

Here is the code:

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

using namespace Upp;

// An adhoc protocol
// A cell will merge all consecutive cells right to it that are with Null value.
//
// note, there is still (at least) one missing piece, the text is not clipped to cell rect

class QtfGridDisplay : 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)
    {
        GridCtrl *p=this->parent;
        Color rgb=parent->Get(row,0)==1 ? Color(0,250,128) : bg;

        int realcx=cx;
        if(col!=1 && IsNull(p->Get(row,col)))
            return; // the cell is merged, so no painting for it

        for(int c=col+1; c<p->GetColumnCount();++c)
            if(IsNull(p->Get(row,c)))
                realcx+=p->GetWidth(c+2);
            else
                break;
        w.DrawRect(x,y,realcx,cy, rgb);
        DrawSmartText(w,x,y,realcx,String(val),fnt,fg);
    }

    static QtfGridDisplay& get(){ static QtfGridDisplay d; return d; }
};

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

private:
```

```
GridCtrl grid1;
};
```

```
GridHilite::GridHilite()
{
    grid1.AddHiddenColumn("bg"); // hidden column with value 1 will be painted with a greenish
    background
    grid1.AddColumn("Label");
    grid1.AddColumn("Action");
    grid1.AddColumn("NFC");
    grid1.AddColumn("Description");
}
```

```
String s="\1[ [ *+75 Set Flags]&][ [@5+75 MiguelFollowing`=true]&][ [@5+75
MiguelLeaving`=false]&][ [@5+75 MiguleWandering`=false]]]";
grid1.AddRow(1,"\1[*+75
Wander]",s).SetDisplay(QtGridDisplay::get()).Height(GetSmartTextHeight(s,500));
grid1.AddRow(0,""," \1[*+75 Trigger]",",",", \1[*+75Trigger
nsfwander]").SetDisplay(QtGridDisplay::get());
```

```
// Here are 4 plain text, I will put it in the 2nd column, and make it with hilite background;
String s1="Line 1";
String s2="grid1.VertGrid(false).HSizePosZ(2,2).VSizePosZ(2,2)";
String s3="grid1.AddColumn(\"Description\")";
String s4="OK, that's about it";
```

```
String combined=String().Cat()<<"\1'<<"[ [ *+75 "<<DeQt(s1)<<"]&]"
"[ [@5+75 "<<DeQt(s2)<<"]&]" // use DeQt to escape certain special characters
"[ [@5+75 "<<DeQt(s3)<<"]&]" // use MainMenu->Assist->QtDesigner to get markup strings.
"[ [@5+75 "<<DeQt(s4)<<"]&]";
```

```
grid1.AddRow(1,"",combined).SetDisplay(QtGridDisplay::get())
.Height(GetSmartTextHeight(s,500));
```

```
grid1.VertGrid(false).HSizePosZ(2,2).VSizePosZ(2,2);
Add(grid1);
}
```

```
GUI_APP_MAIN
{
    GridHilite().Run();
}
```