[$G [
[$G [
[$G [
[$G [
[$G [

Subject: Re: GridCtrl + rich texts
Posted by deep on Sun, 14 Oct 2018 11:34:07 GMT
View Forum Message <> Reply to Message

Sender Ghost wrote on Sat, 04 August 2012 01:09Hello, Matteo.

idkfa46 wrote on Thu, 02 August 2012 14:29Is it possibile to active QTF format inside GridCtrl grid?
Yes, it's possible to create special GridQTFDisplay class (inherited from GridDisplay), which uses QTFDisplay Paint function.

```
#include <GridCtrl/GridCtrl.h>
#include <RichText/RichText.h>

using namespace Upp;

class GridQTFDisplay : public GridDisplay {
public:
 virtual void Paint(Draw& w, int x, int y, int cx, int cy, const Value& val, dword style,
  Color& fg, Color& bg, Font& fnt, bool found, int fs, int fe)
 {
  QTFDisplay().Paint(w, RectC(x, y, cx, cy), val, fg, bg, style);
 }
};

class App : public TopWindow {
public:
 typedef App CLASSNAME;
 App();

 GridCtrl list;
};

App::App()
{
 Title("GridCtrl with QTF Display");
 Sizeable().Zoomable();
 const Size sz(480, 320);
 SetRect(sz); SetMinSize(sz);

 list.Chameleon();
 list.AddColumn("x", 10);
 list.AddColumn("y", 10);
 list.AddColumn("Equation", 50).SetDisplay(Single<GridQTFDisplay>()).HeaderAlignCenter();
 list.AddColumn("Equal");

 for (int i = 0, x = 1, y = 2; i <= 10; ++i, ++x, ++y)
  list.Add(x, y, Format("[= [1 (x + y)][` %d][1  = %d][` %d]]", i, x + y, i),
   pow(double(x + y), i));

 Add(list.VSizePosZ(4, 4).HSizePosZ(4, 4));
}
```

```
GUI_APP_MAIN
{
 App app;
 app.Run();
}
```

In this example also QTF color not working. All other attributes working.
[@G ...] and [$Y ...]

Subject: Re: GridCtrl + rich texts
Posted by deep on Mon, 15 Oct 2018 09:43:03 GMT
View Forum Message <> Reply to Message

Hi
I have done this. Which is working for me for GridCtrl.

It is not elegant but working.

Color in QFT must be given as "[@(nn.nn.nn) " or "[$(nn.nn.nn) "

```
class Gridqtfdisplay : public GridDisplay
{

 public:
  virtual void Paint ( Draw& w, int x, int y, int cx, int cy, const Value& val, dword style,
    Color& fg, Color& bg, Font& fnt, bool found, int fs, int fe )
  {
   String s;
   s = val;
   RegExp r1 ( "(\\[([\\@\\$])\\((\\d+)\\.(\\d+)\\.(\\d+)\\))", RegExp::UNICODE );

   while ( r1.GlobalMatch ( s ) )
   {

    if ( r1[1] == "@" )
     fg = Color ( StrInt ( r1[2] ), StrInt ( r1[3] ), StrInt ( r1[4] ) );

    if ( r1[1] == "$" )
     bg = Color ( StrInt ( r1[2] ), StrInt ( r1[3] ), StrInt ( r1[4] ) );
   }

   QTFDisplay().Paint ( w, RectC ( x, y, cx, cy ), val, fg, bg, style );
```

```
  }
};
```