Subject: Styles and Widgets
Posted by WebChaot on Wed, 14 Mar 2007 11:15:27 GMT

View Forum Message <> Reply to Message

Hi all!

I made some new components and changes in a previous version (610 dev 1). This week my collegue downloaded the last release (2007) and I saw, that some changes are done on chameleon technology.

1. I have to paint some widgets by code. Is there any example, how to do this in the new way (Style().Write() ??)? I'm very interested in an easy way to use the standard upp widgets and only draw them myself. Especially combined widgets (for example a dropdown-box, where I have the box AND the dropdown-button - or a listbox, where I also have to draw the scrollbar).

I dont want to reinvent the wheel and make all widgets new with a specific paint-method. I want to use the standard widgets and style them in an easy way.

2. I didn't look in deep of new release. But what I missed in older version is a style for StatusBar and something for "CoolBar" (the docking panel where windows toolbars are placed). I know, there is still no CoolBar and no Docking ability in upp for now. But maybe the style could be included and for example be assigned on a dummy StaticRect or something else.

3. I put also my third question here, but its not really a chameleon topic: I want create widgets at runtime. Creation is not the problem (I think) - but how to iterate over all widgets of a window for example? How to detect its class? Is that possible? The reason is simple: I have to collect input data of all edit fields - but because they are created by code I need a solution to iterate over all fields on a window.

Finally I have a question about HTTP and HTTPS - but this one I append on the right thread ...

PS: We are using upp now for some month - and we are still satisfied. We are able to do all the stuff we need - great work!

Thanks in advance,

WebChaot.

Subject: Re: Styles and Widgets
Posted by mirek on Wed, 14 Mar 2007 13:54:27 GMT

View Forum Message <> Reply to Message

WebChaot wrote on Wed, 14 March 2007 07:15Hi all!

I made some new components and changes in a previous version (610 dev 1). This week my

collegue downloaded the last release (2007) and I saw, that some changes are done on chameleon technology.

1. I have to paint some widgets by code. Is there any example, how to do this in the new way (Style().Write() ??)?

It is not really too much different.

There were two reason to changes things:

"index based" method of working with style became to be overly complicated, therefore we changed indicies for structures. Usually, those structures are pretty easily understandable.

If you had any code based on indicies, conversion to structures should be pretty straightforward, usually there is one structure data fiels per former index...

While being at it, structures also provide a good way how to allow per-thread customization. So you can either assign a reference to the style for individual widget, or you can use "Write" to change the style for the whole class.

Quote:
2. I didn't look in deep of new release. But what I missed in older version is a style for StatusBar and something for "CoolBar" (the docking panel where windows toolbars are placed). I know, there is still no CoolBar and no Docking ability in upp for now. But maybe the style could be included and for example be assigned on a dummy StaticRect or something else.

OK.

Quote:
3. I put also my third question here, but its not really a chameleon topic: I want create widgets at runtime. Creation is not the problem (I think) - but how to iterate over all widgets of a window for example?

Ctrl::GetFirstChild, Ctrl::GetNext

Quote:
 How to detect its class?

RTTI if you really need to.

Quote:
The reason is simple: I have to collect input data of all edit fields - but because they are created by code I need a solution to iterate over all fields on a window.

Ah. Well, there are better solution to this. See DynamicDlg example. In nutshell, you can create widgets in container, e.g. ArrayMap. GUI never owns widgets, therefore you can still use them in the dialog, but use container to access individual editors.

Mirek

Subject: Re: Styles and Widgets
Posted by WebChaot on Thu, 15 Mar 2007 08:14:29 GMT
View Forum Message <> Reply to Message

Hi Mirek!

>>> It is not really too much different.

OK. Then I did not realize how styles work. I'm not a very experienced upp-programmer (but have programming background in other languages) - so I have to learn from sources and examples. That worked quite good until now. But I do not understand the styles example

In past I created some classes based on upp standard widgets and do my own paint-methods. Thats boring, because after each update I have to update all these classes.

With styles - I guess - I can use standard widgets and only assign new paint-methods to that widgets on application startup (without writing own classes). Thats the theory. In practice I do not understand, how ChEllipse, EllipseLook(), MyLookFn() and INITBLOCK work together. Maybe there is another piece of code somewhere in this forum or in any other project?

What I need to do is to make an office 2003 or 2007 style to all widgets, if possible. Will I have to create each widget new and place my code in each paint-method or would there be a better way?

Would be happy about every information about this topic,

WebChaot.

PS: I'm not sure, if I can solve this problem. But if I finished my office-like widgets (NavigationBar, movable Toolbars, ...), I would share them here in forum, if someone would need them too.

Subject: Re: Styles and Widgets
Posted by mirek on Thu, 15 Mar 2007 11:11:11 GMT
View Forum Message <> Reply to Message

WebChaot wrote on Thu, 15 March 2007 04:14Hi Mirek!

>>> It is not really too much different.

OK. Then I did not realize how styles work. I'm not a very experienced upp-programmer (but have programming background in other languages) - so I have to learn from sources and examples. That worked quite good until now. But I do not understand the styles example

In past I created some classes based on upp standard widgets and do my own paint-methods. Thats boring, because after each update I have to update all these classes.

With styles - I guess - I can use standard widgets and only assign new paint-methods to that widgets on application startup (without writing own classes). Thats the theory. In practice I do not understand, how ChEllipse, EllipseLook(), MyLookFn() and INITBLOCK work together. Maybe there is another piece of code somewhere in this forum or in any other project?

What I need to do is to make an office 2003 or 2007 style to all widgets, if possible. Will I have to create each widget new and place my code in each paint-method or would there be a better way?

Would be happy about every information about this topic,

WebChaot.

PS: I'm not sure, if I can solve this problem. But if I finished my office-like widgets (NavigationBar, movable Toolbars, ...), I would share them here in forum, if someone would need them too.

Ah, now I see the problem. I thought you have been using first chameleon iteration (but you did it by overriding Paint).

Well, the important thing to understand about chameleon is that it is using "Value" to represent visual appearance.

Each widget is usually represented by one or more rectangular areas. Each such area has associated Value, or array of Values to define its appearance (array because area can have more states).

E.g. button has single rectangle that covers it all. It has several Values for normal, pushed etc states. Button::Paint chooses Value based on current state and calls ChPaint to paint the area of button (note that button text (and/or image etc..) is not a part of this process, it is applied later).

Now what gets painted depend on the Value. Value can contain anything and Chameleon allows the client code to register "value painter" based on its Value. Chameleon also supports two basic types - Image and Color - than can be used to paint area.

Now Image - there is important to know, that how Image is applied to area is ruled by positions of two reference points (HotSpot and 2ndSpot) that can be designed in Icon designer. Basically, you use them do designate which areas have to be stretched.

This way appearance of area is parametrized. Of course, more parameters than that are needed to customize widgets; all are stored in those Styles.

Mirek

Subject: Re: Styles and Widgets
Posted by WebChaot on Thu, 15 Mar 2007 11:39:11 GMT
View Forum Message <> Reply to Message

Hi Mirek!

Thanks for your answer. I'm not sure, if it brings light into dark or only new questions

>>> Well, the important thing to understand about chameleon is that it is using "Value" to represent visual appearance.

Maybe thats my problem: How can a function be a value and how it works?

Will I have to use the construct used in the example - or can I simply write a new PaintAnything()-function and Write() it to the four values?

Is it possible to post a simple example of how to redraw a droplist with dropbutton or a toolbar for example (only with a green rectangle or something like that)? Its not urgent - but maybe would help others too.

Thanks,

WebChaot.

Subject: Re: Styles and Widgets
Posted by mirek on Thu, 15 Mar 2007 13:59:54 GMT
View Forum Message <> Reply to Message

WebChaot wrote on Thu, 15 March 2007 07:39Hi Mirek!

Thanks for your answer. I'm not sure, if it brings light into dark or only new questions

>>> Well, the important thing to understand about chameleon is that it is using "Value" to represent visual appearance.

Maybe thats my problem: How can a function be a value and how it works?

Well, that is why I state it as "important thing to understand"

OK, start with something simple (and follow my mental process when designing the whole thing):

It can be Image. Image can be scaled to cover the target rectangle.

Step 2: It will not always look nice. Therefore define rectangular area within Image that is scaled competely; rest is border that is scaled in one dircetion only or not at all (think about how XP button looks like.

Step 3: Value is able to store anything. So if you need to paint something else (like use host platform theming API), create a special value type that contains parameteres describing what to paint. Then register your chameleon routine (ChRegister) that detects such values and provides custom painting.

See that Chameleon example again now please.... OTOH, even Step 2 already provides anything you ever needed to skin your application.

Mirek

Subject: Re: Styles and Widgets
Posted by WebChaot on Wed, 18 Apr 2007 09:24:05 GMT
View Forum Message <> Reply to Message

Hi Mirek!

>>> Ah. Well, there are better solution to this. See DynamicDlg example. In nutshell, you can create widgets in container, e.g. ArrayMap. GUI never owns widgets, therefore you can still use them in the dialog, but use container to access individual editors.

Project grows every day - I think, you will have a new demo application for your homepage in about one month. I understand DynamicDlg now and it works fine. But two more questions about this topic:

1. I make "containers" (=ArrayMaps) for each control type (I know, it would be possible to store all ctrls in one arraymap). That works fine. But I need a similar solution for menu items and toolbuttons. At the moment I store all related data in an array and on every change in menu I recreate the whole menu from that array: menu.Clear() and the step through the array and add each item (and this recursive). Is there an easier way - e.g. to store the menu items in an arraymap and delete them via .find() method directly (but what with clients)? Concerning this topic here is the second question:

2. Menus are build in tree structure. I would need that too for my gui-elements. I have to know, which elements are on which parent (e.g. nested staticrects, ...). In ArrayMap I can only store an "Id" and the control. Nice to find and get value or delete. But for my application I need a way to get all child values of the edit fields on an specific parent element.

For exampe: I have 2 Frames, a splitter - on right frame 2 nested staticrects - and in the inner one two edit fields. I would need now a function with the parameter "parent" which reads all edit fields which are on these parent at runtime. Should also work with parent of the second staticrect and with whole window (which reads all edit fields of the window).

I hope, you know what I mean.

How I would do that now:

Make structs for each edit element - with "ParentId" and the type of element.

For each of that structs an arraymap where I can find the elements via "Id" and then step recursive over the array and look for children ("ParentId" = "Id"). But this sounds a little bit exhausting to me, because I have to read the whole array many times recursively to parse the structure. Same when deleting some element and have to do that with all children before.

I'm sure, there is a better solution, but I dont know it yet

Would be nice, if you have some ideas about this topic.

Or in short form: What I need to do is completely create a window dynamically by code (via xml-like file). It must be possible to remove some elements (with all children on this element - also menu item) and then read all values of edit elements on an specific parent (from topwindow down to last level).

All the examples contains add() - but didn'n find any remove or step over all elements example

Thanks in advance,

WebChaot.

Subject: Re: Styles and Widgets
Posted by mirek on Wed, 18 Apr 2007 10:02:13 GMT
View Forum Message <> Reply to Message

I guess your problem can be summarized as "how to implement hierarchical structure that allows element removal"?

Well, one simple way is this:

```
struct CtrlsTree {
    One<Ctrl> ctrl;
    ArrayMap<String, CtrlsTree> child;
};
```

BTW, ArrayMap supports Remove (or Unlink). You do not have to care about rest; destryoing widget does everything needed to remove it from GUI.

Mirek

Subject: Re: Styles and Widgets
Posted by WebChaot on Wed, 18 Apr 2007 10:08:44 GMT
View Forum Message <> Reply to Message

Hi Mirek!

>>> I guess your problem can be summarized as "how to implement hierarchical structure that allows element removal"?

It could be so simple

>>> BTW, ArrayMap supports Remove (or Unlink). You do not have to care about rest; destryoing widget does everything needed to remove it from GUI.

Thanks a lot - I will try in the next days. When I understand the code right, removing an element also removes all children (automatically). Correct?

Thats the solution I was looking for. Great!

But that doesn't work for menu items - does it?

WebChaot

Subject: Re: Styles and Widgets
Posted by mrjt on Wed, 18 Apr 2007 10:24:22 GMT
View Forum Message <> Reply to Message

It sounds to me like you could do this using something as simple as:

```
Vector<Value> GetAllData(Ctrl *parent) throw (ValidationException)
{
  Vector<Value> data;

  for (Ctrl *c = parent->GetFirstChild(); c; c->GetNext()) {
    if (c->GetFirstChild())
      data.Append(GetAllData(c));
    else {
      Value v = c->GetData();
      if (v.IsError())
        throw ValidationException(c);
      else if (!v.IsVoid) // Prevents empty values
        data.Add(v);
    }
  }
  return data;
```

```
}
```

I just typed that in so no guarantees it will actually work (and the use of exceptions is slightly dubious) but I think the principle is correct. AFAIK only input ctrls ever return non-empty Values from GetData(). See TopWindow::Serialize() for another example (though not heirarchical).