Subject: New graph packages
Posted by koldo on Sun, 15 Jan 2012 08:45:29 GMT
View Forum Message <> Reply to Message

Hello all

There are four new packages in Sandbox to draw Scatter and series graphs in general. They are:
- ScatterDraw: Basic class
- ScatterDraw_demo: Demo of headless drawing for console applications.
- ScatterCtrl: Control
- ScatterCtrl_demo: Different series graph demos.

They combine ScatterCtrl and PlotCtrl/PlotLib features, plus some additional things, like they admit:
- Any kind of data sources through DataSource class
- User defined graph and mark functions

Please test them hard. They are intended to be included in main U++ .

Subject: Re: New graph packages
Posted by BioBytes on Mon, 16 Jan 2012 21:22:38 GMT
View Forum Message <> Reply to Message

Hi Koldo,

Does it mean that bar or stacked bar graphs will be available as in wxFreeChart library?

Kind regards
Biobytes

Subject: Re: New graph packages
Posted by koldo on Mon, 16 Jan 2012 22:24:07 GMT
View Forum Message <> Reply to Message

Hello BioBytes

ScatterDraw package is rather flexible so it can admit many possible new graphs.

If a new graph makes sense for you, just do it or ask for it .

The sooner the better, because now the programming interface is opened in the case a new graph would require to modify it.

And if you know a better simpler way to describe a new graph, just describe it.

Subject: Re: New graph packages
Posted by Didier on Sat, 21 Jan 2012 11:18:50 GMT
View Forum Message <> Reply to Message

Hi Koldo,

this sounds great but unfortunently, you're test project compiles OK but does not execute: the static part gives a X11 error
==4445== Invalid read of size 4
==4445== at 0x69EB1B6: XInternAtom (in /usr/lib/libX11.so.6.3.0)
==4445== by 0x8191FDB: Upp::XAtomRaw(char const*) (X11App.cpp:15)
==4445== by 0x81920EA: Upp::XAtom(char const*) (X11App.cpp:26)
==4445== by 0x8187B92: Upp::Ctrl::IsCompositedGui() (X11Wnd.cpp:662)
==4445== by 0x8167695: Upp::Ctrl::Ctrl() (Ctrl.cpp:560)
==4445== by 0x80A3DD4: Upp::ParentCtrl::ParentCtrl() (Static.cpp:202)
==4445== by 0x80532DE: ScatterDemo::ScatterDemo() (ScatterCtrl_Demo.h:16)
==4445== by 0x80607C4: WithTab10<ScatterDemo>::WithTab10() (ScatterCtrl_Demo.lay:62)
==4445== by 0x806089C: Tab10::Tab10() (ScatterCtrl_Demo.h:153)
==4445== by 0x806001F: s__s69_fn() (tab10.cpp:70)
==4445== by 0x805223C: Upp::Callinit::Callinit(void (*)(), char const*, int) (Defs.h:169)
==4445== by 0x806014D: __static_initialization_and_destruction_0(int, int) (tab10.cpp:69)
==4445== Address 0x4d0 is not stack'd, malloc'd or (recently) free'd

I am under linux with gcc (GCC) 4.4.4 20100630 (Red Hat 4.4.4-10)

I have full Upp up to date and Ide freshly compiled

I'm looking into it

Didier

Subject: Re: New graph packages
Posted by koldo on Sat, 21 Jan 2012 20:29:16 GMT
View Forum Message <> Reply to Message

Hello Didier

I understand. The problem is that I constructed a Ctrl inside an INITBLOCK()...

Now inside the INITBLOCK() there will be only a function call .

Subject: Re: New graph packages
Posted by koldo on Sat, 21 Jan 2012 22:27:22 GMT
View Forum Message <> Reply to Message

Solved.

Subject: Re: New graph packages
Posted by Didier on Sun, 22 Jan 2012 10:21:14 GMT
View Forum Message <> Reply to Message

Hi Koldo,

Thank's !

Looking at the recent posts, I new it was something like that.

Dideir

Subject: Re: New graph packages
Posted by ratah on Mon, 23 Jan 2012 08:39:15 GMT
View Forum Message <> Reply to Message

Hello Koldo,

I'd like to try the new graph packages but I don't have access to the last nightly built u++ (windows version).
The last one in http://code.google.com/p/upp-mirror/downloads/list is 6 days old.

Best regards,

Ratah

Subject: Re: New graph packages
Posted by mirek on Mon, 23 Jan 2012 18:30:44 GMT
View Forum Message <> Reply to Message

Demo looks good.

In Basic Test, there are two strange issues: In non-antialiased mode, bottom x line is missing. And there is quite big color difference between Draw and Painter - but perhaps it is because with Painter, there is some form of Color mixing? (I have not studied the code yet..)

Subject: Re: New graph packages
Posted by mirek on Mon, 23 Jan 2012 18:54:25 GMT
View Forum Message <> Reply to Message

I have fixed some GCC4.5 warnings, except:

[-Wunused-but-set-variable]

perhaps you might want to check the code.

Note: the main change was moving most of member initializations to the constructor body - the reason is that GCC complains a lot if the order of these initializations differs from the order of declarations in class.

Subject: Re: New graph packages
Posted by koldo on Mon, 23 Jan 2012 21:38:20 GMT
View Forum Message <> Reply to Message

Hello Mirek

Thank you for the fixes . I have fixed the "start" warning too. I do not know why the initializations order is so important for GCC... Anyway soon I expect to have an efficient Linux with GCC 4.6. Now I have a good Ubuntu 10.04 but the latest goes through VirtualBox and it works terribly slow.

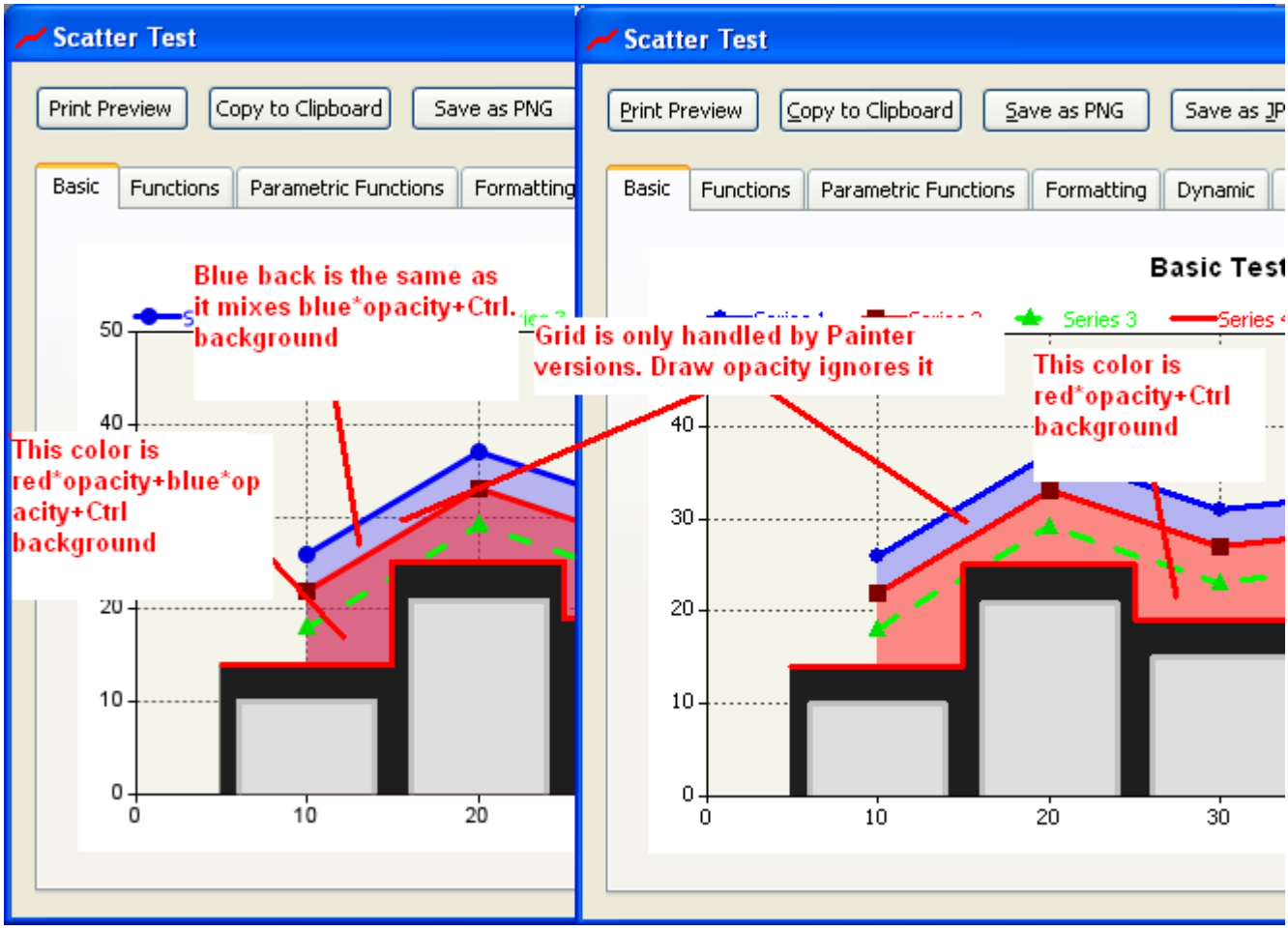
Quote:In Basic Test, there are two strange issues: In non-antialiased mode, bottom x line is missing. And there is quite big color difference between Draw and Painter - but perhaps it is because with Painter, there is some form of Color mixing? (I have not studied the code yet..)

In non-antialiased mode also texts almost disappear. I am not sure if I am doing it right to get the best quality. I move 0.5 in X and Y to get nice thin lines in integer positions.

About color background, the explanation is easy: ScatterDraw supports opacity but... with the ScatterDraw background. Painter manages opacity of all overlapping items, but ScatterDraw Draw version only handles opacity with Ctrl background. Look that the blue background looks the same for all versions.

In addition line opacity was removed as it was a problem to handle the opacity of the line with its own background.

File Attachments
1) dib.PNG, downloaded 2241 times



Subject: Re: New graph packages
Posted by mr_ped on Tue, 24 Jan 2012 07:48:47 GMT
View Forum Message <> Reply to Message

Quote:I do not know why the initializations order is so important for GCC...

IIRC according to C++ standard definition the order of init is either implementation specific or as-defined.
So the order you write inits in the source is not used, and in case some of your init is order depended, this may lead to bugs which are quite difficult to find.

As long as you write the initializers in the as-defined order, you can see such mistake "visually" in case you use some of the variables before it is initialized on the later position.

Subject: Re: New graph packages
Posted by Didier on Fri, 03 Feb 2012 17:22:58 GMT
View Forum Message <> Reply to Message

Hi Koldo,

There is a small bug in ScatterCtrl (also present in Scatter):
When 'SetMajorUnitsNum(xx,yy);' is not called, the ctrl can freeze in very very long loop while truing to adjust y2 unit.

If you add SetMajorUnitsNum(10,10); in the constructor of 'ScatterDraw', this bug is prevented.

NB: I am currently trying to enhance grid recalculation according to zoom, so that the grid does not end up filling the whole paint zone when zooming out real far !
I intend to use the same adjustement technique than the one used in PlotCtrl

---

Subject: Re: New graph packages
Posted by koldo on Fri, 03 Feb 2012 18:53:06 GMT
View Forum Message <> Reply to Message

Hello Didier

Ok. Added.

Quote:NB: I am currently trying to enhance grid recalculation according to zoom, so that the grid does not end up filling the whole paint zone when zooming out real far !
I intend to use the same adjustment technique than the one used in PlotCtrlPerfect, although be careful of avoiding to paint lines that go to points outside viewed window.

Subject: Re: New graph packages
Posted by koldo on Fri, 03 Feb 2012 23:11:19 GMT
View Forum Message <> Reply to Message

Hello Didier

Also do not forget that SetSequentialX() only paints visible zoom.

Subject: Re: New graph packages
Posted by koldo on Sat, 18 Feb 2012 20:15:16 GMT
View Forum Message <> Reply to Message

Hello Didier, Honza and all

Any new comments?.

Please include (Didier) your proposal of new functions, changed function names.

Subject: Re: New graph packages
Posted by Didier on Sun, 19 Feb 2012 12:30:22 GMT
View Forum Message <> Reply to Message

Hi Koldo,

I've just comme back from vacation and I'm gonna go back to my modifications.

Il post them when finished

Subject: Re: New graph packages
Posted by koldo on Sat, 03 Mar 2012 20:56:59 GMT
View Forum Message <> Reply to Message

Hello all

After a month and a half after including new graph packages I will do the next steps:

- Review coding style to be more like standard U++
- Extract DataSource to a new parallel package
- Include the documentation

If you have proposals specially if they affect the external interface, please post them ASAP.

Mirek: In addition if there is anything that you do not like from these packages or you have any proposal please post it ASAP too.

Subject: Re: New graph packages
Posted by Didier on Sun, 04 Mar 2012 12:48:39 GMT
View Forum Message <> Reply to Message

Hi Koldo,

I think DataSource used in ScatterDraw should go template:

The only reason for this is to enable high performance if needed:

```
class DataSource {
public:
 typedef double (DataSource::*Getdatafun)(int id);

 DataSource() : isParam(false) {}
 virtual ~DataSource() {};
 virtual double z(int id)  {return Null;};
 virtual double y(int id)  {return Null;};
```

```
 virtual double x(int id)  {return Null;};
 virtual int GetCount()  {return Null;};
 bool IsParam()    {return isParam;};

 virtual double MinX() {return Min(&DataSource::x);}
 virtual double MinY() {return Min(&DataSource::y);}
 virtual double MinZ() {return Min(&DataSource::z);}

 virtual double MaxX() {return Max(&DataSource::x);}
 virtual double MaxY() {return Max(&DataSource::y);}
 virtual double MaxZ() {return Max(&DataSource::z);}

 virtual double AvgX() {return Avg(&DataSource::x);}
 virtual double AvgY() {return Avg(&DataSource::y);}
 virtual double AvgZ() {return Avg(&DataSource::z);}
```

Class DataSource is a pure virtual class
==> so all method calls need to go by the virtual table ==> poor performance
This is specially true for the x,y,z methods which get called for each point drawn.

If Scatter Draw had was defined the following way:

```
template<class DATASOURCE = DataSource>
class ScatterDraw {
public:
...
```

The following high performance trivial class could be used intead:

```
template<int NBPOINTS>
class DataSource {
private:
 double _x[NBPOINTS];
 double _y[NBPOINTS];
 double _z[NBPOINTS];
public:

DataSource() {}
inline double z(int id)  {return _x[id];}
inline double y(int id)  {return _y[id];}
inline double x(int id)  {return _z[id];}
inline int GetCount()  {return NBPOINTS;}
inline bool IsParam()  {return false;}

inline double MinX() {return ....;}
inline double MinY() {return ....;}
```

```
inline double MinZ() {return ....;}

inline double MaxX() {return ....;}
inline double MaxY() {return ....;}
inline double MaxZ() {return ....;}

inline double AvgX() {return ....;}
inline double AvgY() {return ....;}
inline double AvgZ() {return ....;}
};
```

The backdraw to this is that most of ScatterDraw should go in the header
But after all that is not such a big issue.

NB: I know that most of the performance is due to drawing speed but all enhancements are welcome I think

What do you think ?

Subject: Re: New graph packages
Posted by koldo on Sun, 04 Mar 2012 14:05:02 GMT
View Forum Message <> Reply to Message

Hello Didier

Sorry. I do not understand you well . DataSource class do not keep any data. It is just an interface to other classes.

About going template I did a first try... unsuccessful. If you feel you can get it and it is an advantage, and you can help, it would be great.

Subject: Re: New graph packages
Posted by Didier on Mon, 05 Mar 2012 21:33:40 GMT
View Forum Message <> Reply to Message

Hi Koldo,

As a matter fact it is exactly what I am doing.

I tried to modify ScatterCtrl but understanding everything took me to much time since there aren't any comments.

So I decided to create a new Graph package that uses the maximum of what you did (DataSource class fo example) but fills the areas which I feel need to be gapped.

Things are taking form so I'll publish wehn finished (quickly I hope).

Subject: Re: New graph packages
Posted by koldo on Tue, 06 Mar 2012 19:37:13 GMT
View Forum Message <> Reply to Message

Hello Didier

Quote:I tried to modify ScatterCtrl but understanding everything took me to much time since there aren't any comments.
I understand you. If you have followed my posts about Scatter you will see that I have tried to avoid modifying original code and style as much as possible. However if you see my previous post, you will see this first point:
Quote:After a month and a half after including new graph packages I will do the next steps:

- Review coding style to be more like standard U++
So if main ScatterCtrl/ScatterDraw users agree with interface, I would ask you to wait until I can review the code to do it more legible. Could you wait a little bit? .

Please do not do a new graph package. First was Andrei-Catalin, then I added new things, after that dolk.rce did a new package... many efforts for only one package, and there are many new packages to do...

Subject: Re: New graph packages
Posted by Didier on Tue, 06 Mar 2012 20:17:18 GMT
View Forum Message <> Reply to Message

Hi Koldo,

Quote:First was Andrei-Catalin, then I added new things, after that dolk.rce did a new package... many efforts for only one package, and there are many new packages to do...

I really need a good Graph ctrl and neither PlotCtrl nor ScatterCtrl fit my needs (which are quite simple) .

My goal is not to make yet another Graph Ctrl but to use the maximum of what you created and only rebuilding what needs to.

In fact all the formulas are very complicated (to much) and it's very difficult and tricky to enhance it.

I am focusing on the following points:

Axis management : drawing/steps management/min-max mondification
Calculation simplification
Axis/title/legend position : make all these items modifiable

I dont need to rebuild data painting, the one you build is fine and configurable

So I hope I can comme up with something that can easily be merged with ScatterCtrl

Anyway, even if it finally goes to trash, it's the first time I try to build a Ctrl from scratch, so I will learn many things

---

Subject: Re: New graph packages
Posted by koldo on Tue, 06 Mar 2012 22:40:29 GMT
View Forum Message <> Reply to Message

Hello Didier

Quote:.. and neither PlotCtrl nor ScatterCtrl fit my needs (which are quite simple).
We would be glad to implement what you need. Just describe it or put some samples or screenshots about it.

Quote:So I hope I can come up with something that can easily be merged with ScatterCtrl.
Of course any additions and proposals are very acknowledged. Meanwhile from your point of view could we follow with the proposals as I included them in previous post?:
Quote:- Review coding style to be more like standard U++
- Extract DataSource to a new parallel package
- Include the documentation

There are some new things I would like add to these packages. However I would like to know before if you want anything to change, mainly in the public methods. This way the classes could be used immediatelly.

Subject: Re: New graph packages
Posted by mirek on Wed, 07 Mar 2012 07:02:03 GMT
View Forum Message <> Reply to Message

Didier wrote on Tue, 06 March 2012 15:17
So I hope I can comme up with something that can easily be merged with ScatterCtrl

Anyway, even if it finally goes to trash, it's the first time I try to build a Ctrl from scratch, so I will

learn many things

Well, creating Ctrl from scratch certainly has its educational value.

Long term, I wish there was only one graphing package, which should evolve from current ScatterCtrl. So let us hope your code will be easy to merge

Meanwhile, if you want your code public, please commit it to sandbox (not bazaar).

Mirek

Subject: Re: New graph packages
Posted by koldo on Wed, 07 Mar 2012 08:18:36 GMT
View Forum Message <> Reply to Message

Hello Mirek

Quote:...which should evolve from current ScatterCtrl
Where should evolve ScatterCtrl?. Now it is in a "limbo"...

Subject: Re: New graph packages
Posted by mirek on Wed, 07 Mar 2012 08:31:14 GMT
View Forum Message <> Reply to Message

koldo wrote on Wed, 07 March 2012 03:18Hello Mirek

Quote:...which should evolve from current ScatterCtrl
Where should evolve ScatterCtrl?. Now it is in a "limbo"...

Sense we are adopting it soon to upp.src... I wanted to make one more pass through it before that, but maybe even that is not necessarry. That said, I might need it in my production code, so perhaps will happen anyway. And after that, I it is not dead code, so it will evolve more

---

Subject: Re: New graph packages
Posted by Didier on Wed, 07 Mar 2012 21:24:51 GMT
View Forum Message <> Reply to Message

Quote:Well, creating Ctrl from scratch certainly has its educational value.

Mainly for fun

Quote:Long term, I wish there was only one graphing package, which should evolve from current ScatterCtrl. So let us hope your code will be easy to merge Smile

Several parts of ScatterCtrl can be separated (which I did for needs):

SeriesPlot
MarkPlot
DrawingFunctions
DataSource

I will use them "as is" if possible
So if Koldo can do a clean Separation of these parts from the ScatterCtrl merge could be trivial

Currently I have introduced a GridAxisDraw class that draws the axis and also Grid. The main point is to make grid/axis look extendable and also make grid steps extendable (through class derivation). For ex: maybe I need the steps have certain periodicity or for log a different representation.
This class also contributes to the coordinate conversion ==> this way we could click on the axis, modify ranges/scale ==> and grid/points tied to this axis (and only these points get rescaled)
NB: the coordinate conversion is not owned by this class, another helper class is in charge of this.

Quote:
Meanwhile, if you want your code public, please commit it to sandbox (not bazaar).

When I have something that I find "publishable" I will do it.

Subject: Re: New graph packages
Posted by koldo on Thu, 08 Mar 2012 11:57:06 GMT
View Forum Message <> Reply to Message

Hello Didier

I have begun clarifying (not simplifying) a little bit the mathematics and changing names of some confusing variables (like "l" and "h"). In addition now there will be different left/right and top/bottom margins.

As it is useful for you I will separate in different files, classes:
- SeriesPlot
- MarkPlot
- DrawingFunctions

And DataSource in a different package.

In addition I will try to separate in different classes:
- GridAxisDraw including grid texts
- LegendDraw for graph legends (following PlotCtrl suggestion)

This way Scatter classes will be even more flexible...

Subject: Re: New graph packages
Posted by Didier on Fri, 09 Mar 2012 14:17:52 GMT
View Forum Message <> Reply to Message

Hi Koldo,

Quote:As it is useful for you I will separate in different files, classes:
- SeriesPlot
- MarkPlot
- DrawingFunctionsYou're class separation fit perfectly with my code

Except for one thing:
```
 static Color GetNewColor(int id);
 static String GetNewDash(int id);
 static MarkPlot *GetNewMarkPlot(int id);
```

These methods need to go public.

Looking closer at the SeriesPlot and MarkPlot classes, I think
it could be a good idea to add a virtual void SetScale(int scale) {} method in SeriesPlot and MarkPlot classes.
This would enable children classes to have premultiplied values if needed.

This could be very useful to speed up the of drawing complex marks or line styles since 'SetScale()' would get called only once per drawn series (and not for each point drawn).

Subject: Re: New graph packages
Posted by Didier on Fri, 09 Mar 2012 22:27:25 GMT
View Forum Message <> Reply to Message

Koldo,

I needed to separate another part to make it common: the one that manages the series list.

To do that, I used CRTP (Curiously Recurring Template Pattern) and added a SeriesGroupClass.

I extracted only the common parts, all specif methods like :
```
 SetDataPrimaryY()
 SetSequentialX()
 ...
```

were left in ScatterDraw.

I modified ScatterCtrl svn:4671 (see attached file)
==> all works fine in ScatterCtrl_demo.

With this I have exactly the same interface for:

 data managing
 line style plot
 mark style plot

NB: I also had to do some other minor changes:
Color GetNewColor(int id);
String GetNewDash(int id);
MarkPlot *GetNewMarkPlot(int id); were moved to DrawingFuntions
and Scatter::Refresh() was made public

Do you agree to take these changes in account ?

I think all common parts (at least in version 4671) are axtracted...

File Attachments

1) ScatterDraw_modif_svn4671.tar.gz, downloaded 496 times

Subject: Re: New graph packages
Posted by koldo on Sun, 11 Mar 2012 17:16:06 GMT
View Forum Message <> Reply to Message

Hello Didier

I have tried to understand your changes unsuccessfully. It is difficult for me to follow the changes in the files.

This functions:
Color GetNewColor(int id);
String GetNewDash(int id);
MarkPlot *GetNewMarkPlot(int id);
are already protected because they are only used in ScatterDraw classes and subclasses, so apparently they do not need to be out.

About SetScale(), AFAIK I see all drawing functions get scale only once per drawn series.

It is very interesting to have exactly the same interface for:

 data managing
 line style plot
 mark style plot
Please include a proposal of the declarations of this functions.

Refresh() is now public.

Subject: Re: New graph packages
Posted by Didier on Sun, 11 Mar 2012 18:50:12 GMT
View Forum Message <> Reply to Message

Hello Koldo,

Quote:I have tried to understand your changes unsuccessfully. It is difficult for me to follow the changes in the files.
The changes I made are not functionnal modifications.
The idea is to move the common code from class A to a new template class: B.
The original class (A) then inherits from B: class a : public B {};
==> Class A therefore ahs the same methods as before, except for one (big) difference:
  The return type of the methods : B::x() can only be of type B and not A
  Usually in Upp method calls are chained: we write : x0().x1().x2()

So if I do class a : public B {}; ==> the method chaining is then broken

To prevent this, we can use CRTP in order to have correct return types
CRTP

Better than long text here is an example:
---------- Case 1 : the original code
```
class OriginalClass {
 public:
 OriginalClass() {}
 OriginalClass& fctA() { .....; return *this; }
 OriginalClass& fctD() { .....; return *this; }
};
```

we can write :OriginalClass inst;
isnt.fctA().fctB(); but all is contained in 1 class

---------- Case 2 : naive separation (does not work)
```
class BaseClass {
 public:
 BaseClass() {}
```

```
 BaseClass& fctA() { .....; return *this; }
};
class DerivedClass : public  BaseClass {
 public:
 DerivedClass() {}
 DerivedClass& fctD() { return *this; }
};
```

==> we ##CAN NOT## write :DerivedClass inst;
isnt.fctA().fctB();   ==> compilation error

```
---------- Case 3 : separation using CRTP
template<class DERIVED>
class BaseClass {
 public:
 BaseClass() {}
 DERIVED& fctA() { return *static_cast<DERIVED*>(this); }
};
class DerivedClass : public  BaseClass<DerivedClass> {
 public:
 DerivedClass() {}
 DerivedClass& fctD() { return *this; }
};
```

==> We CAN write :DerivedClass inst;
isnt.fctA().fctB(); ==> ScatterCtrl still works
and separation has been made   ==> I can reuse it !!!

So please reconsider   these changes since I only MOVED code and NOT CHANGED code.

Subject: Re: New graph packages
Posted by koldo on Sun, 11 Mar 2012 20:17:49 GMT
View Forum Message <> Reply to Message

Hello Didier

Sorry, I thought CRTP was a kind of joke instead of a "curiously recurring template pattern" .

A drawback of increasing the degree of class templating is that classes are getting more complex without adding functionality. I think we do not have to forget that.

I will try it again...

Subject: Re: New graph packages
Posted by Didier on Sun, 11 Mar 2012 21:22:43 GMT
View Forum Message <> Reply to Message

Hello Koldo,

Quote:A drawback of increasing the degree of class templating is that classes are getting more complex without adding functionality. I think we do not have to forget that. Yes I know, but without this, I can't take advantage of you're code, and interface will probably diverge

But if it's really a problem I'll just copy it for now.

Subject: Re: New graph packages
Posted by dolik.rce on Sun, 11 Mar 2012 21:42:15 GMT
View Forum Message <> Reply to Message

Hi Koldo
koldo wrote on Sun, 11 March 2012 18:16This functions:
Color GetNewColor(int id);
String GetNewDash(int id);
MarkPlot *GetNewMarkPlot(int id);
are already protected because they are only used in ScatterDraw classes and subclasses, so apparently they do not need to be out.
Simple usecase where having these methods publicly available would be very helpful:

If there is multiple series, I would call GetNewColor(my_series_id) to assign color after each ordering change (where my_series_id is some unique identification of the series that doesn't depend on it's position in graph). The goal is to keep the colors the same even when some series are removed, added or sorted differently. It helps to keep the graphs easily "readable"

The same holds for GetNewDash and for GetNew_MarkPlot... Please consider making them publicly available, all it will cost you is three declarations in header file

Best regards,
Honza

Subject: Re: New graph packages
Posted by koldo on Mon, 12 Mar 2012 08:10:23 GMT
View Forum Message <> Reply to Message

Hello Didier

Yes I know, but without this, I can't take advantage of you're code, and interface will probably divergeIt is difficult to optimize a class to match to something only you knows.

It would be better if you would make public your needs to see how to match code with them.

Subject: Re: New graph packages
Posted by koldo on Mon, 12 Mar 2012 08:12:15 GMT
View Forum Message <> Reply to Message

dolik.rce wrote on Sun, 11 March 2012 22:42Hi Koldo
koldo wrote on Sun, 11 March 2012 18:16This functions:
Color GetNewColor(int id);
String GetNewDash(int id);
MarkPlot *GetNewMarkPlot(int id);
are already protected because they are only used in ScatterDraw classes and subclasses, so apparently they do not need to be out.
Simple usecase where having these methods publicly available would be very helpful:

If there is multiple series, I would call GetNewColor(my_series_id) to assign color after each ordering change (where my_series_id is some unique identification of the series that doesn't depend on it's position in graph). The goal is to keep the colors the same even when some series are removed, added or sorted differently. It helps to keep the graphs easily "readable"

The same holds for GetNewDash and for GetNew_MarkPlot... Please consider making them publicly available, all it will cost you is three declarations in header file

Best regards,
Honza

Hello Honza

Is perhaps the solution to add some kind of graph KEY or ID as in GridCtrl?

Subject: Re: New graph packages
Posted by dolik.rce on Mon, 12 Mar 2012 09:45:37 GMT
View Forum Message <> Reply to Message

koldo wrote on Mon, 12 March 2012 09:12dolik.rce wrote on Sun, 11 March 2012 22:42Hi Koldo
koldo wrote on Sun, 11 March 2012 18:16This functions:
Color GetNewColor(int id);
String GetNewDash(int id);
MarkPlot *GetNewMarkPlot(int id);
are already protected because they are only used in ScatterDraw classes and subclasses, so apparently they do not need to be out.
Simple usecase where having these methods publicly available would be very helpful:

If there is multiple series, I would call GetNewColor(my_series_id) to assign color after each

ordering change (where my_series_id is some unique identification of the series that doesn't depend on it's position in graph). The goal is to keep the colors the same even when some series are removed, added or sorted differently. It helps to keep the graphs easily "readable"

The same holds for GetNewDash and for GetNew_MarkPlot... Please consider making them publicly available, all it will cost you is three declarations in header file

Best regards,
Honza

Hello Honza

Is perhaps the solution to add some kind of graph KEY or ID as in GridCtrl?

Yes, that would solve the situation from the other side It seems little bit more difficult to implement this way, but possibly more versatile.

Honza

Subject: Re: New graph packages
Posted by koldo on Mon, 12 Mar 2012 11:16:58 GMT
View Forum Message <> Reply to Message

Quote:Yes, that would solve the situation from the other side Smile It seems little bit more difficult to implement this way, but possibly more versatile.Hello Honza

What interface do you propose?. I mean, how would you change AddSeries() functions and other?

Subject: Re: New graph packages
Posted by Didier on Mon, 26 Mar 2012 21:53:13 GMT
View Forum Message <> Reply to Message

Hi all,

My graph package is taking form so I'm going to publish a first version.

The design guideline I followed was to make it the most configurable possible, in order to be able to do a lot of things (at least for 2D graphs)

GraphDraw is based on the followinf principles:

A coordinate manager is in charge of each graph<->screen conversion => a unique instance is used for plotting, grid/axis management
The plot can be decorated with "GraphElements" (Left, right, top, bottom, over the whole graph, ...) just like Ctrls.

They are stacked on each side of the plot. You can create whatever class you need for you're graph (like drawing exclusion zones on a graph, ...)
One nice feature of GraphElements is that they have a callback that is called when mouse clicks on it ==> In the example try clicking on the axis
The Grid and Axis drawing are grouped in one class that can be overloaded (in fact it is a GraphElement)
The tick management is dedicated to a specific class: GridStepManager : it calculates the positions where ticks/grid needs to be drawn. The intent of this class is to allow custom step calculation (log, ...)

The main classes are:

CoordinateConverter : it manages the coordinate conversion and can be overloaded for custom needs
GraphLementFrame :
GridAxisDraw
GridStepManager
GraphDraw
EmptyGraphDraw |
StdGraphDraw | helper classes intended to ease the use

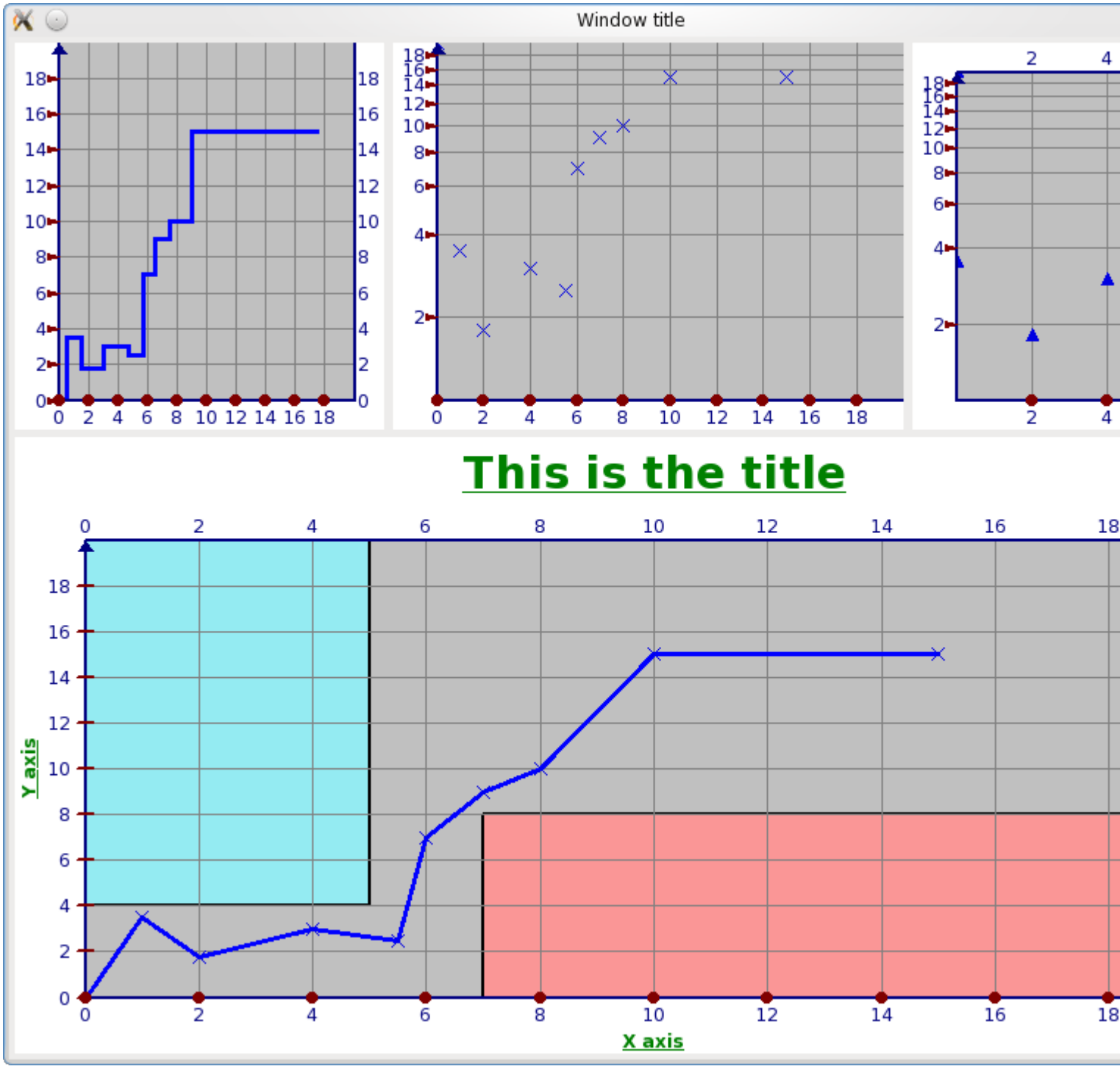
The code is pretty dirty and not cleaned at all (lots and lots of templates, I will kick them out once architecture and optimisations are finished)
The easyest way to use it to use the 'StdGraphCtrl' as in the example

Many functions are missing (zoom, scroll, ...) but the code architecture is pretty much finished although a lot of work is still to be done .

The data management is 100% compatible with ScatterDraw since it uses the same interface.
You can put as many axis as you wan't

File Attachments

1) GraphCtrlExample.png, downloaded 1984 times



Subject: Re: New graph packages
Posted by Didier on Mon, 26 Mar 2012 22:00:08 GMT
View Forum Message <> Reply to Message

And here is the code

I will put it in svn tomorrow

File Attachments
1) GraphDraw.tar.gz, downloaded 499 times

Subject: Re: New graph packages
Posted by Didier on Thu, 29 Mar 2012 20:30:04 GMT
View Forum Message <> Reply to Message

Hy,

I just tried to commit under svn and wasn't allowed to do so.
I thought anyone could commit to sandbox ??

Is the username important ???

Subject: Re: New graph packages
Posted by mirek on Sat, 31 Mar 2012 08:51:33 GMT
View Forum Message <> Reply to Message

Didier wrote on Thu, 29 March 2012 16:30Hy,

I just tried to commit under svn and wasn't allowed to do so.
I thought anyone could commit to sandbox ??

Is the username important ???

Yes. Sending your credentials via PM.

Mirek

Subject: Re: New graph packages
Posted by Didier on Mon, 02 Apr 2012 21:36:38 GMT
View Forum Message <> Reply to Message

Hy everyone,

Here comes another version with scrolling / and zooming

Since I still can't seem to be able to access svn (even with login/password) I publish it in the attached file.

The functions added are the following:
* ON AXIS:
  -scroll : mouse wheel OR mouse left + move
  -zoom : CTRL + mouse wheel
  NB: when modifying an axis only the data linked to this axis is impacted
    ==> you can move/zoom only some plots while the others don't change

* ON PLOT :
  - scroll : mouse left + move
  - zoom : CTRL + mouse wheel
  - select zoom : CTRL + mouse left
  NB: all the data plot is modified

* DRAWING :
 - when zooming/scrolling a REFRESH_FAST has been introduced in order to have smooth quick refresh when scrolling or zooming.
   Once finished, a REFRESH_COMPLETE is done
   ==> this gives great results on a BIG DATASET

I also changed the example a bit, to show the possibilty of having N different axis (like Y & Y2) and their possibilty to scroll/zoom independently

NB: I know the LOG management is buggy, but it's not critical for the moment and I'll correct it later

File Attachments
1) GraphDraw_2.tar.gz, downloaded 508 times

Subject: Re: New graph packages
Posted by Didier on Mon, 16 Apr 2012 19:38:23 GMT
View Forum Message <> Reply to Message

Hi all,

I finaly managed to commit to svn.

The new version has some more enhancements:

recovered function support (as in scatter ctrl)
cursor image management
fast draw corrections and enhancements
vertical/horizontal ZOOM/SCROLL can now select whole graph or only the selected axis

The ZOOM/SCROLL behaviour on key has been modified:
[CTRL] => ZOOM
[SHIFT] => apply function to selected axis only
[NO KEY]=> SCROLL

so it gives the following:
On plot:
[wheel]   : +-20% SCROLL
[MOUSE LEFT] : SCROLL
[CTRL + wheel] : +-20% ZOOM
[CTRL + MOUSE LEFT] : select area for ZOOM

On axis:
[wheel]   : +-20% SCROLL on whole plot
[MOUSE LEFT] : SCROLL   on whole plot
[CTRL + wheel]: +-20% ZOOM   on whole plot

[SHIFT + wheel]   : +-20% SCROLL on selected axis only
[SHIFT + MOUSE LEFT] : SCROLL   on selected axis only
[SHIFT + CTRL + wheel]: +-20% ZOOM   on selected axis only

The cursor image is also adapted to reflect the function goiing top be done.

I also added an 'hourglass' cursor image when drawing is in progress
==> when drawing gets long on big datasets, the user knows he can't do anything else

The next work I will do is :

add legend management
do some documentation

Subject: Re: New graph packages
Posted by Didier on Thu, 19 Apr 2012 12:42:23 GMT
View Forum Message <> Reply to Message

Hi Koldo,

While retreiving the DrawLegend code of ScatterCtrl to inject it into GraphDraw, I found a bug:

At ScatterDraw.cpp L192

if (series[i].opacity > 0 && series[j].seriesPlot)
sould be replace by:
if (series[i].opacity > 0 && series[i].seriesPlot)
j ==> i

Subject: Re: New graph packages
Posted by koldo on Fri, 20 Apr 2012 06:00:16 GMT
View Forum Message <> Reply to Message

Done.

Subject: Re: New graph packages
Posted by Didier on Tue, 01 May 2012 20:50:54 GMT
View Forum Message <> Reply to Message

My GraphDraw / GraphCtrl packages are progressing:

Here are the latest evolutions (all in svn sandbox):

Corrected all the series drawing in order to work with all ScatterDraw SeriesPlots
Corrected LOG (works with zoom/scroll)
Corrected SCALING
Added the copy/save image functions
Added a ScatterCtrl replacement (just to profit the ScatterCtrl_demo). It is not a full replacement (all mouse config is not replaced, but all the rest works)

RECALL: The intent of this package is to give the maximum of flexibility in graph, so it inherits a lot from ScatterCtrl while adding different code strucure and some enhancements:

add different drawing schemes easily : axis/grids/legend/... while keeping UI reactions factored
Zooming/Scrolling is a factored functionality
Code Structure is a "ctrl like" => the graph is made of a plot zone + other "ctrls" adding axis/grids/legends/labels/spaces/.... whatever you invent
You can put as many axis as you wan't (just as long as it's useful)

Next modifications in the list:

Add zoom/scroll SLAVE mechanism ==> so you can synchronize several graphs together or make a global view that manages the local zoom/pan
Make a grid dedicated to LOG
Add axis text formating (printing dates, ...)
Documentation (not started yet)

Subject: Re: New graph packages
Posted by Didier on Sun, 27 May 2012 10:24:40 GMT
View Forum Message <> Reply to Message

Hi all,

I just published some modifications under SVN:

Corrected compilation under WINDOWS (sorry it didn't work before)
Added Graph image buffering to speed up display when other windows overlap (or when Graph is in tabs)

Subject: Re: New graph packages
Posted by Didier on Tue, 03 Jul 2012 16:12:25 GMT
View Forum Message <> Reply to Message

Here are the latest changes I made in my Graph packages:

Some documentation borning (an UML graph has been added using PlantUML)
MaxSize of a graph can be limited so that the user cannot zoom out to much
Axis step text formating can be changed using a callback :
Added support for Date and Time axis : the value of the axis must be Date().Get() or Time().Get() ==> the grid steps are managed in order to deal with years/months/week(partial)/days/hours/minutes/seconds

GridAxisDraw::setAxisTextFormat(TypeFormatTextCbk v)
GridAxisDraw::setAxisDateFormat()
GridAxisDraw::setAxisTimeFormat()

CoordinateConverter::setGraphMaxRangeLimit(TypeGraphCoord v)
CoordinateConverter::setGraphMinRangeLimit(TypeGraphCoord v)
CoordinateConverter::resetGraphRangeLimits()

The grid step management for log10() axis is coming next

Subject: Re: New graph packages
Posted by 281264 on Mon, 16 Jul 2012 19:35:41 GMT
View Forum Message <> Reply to Message

Hi,

I would like to use this graphs/plots packages but it seems to be more than one package; for instance, in bazaar you can find PlotCtrl/PlotLib but there are others.

Any suggestion about which of them to use? Apart from PlotCtrl/PlotLib that you can find in Bazaar, how can you get the other packages?.

Thank you,

Javier

Subject: Re: New graph packages
Posted by dolik.rce on Mon, 16 Jul 2012 20:02:13 GMT
View Forum Message <> Reply to Message

281264 wrote on Mon, 16 July 2012 21:35Hi,

I would like to use this graphs/plots packages but it seems to be more than one package; for instance, in bazaar you can find PlotCtrl/PlotLib but there are others.

Any suggestion about which of them to use? Apart from PlotCtrl/PlotLib that you can find in Bazaar, how can you get the other packages?.

Thank you,

Javier
Hi Javier,

PlotLib/PlotCtrl are slowly getting obsolete now. Koldo adapted the main features from them into the ScatterCtrl, so you should probably use that (unless you have some very specific needs).

Both Koldos and Didiers packages can be found in sandbox under the names ScatterCtrl and GraphDraw, respectively.

Best regards,
Honza

Subject: Re: New graph packages
Posted by 281264 on Mon, 16 Jul 2012 20:29:12 GMT
View Forum Message <> Reply to Message

Honza,

Many thanks; I've found that ScatterCtrl is already a package and that there is a demo in Examples. Two questions:

Which of them to use?
Is there a simple way to get access to an application in Sandbox?

Cheers,

Javier.

Subject: Re: New graph packages
Posted by Didier on Mon, 16 Jul 2012 21:56:38 GMT
View Forum Message <> Reply to Message

Hi All,

As promised I added LOG grid steps management but I also did the following:

kicked some template parameters (for GridStepManager() and GrodStepIterator() classes)
Grid text can now be set using QTF which gives the ability to make multilines, bold, ...in the grid lines
Enhanced Grid text drawing for Time

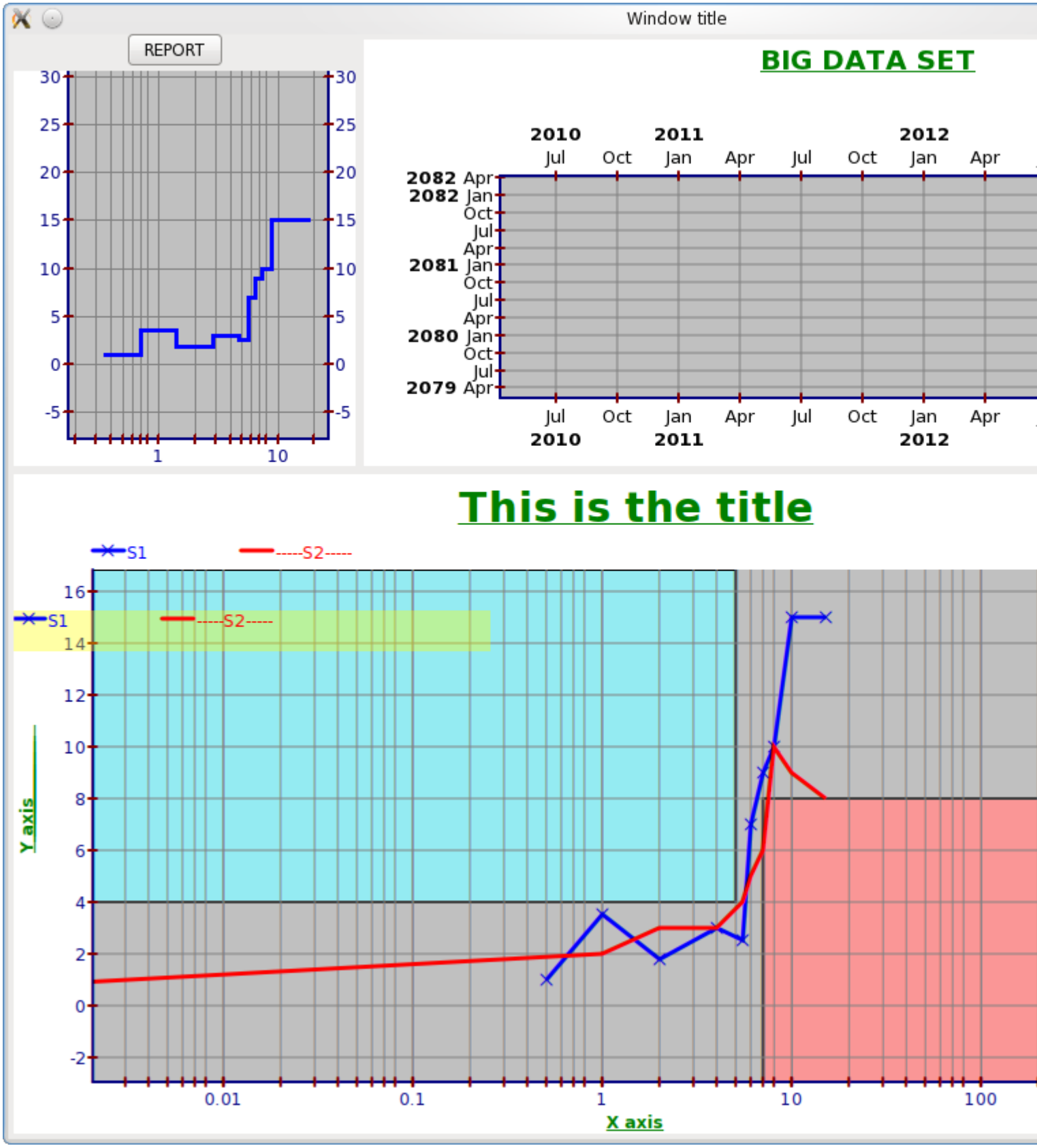
Note: All these features are build to work with zoom and scrolling, so the grid and text is adapted to the range

Some bugs are still creeping around since all this is work in progress but overall working pretty good.

I Also need to post a Demo Package which I will try to do soon enough.

Here is an image showing the result with LOG10 scales and Time data

File Attachments
1) GraphCtrl.png, downloaded 1664 times


Window title
REPORT
BIG DATA SET
2010
2011
2012
Jul
Oct
Jan
Apr
2082
2081
2080
2079
This is the title
S1
-----S2-----
Y axis
X axis
0.01
0.1
1
10
100

Subject: Re: New graph packages
Posted by Didier on Fri, 20 Jul 2012 14:23:52 GMT
View Forum Message <> Reply to Message

Hi,

I just uploaded some corrections:

Log10 management corrections
GraphCtrl/ScatterCtrl.h corrections

Now if you want to see GraphCtrl in action all you have to do is

open the Examples/ScatterCtrl_demo package
add GraphCtrl package to it
in ScatterCtrl_demo.h replace #include "ScatterCtrl/ScatterCtrl.h" by #include "GraphCtrl/ScatterCtrl.h"

Not all is equivalent, but everything compiles and you can see GraphCtrl in action.

Subject: Re: New graph packages
Posted by ratah on Mon, 24 Sep 2012 09:43:34 GMT
View Forum Message <> Reply to Message

Hello everybody,

I try to use ScatterCtrl.
I have a data serie in the range [-10, 10].
How to set min range of my y axis?

SetRange(10, -10, 10)? or what?
I do not want FitXYData for this case.

Thank you

Subject: Re: New graph packages
Posted by Didier on Mon, 24 Sep 2012 17:29:41 GMT
View Forum Message <> Reply to Message

Hi ratah,

If you are Using ScatterCtrl/ScatterCtrl or GraphCtrl/ScatterCtrl all you have to do is:

scatter.SetXYMin(xmin, ymin);
scatter.SetRange(xRange,yRange); with xRange=xMax-xMin and yRange=yMax-yMin

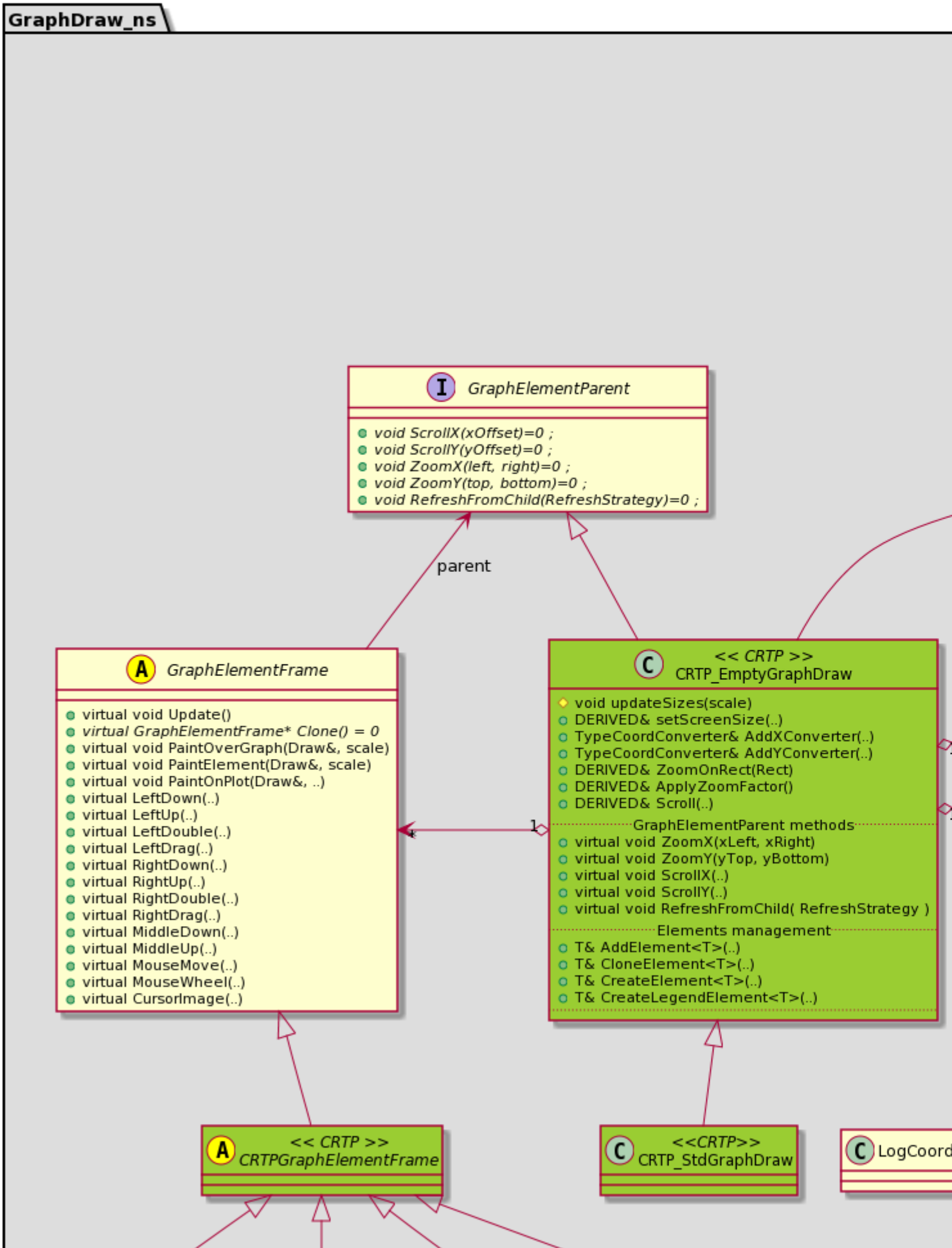
DO IT in this ORDER !!!if you are using ScatterCtrl/ScatterCtrl otherwise i doesn't work(if my memory is wright)

Or, if you are using GraphCtrl classes you can also do the following:
scatter.GetYCoordConverter().updateGraphSize(yMin, yMax);

There also is a class hierarchy doc in the GraphDraw package

File Attachments

1) GraphDraw.png, downloaded 1625 times


GraphDraw_ns
GraphElementParent
void ScrollX(xOffset)=0 ;
void ScrollY(yOffset)=0 ;
void ZoomX(left, right)=0 ;
void ZoomY(top, bottom)=0 ;
void RefreshFromChild(RefreshStrategy)=0 ;
parent
GraphElementFrame
virtual void Update()
virtual GraphElementFrame* Clone() = 0
virtual void PaintOverGraph(Draw&, scale)
virtual void PaintElement(Draw&, scale)
virtual void PaintOnPlot(Draw&, ..)
virtual LeftDown(..)
virtual LeftUp(..)
virtual LeftDouble(..)
virtual LeftDrag(..)
virtual RightDown(..)
virtual RightUp(..)
virtual RightDouble(..)
virtual RightDrag(..)
virtual MiddleDown(..)
virtual MiddleUp(..)
virtual MouseMove(..)
virtual MouseWheel(..)
virtual CursorImage(..)
<< CRTP >>
CRTP_EmptyGraphDraw
void updateSizes(scale)
DERIVED& setScreenSize(..)
TypeCoordConverter& AddXConverter(..)
TypeCoordConverter& AddYConverter(..)
DERIVED& ZoomOnRect(Rect)
DERIVED& ApplyZoomFactor()
DERIVED& Scroll(..)
GraphElementParent methods
virtual void ZoomX(xLeft, xRight)
virtual void ZoomY(yTop, yBottom)
virtual void ScrollX(..)
virtual void ScrollY(..)
virtual void RefreshFromChild(RefreshStrategy)
Elements management
T& AddElement<T>(..)
T& CloneElement<T>(..)
T& CreateElement<T>(..)
T& CreateLegendElement<T>(..)
1
*
<< CRTP >>
CRTPGraphElementFrame
<<CRTP>>
CRTP_StdGraphDraw
LogCoord

Subject: Re: New graph packages
Posted by koldo on Mon, 24 Sep 2012 19:13:09 GMT
View Forum Message <> Reply to Message

Hello Ratah

Didier is right.

I would do:

scatter.SetXYMin(xmin, ymin).SetRange(-10, 10);

Subject: Re: New graph packages
Posted by Didier on Sat, 26 Jan 2013 19:13:15 GMT
View Forum Message <> Reply to Message

Hi all,

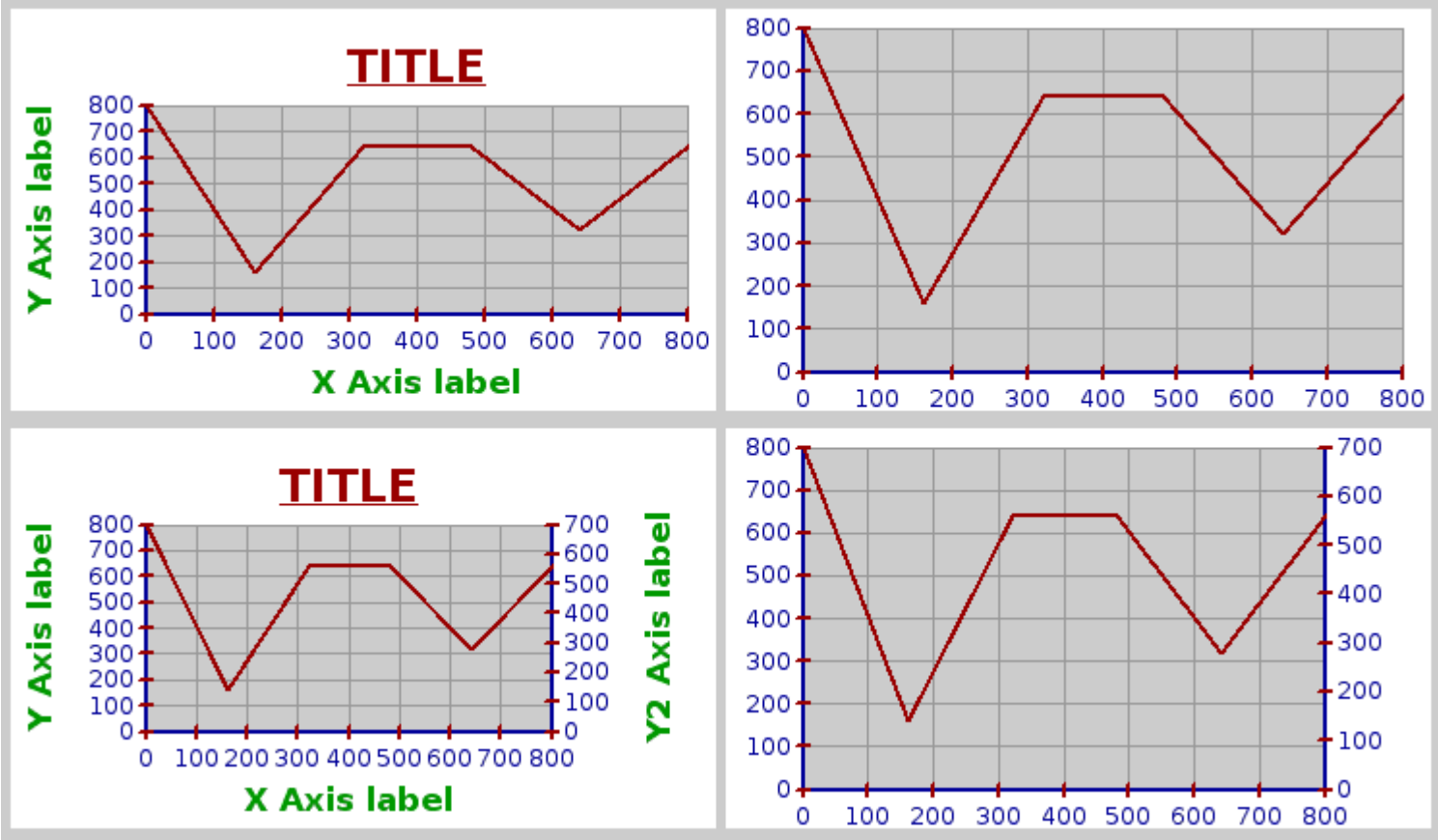
I just updated my GraphDraw/GraphCtrl Packages in the svn sandbox.
I added some predefined Graphs along with their description in '.usc' files to allow easy use in layout designer.

I tried to make the look in designer the closest possible to the real Graphs: so almost ALL parameters are modifiable directly in designer :

colors
fonts
width
margins
axis scale type : STD/LOG
Axis format type: STD/LOG/DATE/TIME
Labels (axis labels, title)
... and more

Here is how the four predefined graphs look when dropped in the designer

File Attachments
1) GraphCtrls.png, downloaded 1560 times



Subject: Re: New graph packages
Posted by Didier on Sat, 26 Jan 2013 19:18:43 GMT
View Forum Message <> Reply to Message

Here is what the application displays for the previous layout designer config:
edit : image deleted

Subject: Re: New graph packages
Posted by Didier on Fri, 01 Feb 2013 19:36:31 GMT
View Forum Message <> Reply to Message

Hi all,

I just uploaded some more modifications

Stacking priority management (manages the order in which elements are stacked on the sides)
Legend drawing corrections (scaling issues)
Added legend to StdGraphs + USC
Added Title + Legend position management (can be put on any side of the graph)
Some refactoring

I also just noticed that some files were missing causing no compile and no '.usc' file

Can somebody try it in the designer to see if I forgot something else (normally no)

Subject: Re: New graph packages
Posted by Didier on Mon, 11 Feb 2013 23:08:24 GMT
View Forum Message <> Reply to Message

Hi all,

I uploaded a lot of corrections and changes including:

popup menu for changing of element properties (only base properties for the moment)(legend, labels, title)
popup menu for changing series properties
series dynamic show/hide through popup menu

A lot of the code is still 'in progress' but things are coming to shape now (although some parts still need a heavy refactoring)

I will upload a new demo soon

Subject: Re: New graph packages
Posted by Didier on Fri, 22 Feb 2013 16:45:08 GMT
View Forum Message <> Reply to Message

Hi all,

I've uploaded some more corrections and enhancements since last post:

Refactoring
Bug corrections
Added Label hide management in .usc
WIN compile issue
Mouse ctrl corrections for elements that are hidden
Added LabelPropertiesEditorDlg

Has anybody tried this Ctrl ??

Subject: Re: New graph packages
Posted by Didier on Sat, 04 May 2013 13:18:32 GMT
View Forum Message <> Reply to Message

Hi all,

I've uploaded to svn sandbox some more corrections:
* compiles with lattest ScatterDraw modifications from Koldo
* other enhancements and corrections
  * legend has exact same aspect as serie in the graph
  * ...

EDIT: image deleted

Subject: Re: New graph packages
Posted by 281264 on Mon, 20 May 2013 19:56:53 GMT
View Forum Message <> Reply to Message

Didier,

I have tested this Ctrl and it works fine; I'd suggest documenting it better.

Thanks,

Javier

Subject: Re: New graph packages
Posted by Didier on Tue, 21 May 2013 17:50:42 GMT
View Forum Message <> Reply to Message

Hi,

Thank you for trying it.

I will try documenting it now that the interface is a bit more stable

Subject: Re: New graph packages
Posted by jerson on Wed, 22 May 2013 01:49:21 GMT
View Forum Message <> Reply to Message

I am not sure if this has already been covered in the widget. However, couple of things which I have used in the past have been a LOG-LOG scale and NYQUIST(also known as Polar) plot.

The nyquist plot would make the graph pretty complete in all respects and adding it is just a variation of pie. I could try, but, as you would realize, I am not upto speed with this right now. This would make it useful to many engineers.

This is what I mean by nyquist plot

Picked the reference from this link

Subject: Re: New graph packages
Posted by Didier on Wed, 22 May 2013 20:02:29 GMT
View Forum Message <> Reply to Message

Hi Jerson,

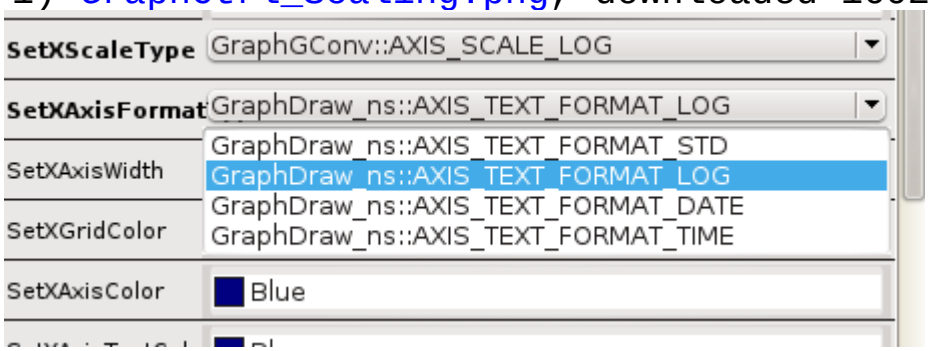
well if you want to have LOG/LOG : no problem, you can even configure it through LayoutDesigner.
You can even have LOG scaling while having normal style graduation, all is configurable ... and extensible

For the Nyquist graph, I don't have polar styled graph since I haven't needed it for now.

If I have time I may try (but I don't garanty it)

File Attachments

1) GraphCtrl_Scaling.png, downloaded 1602 times



Subject: Re: New graph packages
Posted by jerson on Thu, 23 May 2013 00:40:12 GMT
View Forum Message <> Reply to Message

Hello Didier

Thank you for considering the option. I am not in need for it right now. I used it in the past to do some scientific natured software and thought it will make the package complete. Nyquist seems to be the only thing missing. Overall, it's an excellent piece of work which I am learning from.

Regards
Jerson

Subject: Re: New graph packages
Posted by Didier on Mon, 14 Oct 2013 21:40:32 GMT
View Forum Message <> Reply to Message

Hi all,

I've updated GraphCtrl/GraphDraw packages in svn sandbox.
The following enhancements/corrections were done:

correction : Mouse move management now uses LocalLoop to catch mouse when scrolling and remove a nasty bug when moving mouse from one graph to another
Some cleaning up of code
Added some documentation (work in progress)
Added FitToData() method (for each axis, and for all graph)
Added UNDO/REDO capacities
Added UNDO/REDO automatic merging ==> when actions are done very quickly(<500ms) (like wheel scroll or zoom), they are merged into ONE undo / redo action

Undo/redo works for all the move/scroll actions.
Properties undo/redo is next to come

Subject: Re: New graph packages
Posted by Didier on Fri, 01 Nov 2013 12:04:00 GMT
View Forum Message <> Reply to Message

Hi all,

Here comes the last updates concerning the GraphDraw/GraphCtrl. The following modifications have been made :

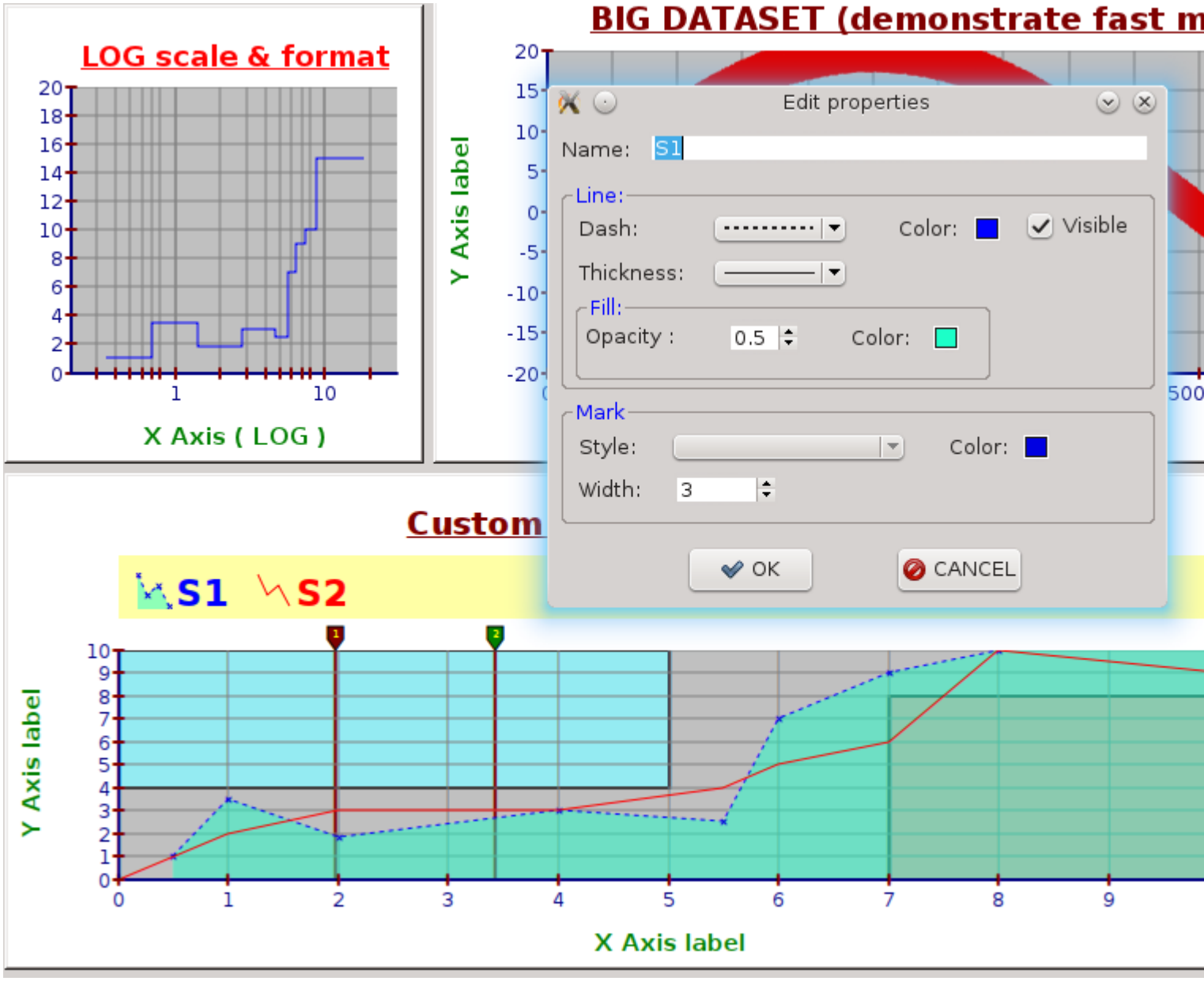
Code cleaning / refactoring / reorganisation
Undo/redo additions
Undo/Redo refactoring to ease it's use in GraphElements
Added MarkerElement/DynamicMarkerCtrl
refactored TickMark class (needed for MarkerElement)
Series Properties Dialog refactored

The main addition is the management of DynamicMarkers : it allows to manage markers that can be freely moved on the side of the graph (to measure distances ... or anything else). It is still work in progress (how to pass parameters to TickMark classes) but a working example is show in GraphCtrl_test Package

The look and feel of the markers used is completely configurable and depends only on the TickMark class used

File Attachments

1) GraphCtrls.png, downloaded 1511 times



Subject: Re: New graph packages
Posted by Didier on Mon, 11 Nov 2013 22:00:20 GMT
View Forum Message <> Reply to Message

Hi all,

I just uploaded some more updates to GraphDraw/GraphCtrl.
The modifications brought are the following:

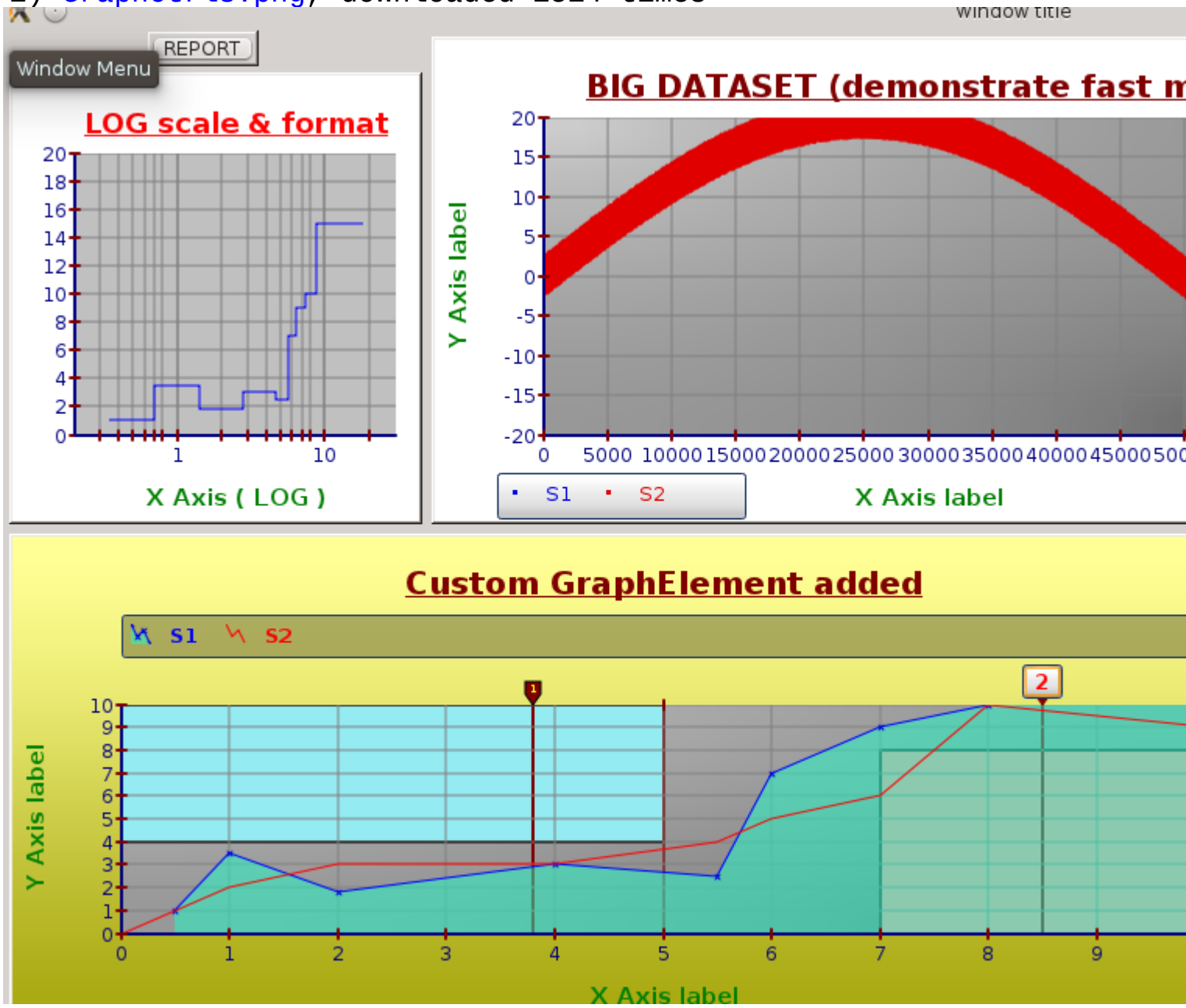
DynamicMarker refactoring
Added some skinning possibilities to the Graph and to the TickMarks used for DynamicMarkers

some code cleaning
some documentation

The next modifications to come :
Their is an issue when moving a marker with the mouse ==> all the graphs get's drawn ... so if the grap has big data, moving the marker is not smooth : I will try to correct this point.

File Attachments

1) GraphCtrls.png, downloaded 1514 times



Subject: Re: New graph packages
Posted by Didier on Sat, 16 Nov 2013 23:47:22 GMT
View Forum Message <> Reply to Message

Hi all,

I just uploaded some updates :

DynamicMarker::CursorImage() correction
Code cleaning / refactoring
DOCUMENTATION (big effort)
Added WhenMove callbacks to DynamicMarker and added example usage in GraphCtrl_test

The documentation is now starting to be consistant but there is still much to do.
If someone reads the doc and has remarks (phrase not clear enough for example), I'll be glad to take into account the remarks

Subject: Re: New graph packages
Posted by Didier on Sat, 21 Dec 2013 22:21:15 GMT
View Forum Message <> Reply to Message

Hi all,

I just updated my GraphCtrl/GraphDraw packages.
The modifications are the following:

Code cleaning and refactoring
Some bug corrections (only one or two corrections)
DRAWING SPEED enhancement when used as a Ctrl
BackGround style managment has been homogenized through the different classes (SetBackGndStyle() method)
Graph Background can now be transparent

The main enhancement is the update DRAWING SPEED when used as a Ctrl : now moving a Marker on the graph is very fluid (and so will the other GraphElements). It is so fast that it stays usable when the grap is in full screen (which wasn't the case before).

The known problems/bugs remaining are the following:

SUBPIXEL painting mode is no more useable and I hope correct it soon (see http://www.ultimatepp.org/forum/index.php?t=msg&goto=414 62&&srch=GraphCtrl#msg_41462)
Some sync bugs have been injected since last refactorings (these bugs were introduced a few svn deliveries ago)

Subject: Re: New graph packages
Posted by Didier on Mon, 23 Dec 2013 22:33:00 GMT
View Forum Message <> Reply to Message

SUBPIXEL issue is now corrected thanks to mirek.

Subject: Re: New graph packages
Posted by Didier on Sun, 05 Jan 2014 22:48:39 GMT
View Forum Message <> Reply to Message

Hi all,

I just delivered to svn sandbox an update of GraphDraw/GraphCtrl.
The modifications are listed below:

* GraphElementCtrl_FloatMoveResize : improved re-size with full border/corner selection and adapted MouseCursor when resizing
* Added LocalLoopSelect & Select style in GraphCtrl
.....* In fact this draws nice blue transparent rectangles instead of the animated rectangle obtained using RectTracker
.....* Select Rect style is completelly configurable (Blue is default)
* StdGridAxisDrawCtrl:
* Added AXIS SELECT-ZOOM
.....* Added LocalLoopSelect
.....* Added configurable select style
* Added global KEY management ==> replace K_CTRL, ... by predefined constants K_ZOOM, K_SCROLL, ...
 only a starting point, may do more if needed
* BUG : Corrected BUGs when LEFTMOUSE pressed outside GRAPH and moves into graph area

* Undo stack :
.....* Added isModified Verification before adding UndoAction to UNDO STACK
.....* Added UNDOSTACK limitation (100 undo actions max by default)
* Layout() requests :
.....* Painting done in 'fastPaint' mode when requested by Layout() ==> Application resizing
.....* Added FULL repaint 300 ms after last Layout() request
.....==> this gives fluid application re-size (no app freeze when re-sizing)
* Add AllowZoom, AllowScroll for X/Y management
* Added additional methods in GraphCtrl.usc
.....* UseLocalSelectLoop
.....* DisableGraphZoom

.....* DisableGraphScroll
.....* DisableXZoom
.....* DisableYZoom
.....* DisableXScroll
.....* DisableYScroll
.....* DisableAxisScroll
.....* DisableAxisZoom

The major improvements are:
* Stylable selection rectangle
* no app freeze when re-sizing application (in case of big data)
* Zoom/Scroll authorizations completely configurable through layout designer

---

Subject: Re: New graph packages
Posted by ManfredHerr on Fri, 10 Jan 2014 20:09:14 GMT
View Forum Message <> Reply to Message

Hi Didier,

why can't I see your GraphDraw/Ctrl package neither in bazaar of upp-nightly nor in upp-mirror-read-only?

Just a hint...

Manfred

Subject: Re: New graph packages
Posted by Didier on Fri, 10 Jan 2014 20:58:31 GMT
View Forum Message <> Reply to Message

Hi ManfredHerr,

These packages are only available in svn sandbox for the moment, but I hope to be able to upload them to bazaar this WE

The svn sandbox can be found here:
svn://www.ultimatepp.org/upp/sandbox

Subject: Re: New graph packages
Posted by ManfredHerr on Sat, 11 Jan 2014 13:59:12 GMT

View Forum Message <> Reply to Message

I got it! Thank you very much for the prompt response!
Merci beaucoup!

Edit:
I tried to have fun with it a little bit and detected an issue:
If you drag in the "Big Data" Area and move the hand outside the window before you release the mouse button then no update of the graph takes place unless you enter the window again. Is this intended?

Subject: Re: New graph packages
Posted by Didier on Sat, 11 Jan 2014 21:55:09 GMT
View Forum Message <> Reply to Message

In fact this was the first implementation I made to try to prevent abusive FULL refresh (which can take some time) ==> Refresh is made (if needed) when the mouse moves over the graph.

I was already thinking about modifying this since all the recent changes I made made new solutions available.

So I guess the time has come

Subject: Re: New graph packages
Posted by Didier on Sun, 12 Jan 2014 15:15:28 GMT
View Forum Message <> Reply to Message

Hi all,

I just delivered in SVN.6763 (sandbox) the following modifications for GraphCtrl/GraphDraw:
* Documentation
* Added DynamicMarkerCtrl::ResetMarkers() ==> is called through [popup menu]/[Reset markers]
* Added FULL REFESH on MouseLeave() to prevent letting Graph partially drawn when mouse is outside the GraphCtrl
* Add management of GraphElements with NEGATIVE priority (several elements with negative prio are allowed on same side)
* GridAxisDraw::Paint() now takes in account StackingPriority to paint the best way possible

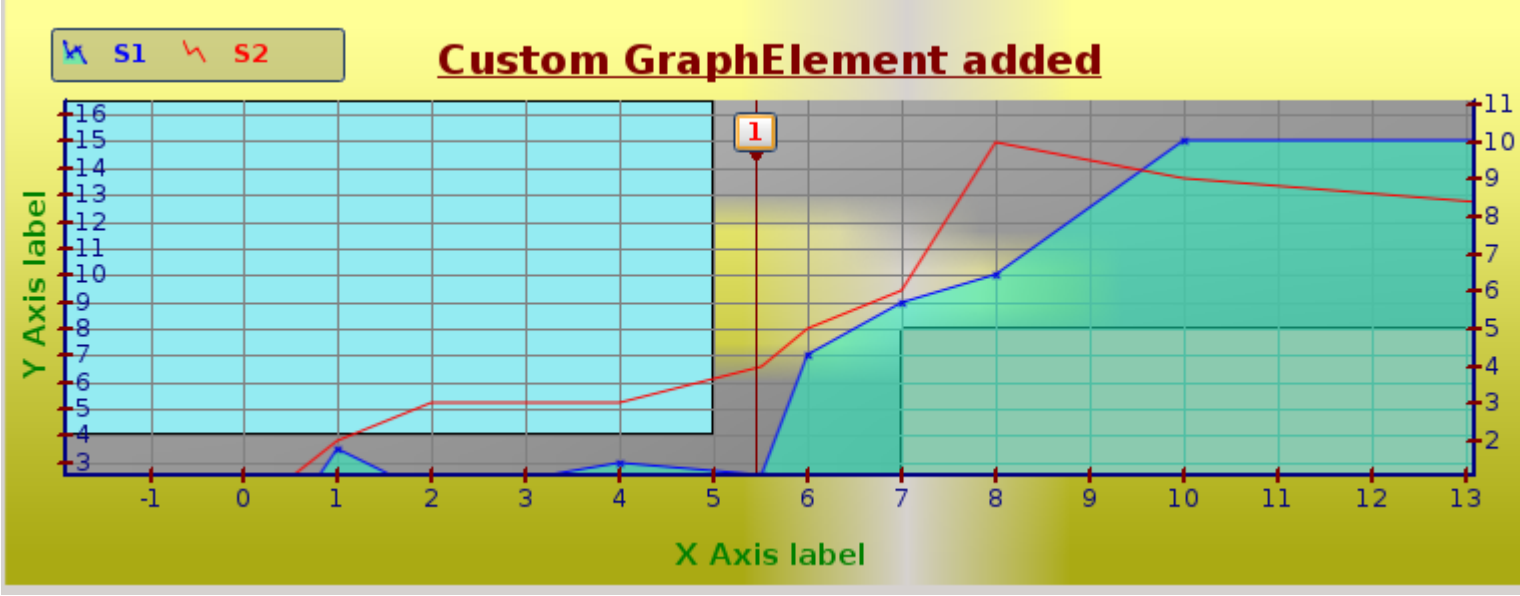
With NEGATIVE Stacking Priority values you can make Graphs like this :

Left Y axis is displayed inside Graph
Markers are displayed inside graph
All the Element's UI controls stays functional (Scroll, zoom, marker move, popup menus, ...)

ManfredHerr: The issue you detected is now corrected => if mouse leaves the GraphCtrl : GraphCtrl schedules a refresh

File Attachments

1) mirek1.png, downloaded 1386 times



Subject: Re: New graph packages
Posted by Didier on Tue, 14 Jan 2014 22:43:11 GMT
View Forum Message <> Reply to Message

Hi all,

I just delivered some new modifications:
** Added FULL KBD support
Added KBD shortcut for: for UNDO/REDO & COPY
Added LEFT/RIGHT/UP/DOWN scroll KBD actions
KBD shortcuts for context menu

** Added AUTO-HIDE feature : When using AutoHideElementCtrl<BASE_ELEMENT> template the GraphElement Hides once the mouse is not on it
==> this brings the possibility of adding tools on graphs that do not parasite the visual aspect of the graph

The GraphCtrl_test application has a working example of the AUTO-HIDE feature

---

Subject: Re: New graph packages
Posted by Didier on Thu, 16 Jan 2014 18:44:18 GMT
View Forum Message <> Reply to Message

Reply to : http://www.ultimatepp.org/forum/index.php?t=msg&&th=8159&goto=41747#msg_41747
Quote:Didier,

tried it again on my home computer, good news: it works now!

I had to extract the packages GraphDraw & GraphCtrl & DropLineDash into an own directory to work.
Sandbox has a "bad" plugin/tif...

BTW, your GraphCtrl_test.upp has a TimingPolicies package I can't find anywhere, thus is not needed any where ?!?.

Cheers Ralph

Hello Ralph,

I am glad to hear that it now compiles and works on MSC10.

The TimingPolicies package is only a helper package that I use when I need to do some precise time mesures. It is required by GraphDraw only if GD_TIMINGS compilation option is set, so by default it is not needed

Subject: Re: New graph packages
Posted by Didier on Tue, 18 Feb 2014 22:40:22 GMT
View Forum Message <> Reply to Message

Hi all,

Here is another post about the lattest GraphCtrl/GraphDraw modifications (some which have been present in svn sandbox but for which I didn't post any message:

The main features added/modified are the following:
[*] Code completion now works on GraphCtrl
==> this particular point makes GraphCtrl much easier to use (I also modified a bit the inheritance tree to take advantage of the hierarchy in the completion by layering functionalities)

[*] Grid step take in account text size ==> grid auto resizes according to text size
==> It is not perfect (some artifacts are still there), but it globally works much better than before.

18/02/2014 - SVN.6927
---
* Added DrawFocus(bool) in '.USC' file to enable/disable focus drawing (in case the layout contains only one Ctrl and don't need to draw the Focus)
* Refactored GridAxisDraw and GridStepIterator
  - Added management for tick level : majorTickMark and minorTickMark
  - Axis text size automatically taken in account to adjust grid step
  - Corrected LOG scale management

31/01/2014 - SVN.6840
---
* COMPLETION now works with on FULL class hierarchy of GraphCtrl !!!
* Refactored GraphDraw instance classes in order to improve usability through taking advantage of inheritance tree in completion

25/01/2014 - SVN.6811
---
* Corrected FAST PAINT for PARAMETRIC functions
* Added AddBlankArea()
* Added ExecuteWithUndo()
* Corrected KBD scroll & FitToData authorisations
* Replaced Margins with BlankAreas
* Corrected FOCUS management and Added WantFocus to .USC

Subject: Re: New graph packages
Posted by Didier on Sun, 23 Feb 2014 21:56:00 GMT
View Forum Message <> Reply to Message

Hi all,

I just uploaded to svn some corrections concerning automatic management of grid scaling.
Hopefully this corrects all the artifacts (mostly visible when scrolling)

One bug still remaining : GetImage() get's bad grid (needs to be recalculated first)

Subject: Re: New graph packages

Posted by Didier on Wed, 05 Mar 2014 23:30:34 GMT
View Forum Message <> Reply to Message

Hi I just uploaded some new fixes.

I recommend to update to this version if you are using it : it contains some uninitialized data fixes.

Complete modifications are listed below :

05/03/2014 - SVN.6999
---
* Corrected some initialized vars issues.
* Added missing setGraphSize() methods (regression since hierarchy refactoring)
* Added some predefined Graphs usable in GraphDraw (no need to be Ctrl)
* Added ChangeStatus class to introduce change management in some parts of code
* Renamed + Refactoring GridStepIterator.h ==> GridStepManager.h + GridStepManager.cpp
* Refactoring : code simplification : replaced all TypeGridStepManager::Iterator with GridStepIterator
* Corrected GetImage() regression ==> grid is now correctly resized according to scale

It took me quite hard time to correct the GetImage() regression because some Size() variables were not initialized ...
I didn't noticed before that the default constructor Size() does NOT initialize internal members ..This of coarse making unpredictable behavior ...

The automatic grid adjustment (taking in account text size) now works quite well, if anyone stumbles on a corner case I'll be happy to hear about it

Subject: Re: New graph packages
Posted by 281264 on Sun, 09 Mar 2014 10:55:13 GMT
View Forum Message <> Reply to Message

Dear Didier,

It would be interesting that you publish your application in Bazaar so that we can make the most of it. I really appreciate your effort for developping this interesting tool.

Cheers,

Javier

Subject: Re: New graph packages

Posted by Didier on Mon, 17 Mar 2014 20:06:51 GMT
View Forum Message <> Reply to Message

Hello Javier,

Sorry for late reply but I was away for a week.

I'm pleased to hear about you're interest for the GraphCtrl package but the package hasn't been accepted in bazaar for the moment although I requested that some time ago.

Anyway it's no big deal to pull the svn repo and the Packages aren't very stable (although getting more more stable)