Subject: Translation in static members
Posted by dolik.rce on Sat, 05 Feb 2011 13:58:56 GMT

View Forum Message <> Reply to Message

Is it possible to translate test in static member of a class?

Consider this simplified code:

```
#include <Core/Core.h>
using namespace Upp;

#define TFILE <test/test.t>
#include <Core/t.h>

struct test{
 static const char* str;
 const char* str2;
 test():str(t_("translation")){};
};
const char* test::str=t_("translation");

CONSOLE_APP_MAIN{
 SetLanguage(GetSystemLNG());
 test t;
 DUMP(t.str); // doesn't translate
 DUMP(t.str2); // works fine
 DUMP(t_("translation")); // works fine
}
```

I believe the problem is that the static member is initialized earlier than the translations. Is there some reasonable workaround? Or is the only solution to make the member non-static?

Thanks,
Honza

Subject: Re: Translation in static members
Posted by dolik.rce on Sat, 05 Feb 2011 17:29:15 GMT

View Forum Message <> Reply to Message

I'll answer myself

The simplest and probably correct solution is to use t_GetLngString() whenever using the static member, instead when initializing it:

```
#include <Core/Core.h>
using namespace Upp;

#define TFILE <test/test.t>
#include <Core/t.h>
```

```
//simple shorthand, to keep code nice looking
#define _t(X) t_GetLngString(X)

struct test{
 static const char* str;
};
const char* test::str=tt_("translation");

CONSOLE_APP_MAIN{
 SetLanguage(GetSystemLNG());
 test t;
 DUMP(_t(t.str)); //<- added _t() to translate the string at runtime
}
```

The _t macro is quite handy thing. It would actually work with t_ as well, but that confuses theide when syncing the translations. Maybe there could be some shorthand for t_GetLngString added in the U++, what do you think?

Honza

Subject: Re: Translation in static members
Posted by tojocky on Sat, 05 Feb 2011 17:38:21 GMT

View Forum Message <> Reply to Message

dolik.rce wrote on Sat, 05 February 2011 19:29I'll answer myself

The simplest and probably correct solution is to use t_GetLngString() whenever using the static member, instead when initializing it:#include <Core/Core.h>
using namespace Upp;

```
#define TFILE <test/test.t>
#include <Core/t.h>
```

```
//simple shorthand, to keep code nice looking
#define _t(X) t_GetLngString(X)

struct test{
 static const char* str;
};
const char* test::str=tt_("translation");

CONSOLE_APP_MAIN{
 SetLanguage(GetSystemLNG());
 test t;
 DUMP(_t(t.str)); //<- added _t() to translate the string at runtime
}
```

The _t macro is quite handy thing. It would actually work with t_ as well, but that confuses theide when syncing the translations. Maybe there could be some shorthand for t_GetLngString added in

the U++, what do you think?

Honza

This is very simple:

if you have static property than you can address only: ClassName::StaticPropertyName or from method of class by simple StaticPropertyName.

Hope if this help you!

Subject: Re: Translation in static members
Posted by mirek on Fri, 18 Feb 2011 11:38:38 GMT

View Forum Message <> Reply to Message

dolik.rce wrote on Sat, 05 February 2011 12:29I'll answer myself

The simplest and probably correct solution is to use t_GetLngString() whenever using the static member, instead when initializing it:

```
#include <Core/Core.h>
using namespace Upp;

#define TFILE <test/test.t>
#include <Core/t.h>

//simple shorthand, to keep code nice looking
#define _t(X) t_GetLngString(X)

struct test{
 static const char* str;
};
const char* test::str=tt_("translation");

CONSOLE_APP_MAIN{
 SetLanguage(GetSystemLNG());
 test t;
 DUMP(_t(t.str)); //<- added _t() to translate the string at runtime
}
```

The _t macro is quite handy thing. It would actually work with t_ as well, but that confuses theide when syncing the translations. Maybe there could be some shorthand for t_GetLngString added in the U++, what do you think?

Honza

I guess calling t_GetLngString or GetLngString here is not a big problem, as IME static texts are not that frequent.

However, if we decided on shortcat synonyme, it would be better done as inline function - no need to use macro here

Mirek

Subject: Re: Translation in static members
Posted by dolik.rce on Fri, 18 Feb 2011 15:08:24 GMT
View Forum Message <> Reply to Message

Whatever, in this case there is no difference between inline function and macro

I don't even require it in upp, I can write shortcut into my project. As you say, it is not very common case. I just wanted to note that such shortcut might be handy and to note the problem and solution here in case someone else ever runs into this situation in the future.

Honza