
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.")
```