Subject: Deprecating THISBACK

Posted by Klugier on Mon, 12 Oct 2020 20:31:33 GMT

View Forum Message <> Reply to Message

Hello,

I had the discussion with Mirek about this article https://evileg.com/en/post/629/ he found that the user uses deprecated THISBACK construction. Well we deprecated it in the documentation in favor or lambdas or newer c++ constructions such as Function, Gate etc.

In c++14 which we targets on all platforms the solution is easy. We can deprecate it on compilation level, which means something. Just mark THISBACK with deprecated atribute like this:

[[deprecated("THISBACK is deprecated use lambda instead.")]] const int thisbackd = 0;

```
#define THISBACK(x) callback(this + thisbackd, &CLASSNAME::x)
#define THISBACK1(x, arg) callback1(this + thisbackd, &CLASSNAME::x, arg)
#define THISBACK2(m, a, b) callback2(this + thisbackd, &CLASSNAME::m, a, b)
#define THISBACK3(m, a, b, c, d) callback3(this + thisbackd, &CLASSNAME::m, a, b, c, d)
#define THISBACK5(m, a, b, c, d, e) callback5(this + thisbackd, &CLASSNAME::m, a, b, c, d, e)
```

Then in all places when THISBACK is used the warning pop up "/home/klugier/upp/uppsrc/CtrlLib/FileSel.h (239): warning: 'thisbackd' is deprecated: THISBACK is deprecated use lambda instead. [-Wdeprecated-declarations]".

If you are not aware how you should replace thisback. Example below:

```
void MyWidnow::ErrorMenu(Bar& bar)
  // bar.Add("Copy all", THISBACK1(CopyError, true));
  bar.Add("Copy all", [=] { CopyError(true); });
```

// In the header file no need for "typedef MyWindow CLASSNAME" declaration...

I think we have multiple places when [[deprecated]] macro can be used. If you search for "deprecated" in uppsrc then there is around 52 entries... Definitely not the topic for 2020.2, but for 2021.1...

Klugier

Subject: Re: Deprecating THISBACK

Posted by mirek on Mon, 12 Oct 2020 21:25:22 GMT

Klugier wrote on Mon, 12 October 2020 22:31Hello,

I had the discussion with Mirek about this article https://evileg.com/en/post/629/ he found that the user uses deprecated THISBACK construction. Well we deprecated it in the documentation in favor or lambdas or newer c++ constructions such as Function, Gate etc.

In c++14 which we targets on all platforms the solution is easy. We can deprecate it on compilation level, which means something. Just mark THISBACK with deprecated atribute like this:

[[deprecated("THISBACK is deprecated use lambda instead.")]] const int thisbackd = 0;

```
#define THISBACK(x) callback(this + thisbackd, &CLASSNAME::x)
#define THISBACK1(x, arg) callback1(this + thisbackd, &CLASSNAME::x, arg)
#define THISBACK2(m, a, b) callback2(this + thisbackd, &CLASSNAME::m, a, b)
#define THISBACK3(m, a, b, c, d) callback3(this + thisbackd, &CLASSNAME::m, a, b, c, d)
#define THISBACK5(m, a, b, c, d, e) callback5(this + thisbackd, &CLASSNAME::m, a, b, c, d, e)
```

Then in all places when THISBACK is used the warning pop up "/home/klugier/upp/uppsrc/CtrlLib/FileSel.h (239): warning: 'thisbackd' is deprecated: THISBACK is deprecated use lambda instead. [-Wdeprecated-declarations]".

I think we have multiple places when [[deprecated]] macro can be used. If you search for "deprecated" in uppsrc then there is around 52 entries... Definitely not the topic for 2020.2, but for 2021.1...

Klugier

I generally hate this. This would mean that I would have to fix a tons of code to avoid that warning for no good reason. Like a week of work...

Mirek

Subject: Re: Deprecating THISBACK
Posted by Klugier on Mon, 12 Oct 2020 22:28:59 GMT
View Forum Message <> Reply to Message

Hello Mirek,

I like your message - it is funny:) This is the task for student or someone who would like to help in terms of uppsrc. I can imagine first job in coputer software industry when you need to replace THISBACK with lambda:) You are doing this for your tow weeks, but finally you are proud of

yourself that you manage to do it.

From the high standards of software development process this is action in good direction. In the end we will have better code without THISBACK and typedef declaration and after 2/3 years we could completely remove THISBACK code base.

I compiled TheIDE with deprication for THISBACK and there are 1275 warnings...

Klugier

Subject: Re: Deprecating THISBACK

Posted by mirek on Mon, 12 Oct 2020 22:36:21 GMT

View Forum Message <> Reply to Message

Klugier wrote on Tue, 13 October 2020 00:28Hello Mirek,

I like your message - it is funny:) This is the task for student or someone who would like to help in terms of uppsrc. I can imagine first job in coputer software industry when you need to replace THISBACK with lambda:) You are doing this for your tow weeks, but finally you are proud of yourself that you manage to do it.

You do not understand. The problem is not the U++ sources.

I am currently maintaining 2 mission critical applications. The simpler one contains only 1800 THISBACKs. Think of all errors I would introduce by "fixing" it.

And this would be the same for every single user. I hate it when I am forced to rewrite perfectly good code for some stupid reason like somebody feels that some feature is not fancy anymore.

Attribute that produces warning is not a good path. Better is flag. Or maybe the attribute activated by flag. But then again, if it is flag, it can as well be compilation error.

Mirek

Subject: Re: Deprecating THISBACK

Posted by Klugier on Wed, 14 Oct 2020 3

Posted by Klugier on Wed, 14 Oct 2020 22:10:14 GMT

View Forum Message <> Reply to Message

Hello.

Backing to this problem. I see that several templates (.upt) have CLASSNAME declaration. From my perspective the good idea would be to remove that declaration. After that all new code shouldn't have dependency to THISBACK:

```
CtrlLib/Complex.upt:
```

```
class <:classname:> : public <:lay > 1 ? "With" + classname + "Layout<TopWindow>" :
"TopWindow":> {
  public:
  typedef <:classname:> CLASSNAME;
  <:classname:>();
};
...

Could be:
...
class <:classname:> : public <:lay > 1 ? "With" + classname + "Layout<TopWindow>" :
"TopWindow":> {
  public:
  <:classname:>();
};
...
```

Skylark and Tutrle templates are also affected...

Klugier

Subject: Re: Deprecating THISBACK Posted by mirek on Wed, 14 Oct 2020 23:01:23 GMT View Forum Message <> Reply to Message

Klugier wrote on Thu, 15 October 2020 00:10Hello,

Backing to this problem. I see that several templates (.upt) have CLASSNAME declaration. From my perspective the good idea would be to remove that declaration. After that all new code shouldn't have dependency to THISBACK:

CtrlLib/Complex.upt:

```
...
class <:classname:> : public <:lay > 1 ? "With" + classname + "Layout<TopWindow>" :
"TopWindow":> {
public:
typedef <:classname:> CLASSNAME;
<:classname:>();
};
```

Could be:

...

class <:classname:> : public <:lay > 1 ? "With" + classname + "Layout<TopWindow>" :
"TopWindow":> {
 public:
 <:classname:>();
};
...

Skylark and Tutrle templates are also affected...

Klugier

Agreed. Will do.

Mirek

Subject: Re: Deprecating THISBACK
Posted by Oblivion on Wed, 14 Oct 2020 23:14:26 GMT
View Forum Message <> Reply to Message

Hi,

FYI, since you've agreed, I have removed the CLASSNAME declaration from the Turtle upt. :)

Best regards, Oblivion

Subject: Re: Deprecating THISBACK Posted by Klugier on Sat, 14 Nov 2020 23:44:16 GMT

View Forum Message <> Reply to Message

Hello Mirek,

We should also remove CLASSNAME from layout visgen for consistancy:

Klugier

File Attachments

1) LayoutDesignerCodeGenerationDialog.png, downloaded 813 times

Subject: Re: Deprecating THISBACK

Posted by mirek on Sat, 14 Nov 2020 23:58:56 GMT

View Forum Message <> Reply to Message

Good catch, yes!

Subject: Re: Deprecating THISBACK

Posted by mirek on Sun, 15 Nov 2020 00:42:32 GMT

View Forum Message <> Reply to Message

Done.

Subject: Re: Deprecating THISBACK

Posted by Lance on Sun, 15 Nov 2020 23:58:06 GMT

View Forum Message <> Reply to Message

Hi Klugier:

Thank you very much for all your effort!

As I go through current U++ document, in particular in this link https://www.ultimatepp.org/srcdoc\$CtrlLib\$Tutorial\$en-us.htm I#13

I still see references to THISBACK and CLASSNAME (just search in the webpage). I believe it should be revised to reflect current U++ trends and save some trouble for newcomers.

Thanks, Lance

Subject: Re: Deprecating THISBACK

Posted by Klugier on Mon, 16 Nov 2020 22:37:06 GMT

View Forum Message <> Reply to Message

Hello Lance.

Thanks for reporting! I removed all THISBACK and CLASSNAME from GUI tutorial. Please see tomorrows version. There are probably other places, but GUI tutorial is the most significant one.

Subject: Re: Deprecating THISBACK

Posted by MeerMusik on Sun, 23 May 2021 10:41:36 GMT

View Forum Message <> Reply to Message

Hello. As someone who just started learning UPP:

Is 'THISBACK' for Menus etc. also deprecated or just for connecting Items with Functions?

Thanks!

Subject: Re: Deprecating THISBACK

Posted by mirek on Sun, 23 May 2021 11:17:58 GMT

View Forum Message <> Reply to Message

THISBACK is deprecated. In U++ that does usually mean it will most likely be supported for ever, but there are better options now (lambdas).

Mirek

Subject: Re: Deprecating THISBACK

Posted by MeerMusik on Sun, 23 May 2021 11:37:01 GMT

View Forum Message <> Reply to Message

mirek wrote on Sun, 23 May 2021 13:17THISBACK is deprecated. In U++ that does usually mean it will most likely be supported for ever, but there are better options now (lambdas).

Mirek

Just to be sure I get this right: To create Menu Entries and similar: I should use Lambdas as well?

Subject: Re: Deprecating THISBACK

Posted by Klugier on Sun, 23 May 2021 12:46:53 GMT

View Forum Message <> Reply to Message

Hello MeerMusik.

Yes, you should use lambdas instead of THISBACK for creating menus. Last year, I updated our GUI tutorial to illustrate these changes. Please, check all topics related to menus:

- 7. Menu (Also 8, 9 & 10 are valuable)
- 11. Toolbar

Subject: Re: Deprecating THISBACK

Posted by MeerMusik on Sun, 23 May 2021 12:49:58 GMT

View Forum Message <> Reply to Message

Klugier wrote on Sun, 23 May 2021 14:46Hello MeerMusik,

Yes, you should use lambdas instead of THISBACK for creating menus. Last year, I updated our GUI tutorial to illustrate these changes. Please, check all topics related to menus:

- 7. Menu (Also 8, 9 & 10 are valuable)
- 11. Toolbar

Klugier

Thank you! I will update my UPP "Test Code" from 2018 then :)