
Subject: Witz internationalization

Posted by [Zbych](#) on Fri, 29 Mar 2013 18:13:26 GMT

[View Forum Message](#) <> [Reply to Message](#)

Hi,

I was playing little bit with translating texts in witz scripts and it appears that it is possible to use `t_("xxx")` syntax like in *.cpp files (e.x. `$t_("some text")`).

TheIDE even scans *.witz scripts for `t_("")`. So far so good. But I have some questions:

- GetLngString uses cache to speed up translation. Is it possible that different texts from *.witz can have the same location in memory? Or maybe the same text can have different address each time and this will make cache grow after each translation. Maybe it is better to use non-cached version (`GetLngString(GetCurrentLanguage(), (String)arg[0])`)

- is it possible to detect user agent language and automatically show him proper translation?

Modified Skylark11 example:

main.cpp:

```
#include <Skylark/Skylark.h>
#include <plugin/sqlite3/Sqlite3.h>
```

```
#define TFILE <Skylark11/lang.t>
#include <Core/t.h>
```

```
using namespace Upp;
```

```
SKYLARK(HomePage, "")
{
    http("VAR", t_("Aborted by user.")).
        .RenderResult("Skylark11/index");
}
```

```
SKYLARK(SetLanguage, "setlanguage/*")
{
    int lang = LNGFromText(http[0]);
    if(lang)
        http.SetLanguage(lang);
    http.Redirect(HomePage);
}
```

```
struct MyApp : SkylarkApp {
    MyApp() {
        root = "myapp";
        threads = 1; // Sqlite3 does not like threads...
#ifdef _DEBUG
```

```

    prefork = 0;
    use_caching = false;
#endif
}
};

```

```

Value translate(const Vector<Value>& arg, const Renderer *)
{
    if(arg.GetCount() == 1 && IsString(arg[0]))
        return GetLngString((String)arg[0]);
    // return GetLngString(GetCurrentLanguage(), (String)arg[0]);
    return arg[0];
}

```

```

INITBLOCK {
    Compiler::Register("t_", translate);
};

```

```

CONSOLE_APP_MAIN
{
#ifdef _DEBUG
    StdLogSetup(LOG_FILE|LOG_COUT);
    Ini::skylark_log = true;
#endif
    MyApp().Run();
}

```

index.witz:

```

<html>
<body>
<a href=$SetLanguage("en-us")>EN</a>
<a href=$SetLanguage("cs-cz")>CZ</a>
<a href=$SetLanguage("pl-pl")>PL</a>
<br>
$t_("Current language"): $.language<br>
$t_("English version")<br>
$t_("Variable value"): $VAR
</body>
</html>

```

lang.t:

```

#ifdef _MSC_VER
#pragma setlocale("C")

```

```
#endif
// lang.witz

T_("Current language")
```

```
// index.witz
```

```
T_("English version")
pIPL("Wersja polska")
```

```
T_("Variable value")
```

```
// main.cpp
```

```
T_("Aborted by user.")
```

Subject: Re: Witz internationalization
Posted by [dolik.rce](#) on Sat, 30 Mar 2013 09:16:01 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hi Zbych,

I wrote almost exactly the same function a while back
Quote:Value Translate(const Vector<Value>& arg, const Renderer *) {
if(arg.GetCount() != 1 || !IsString(arg[0]))
return String();
return GetLngString(String(arg[0]));
}

```
INITBLOCK {  
Compiler::Register("t_", Translate);  
}
```

I had no problem with this implementation, but I haven't use it too extensively, so I probably wouldn't notice cache problems. Also, it is only useful for shorter strings, anything bigger is better to be placed in separate files (*.xy-xy.witz).

User language can be detected by parsing the Accept-Language HTTP header. However, it is always good idea to let the user override the setting manually (e.g. for cases when they use someone else's computer with different language settings).

Best regards,

Subject: Re: Witz internationalization

Posted by [mirek](#) on Sat, 30 Mar 2013 10:24:43 GMT

[View Forum Message](#) <> [Reply to Message](#)

- there definitely IS a problem with GetLngString and cache. t_ macro and GetLngString are only defined when being used with string literal. You are using them with String...

- to get HTTP header fields, use Http::GetHeader

Mirek

Subject: Re: Witz internationalization

Posted by [Zbych](#) on Sat, 30 Mar 2013 11:22:53 GMT

[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Sat, 30 March 2013 11:24- there definitely IS a problem with GetLngString and cache. t_ macro and GetLngString are only defined when being used with string literal. You are using them with String...

As you can see insane people like me are always trying push your code to the limits (and even beyond). Maybe there should be overloaded version of GetLngString that takes String and uses hash instead of address to cache translations? I can make it.

mirek wrote on Sat, 30 March 2013 11:24

- to get HTTP header fields, use Http::GetHeader

dolik.rce wrote on Sat, 30 March 2013 10:16

User language can be detected by parsing the Accept-Language HTTP header.

Thank you both. I will try it.
