Subject: Re: Helper for internazionalize arrays of literals
Posted by mirek on Sun, 10 Jan 2010 13:40:54 GMT
View Forum Message <> Reply to Message

[quote title=luzr wrote on Sun, 10 January 2010 08:26]mdelfede wrote on Sat, 09 January 2010 15:39This one corrects a subtle bug and delays loading of strings (needed to setup properly the language) :

```
class StringTable : public Vector<String>
{
 public:
  StringTable &operator,(const char *s) { Add(s); return *this; }
  String operator[](int i) { ASSERT(i < GetCount()); return GetLngString(At(i)); }
};

#define STRINGTABLE(s) StringTable s; INITBLOCK { s,
#define ENDTABLE ; }
```

Stringtable must be defined with tt_ :

```
STRINGTABLE(Test)
    tt_("One"),
    tt_("Two"),
    tt_("Three")
ENDTABLE;
```

The tt_ macros are needed just to make theide export the translation; can be avoided if no need to sync translations.

Ciao

Max

Still not sure whether all of this is worth the trouble....

BTW, there IMO could be even more effective solutions. E.g.:

```
#include <Core/Core.h>

using namespace Upp;

struct tt_char {
 const char *s;
```

```
 String ToString() const       { return GetLngString(s); }
 operator const char *() const { return ToString(); }
};

CONSOLE_APP_MAIN
{
 static tt_char x[] = {
  tt_("Aborted by user."),
  tt_("Two"),
  tt_("Three")
 };

 SetLanguage(LNG_('I','T','I','T'));
 DDUMP(x[0]);
}
```