Subject: Enhancing project templates (upt files)
Posted by Werner on Tue, 19 Sep 2006 14:57:58 GMT
View Forum Message <> Reply to Message

It seems to be a well-established standard to write CAPITALIZED include guards. But this is not possible when using project templates.

As I'm writing my own project templates anyway, I decided to patch Template.cpp to enable uppercase include guards (which might be selected by a corresponding option).

The patch requires just a minor change of a single function:

Original function

```
ArrayMap<String, EscValue> TemplateDlg::MakeVars0()
{
 ArrayMap<String, EscValue> var;
 String n = ~package;
 int q = n.ReverseFind('/');
 var.Add("PACKAGE", q >= 0 ? n.Mid(q + 1) : n);
 return var;
}
```

Patched function

```
ArrayMap<String, EscValue> TemplateDlg::MakeVars0()
{
 ArrayMap<String, EscValue> var;
 String n = ~package;
 int q = n.ReverseFind('/');
 n = q >= 0 ? n.Mid(q + 1) : n;
 var.Add("PACKAGE", n);
 var.Add("PACKAGE_UPPERCASE", ToUpper(n));
 return var;
}
```

Might I suggest to consider the adoption of this patch into the official Ultimate++ release?

If so, please let me know as soon as possible, as I'm writing a project template documentation.

Werner

Subject: Re: Enhancing project templates (upt files)

Posted by mirek on Tue, 19 Sep 2006 16:27:00 GMT
View Forum Message <> Reply to Message

OK, patch accepted.

Mirek

Subject: Re: Enhancing project templates (upt files)
Posted by Werner on Wed, 20 Sep 2006 08:12:00 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 19 September 2006 18:27OK, patch accepted.

Mirek

Thanks.

Project template documentation in full progress ...

Please let me know if you change the identifier (key) "PACKAGE_UPPERCASE". Otherwise I assume for documentation purposes that this feature will be part of the next development release.

Werner

Subject: Re: Enhancing project templates (upt files)
Posted by mirek on Wed, 20 Sep 2006 08:30:09 GMT
View Forum Message <> Reply to Message

I have kept your patch as suggested.

Subject: Re: Enhancing project templates (upt files)
Posted by Werner on Thu, 21 Sep 2006 09:25:59 GMT
View Forum Message <> Reply to Message

When working on the project template documentation, I noticed that the "Console application (no U++)" template defines a non-standard main function:

The standard requires main() to return an int.

Furthermore the standard requires the strings to which argv[] points not to be constant.

Last but not least "stdio.h" should be "<iostream>".

After all, if somebody exploits TheIDE to write plain C++ programs, he or she should get the plain standard.

Werner

Subject: Re: Enhancing project templates (upt files)
Posted by mirek on Thu, 21 Sep 2006 13:50:46 GMT
View Forum Message <> Reply to Message

Werner wrote on Thu, 21 September 2006 05:25When working on the project template documentation, I noticed that the "Console application (no U++)" template defines a non-standard main function:

The standard requires main() to return an int.

Furthermore the standard requires the strings to which argv[] points not to be constant.

Last but not least "stdio.h" should be "<iostream>".

After all, if somebody exploits TheIDE to write plain C++ programs, he or she should get the plain standard.

Werner

Thanks. As I am not planning to do anything with templates in near future.. what about posting here all your templates when you are finished?

Subject: Re: Enhancing project templates (upt files)
Posted by Werner on Thu, 21 Sep 2006 18:46:20 GMT
View Forum Message <> Reply to Message

luzr wrote on Thu, 21 September 2006 15:50As I am not planning to do anything with templates in near future.. what about posting here all your templates when you are finished?

Just wait for my project template documentation which I hope to complete by the weekend.

As soon as it is published you will be swamped with templates ...

Seriously: as soon as the documentation is completed I have to switch back to a project which I pushed on the stack for some days. But I'll remember your suggestion in my own good time.

Werner

Subject: Re: Enhancing project templates (upt files) [FEATURE REQUEST]
Posted by Werner on Tue, 26 Sep 2006 10:17:06 GMT
View Forum Message <> Reply to Message

Thank you for adopting my suggestion.

After using "PACKAGE_UPPERCASE" for a couple of days, I'm no longer happy with the simplistic creation of its content. This was obviously an over-quick suggestion. Sorry!

As far as I can see, it is quite common - and I adhere to this quasi-standard - to capitalize mixed-uppercase-lowercase identifiers by inserting an underscore ("_") when the change from a lowercase letter to an uppercase letter indicates a new component.

That is why I wrote the following tiny function, which I suggest to include into Ultimate++, maybe into "uppsrc/ide".

I intentionally wrote it as a non-menber function of "String" (compare e. g., Stroustrup, The C++ Programming Language, Special Edition, 10.3.2; Sutter, Alexandrescu, C++ Coding Standards, 44; Meyers, Effective C++, 4.6 / 23).

```
String MkInclGuard(const String& name)
{
 String output("");
 int str_len = name.GetLength();
 for (int i = 0; i < str_len; ++i)
 {
  int ch = name[i];
  if (IsAlNum(ch))
  {
   output += ToUpper(ch);
   if (i < str_len - 1)
    if (IsLower(ch) && IsUpper(name[i + 1]))
     output += '_';
  }
  else
   output += '_';
 }
 return output;
}
```

The patched function then reads:

ArrayMap<String, EscValue> TemplateDlg::MakeVars0()

```
{
 ArrayMap<String, EscValue> var;
 String n = ~package;
 int q = n.ReverseFind('/');
 n = q >= 0 ? n.Mid(q + 1) : n;
 var.Add("PACKAGE", n);
 var.Add("PACKAGE_UPPERCASE", MkInclGuard(n));
 return var;
}
```

Werner

Subject: Re: Enhancing project templates (upt files) [FEATURE REQUEST]
Posted by mirek on Tue, 26 Sep 2006 10:26:40 GMT
View Forum Message <> Reply to Message

Werner wrote on Tue, 26 September 2006 06:17Thank you for adopting my suggestion.

After using "PACKAGE_UPPERCASE" for a couple of days, I'm no longer happy with the simplistic creation of its content. This was obviously an over-quick suggestion. Sorry!

As far as I can see, it is quite common - and I adhere to this quasi-standard - to capitalize mixed-uppercase-lowercase identifiers by inserting an underscore ("_") when the change from a lowercase letter to an uppercase letter indicates a new component.

That is why I wrote the following tiny function, which I suggest to include into Ultimate++, maybe into "uppsrc/ide".

I intentionally wrote it as a non-menber function of "String" (compare e. g., Stroustrup, The C++ Programming Language, Special Edition, 10.3.2; Sutter, Alexandrescu, C++ Coding Standards, 44; Meyers, Effective C++, 4.6 / 23).

```
String MkInclGuard(const String& name)
{
 String output("");
 int str_len = name.GetLength();
 for (int i = 0; i < str_len; ++i)
 {
  int ch = name[i];
  if (IsAlNum(ch))
  {
   output += ToUpper(ch);
   if (i < str_len - 1)
    if (IsLower(ch) && IsUpper(name[i + 1]))
     output += '_';
```

```
  }
  else
   output += '_';
 }
 return output;
}
```

The patched function then reads:

```
ArrayMap<String, EscValue> TemplateDlg::MakeVars0()
{
 ArrayMap<String, EscValue> var;
 String n = ~package;
 int q = n.ReverseFind('/');
 n = q >= 0 ? n.Mid(q + 1) : n;
 var.Add("PACKAGE", n);
 var.Add("PACKAGE_UPPERCASE", MkInclGuard(n));
 return var;
}
```

Werner

I think we should add it to the Core, it is a nice complement to "InitCaps" sort of function.

I only do not quite like the name (because it is not necessarily related to include guards) - what about something like "ToLowerUnderscoreCaps"?

Subject: Re: Enhancing project templates (upt files) [FEATURE REQUEST]
Posted by Werner on Tue, 26 Sep 2006 11:52:18 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 26 September 2006 12:26
I think we should add it to the Core, it is a nice complement to "InitCaps" sort of function.

I only do not quite like the name (because it is not necessarily related to include guards) - what about something like "ToLowerUnderscoreCaps"?

I don't stick to the name. But I think it shouldn't start with "ToLower" because it's more "ToUpper". And it shouldn't be to long.

What about "ToUpperMixedCase"?

Subject: Re: Enhancing project templates (upt files) [FEATURE REQUEST]
Posted by mirek on Tue, 26 Sep 2006 12:03:31 GMT
View Forum Message <> Reply to Message

Werner wrote on Tue, 26 September 2006 07:52luzr wrote on Tue, 26 September 2006 12:26
I think we should add it to the Core, it is a nice complement to "InitCaps" sort of function.

I only do not quite like the name (because it is not necessarily related to include guards) - what about something like "ToLowerUnderscoreCaps"?

I don't stick to the name. But I think it shouldn't start with "ToLower" because it's more "ToUpper".

Right But we can have both.

Quote:
What about "ToUpperMixedCase"?

I got it: ToUpper_Caps

Subject: Re: Enhancing project templates (upt files) [FEATURE REQUEST]
Posted by Werner on Tue, 26 Sep 2006 14:25:06 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 26 September 2006 14:03I got it: ToUpper_Caps

"Aye. I could do that."

It's up to you to decide. Just let me know your final decision, so that I can take it into account when upgrading the UPT documentation (refer to message #5463).

Werner

Subject: Re: Enhancing project templates (upt files) [FEATURE REQUEST]
Posted by mirek on Tue, 26 Sep 2006 19:02:45 GMT
View Forum Message <> Reply to Message

ArrayMap<String, EscValue> TemplateDlg::MakeVars0()

```
{
 ArrayMap<String, EscValue> var;
 String n = ~package;
 int q = n.ReverseFind('/');
 var.Add("PACKAGE", q >= 0 ? n.Mid(q + 1) : n);
 var.Add("PACKAGE_TOUPPER", ToUpper(n));
 var.Add("PACKAGE_TOUPPER_CAPS", ToUpper_Caps(n));
 return var;
}
```

Mirek

Subject: Re: Enhancing project templates (upt files) [FEATURE REQUEST]
Posted by Werner on Tue, 26 Sep 2006 19:36:28 GMT
View Forum Message <> Reply to Message

luzr wrote on Tue, 26 September 2006 21:02

```
ArrayMap<String, EscValue> TemplateDlg::MakeVars0()
{
 ArrayMap<String, EscValue> var;
 String n = ~package;
 int q = n.ReverseFind('/');
 var.Add("PACKAGE", q >= 0 ? n.Mid(q + 1) : n);
 var.Add("PACKAGE_TOUPPER", ToUpper(n));
 var.Add("PACKAGE_TOUPPER_CAPS", ToUpper_Caps(n));
 return var;
}
```

Mirek

If "package" holds a path and a filename,

only the filename goes into "PACKAGE", but
both the path and the filename go into "PACKAGE_TOUPPER" and
both the path and the filename go into "PACKAGE_TOUPPER_CAPS".

Is that what you want?

I myself would prefer to find just the filename in ("PACKAGE_TOUPPER" and)
"PACKAGE_TOUPPER_CAPS".

Werner

Subject: Re: Enhancing project templates (upt files) [FEATURE REQUEST]
Posted by mirek on Tue, 26 Sep 2006 20:58:05 GMT
View Forum Message <> Reply to Message

If I remember well, it should be "package/filename.h"

As there is high probability that that two packages have the same header, I think it is a good idea to add package name to the mix.

Mirek

Subject: Re: Enhancing project templates (upt files)
Posted by Werner on Sun, 03 Dec 2006 12:43:16 GMT
View Forum Message <> Reply to Message

luzr wrote on Thu, 21 September 2006 15:50As I am not planning to do anything with templates in near future.. what about posting here all your templates when you are finished?

You can find the first two in

"Function Test Frame" and "Function Test"

I couldn't find a more suitable topic.

Werner