

В то время, как ресурсные файлы позволяют хранить больше, чем код программы в EXE файле, включением ресурсов **StringTable** в приложение, разработчик Delphi может легко проектировать многоязыковые приложения. Давайте научимся, как это делается.

Ресурсы StringTable

В этой статье Вы научитесь проектировать приложение, которое поддерживает большое количество строковых ресурсов и может быть переведено на другой язык с минимальными усилиями. Сохранив локализованные заголовки меток, кнопок и других управлений в ресурсах **StringTable** мы получаем много преимуществ:

1. Символьные строки, содержащиеся в **StringTable** не занимают память, пока они не будут определены загружены приложением.

2. При использовании ресурсов **StringTable** при создании многоязычных приложений, добавления нового языка является хорошим тоном, так как **StringTable** могут быть легко отредактированы. Более того, если **StringTable** хранится в ресурсном **DLL**, Вам даже не нужно будет повторно компилировать приложение.

Как и любой другой тип ресурсов, ресурсы **StringTable** компилируются в .RES файл, который прилагается к EXE файлу Вашего приложения во время компиляции.

Создание StringTable ресурсов

Для создания ресурса **StringTable** приложения для двух языков:

1. Создайте текстовый **.RC** файл, который содержит Ваши строковые ресурсы в директории Вашего проекта. Назовите файл *StringTableLang*
uage.rc.
Вообще-то Вы можете назвать его по-любому, главное, чтобы расширение файла было **.RC**

```
StringTableLanguage.rc  STRINGTABLE { 1000, "English" 1001, "Display selected"
1002, "Yes" 1003, "No" 1004, "Maybe" 2000, "Русский" 2001, "Выбор отображения"
2002, "Да" 2003, "Нет" 2004, "Возможно" }
```

2. Компилируйте этот **RC** файл в **RES** файл при помощи компилятора ресурсов **BRCC32**.

Обратите внимание: Файл *StringTableLanguage.rc* может содержать любое дополнительное количество ресурсов другого типа (иконки, изображения, данные и т.д.)

Таблица строк начинается с ключевого слова **StringTable**. Строки заключены в фигурные скобки. Каждой строке присвоен числовой идентификатор. Символьные строки заключены в кавычки. Если Вы хотите использовать нестандартный символ, вставьте обратный слеш и далее номер символа, который Вы хотите вставить. Единственное ограничение: когда Вам нужно будет вставить обратный слеш, Вам необходимо будет вставить двойной обратный слеш.

Например:

```
1, "A two-line string" 2, "c:\Borland\Delphi" Использование таблиц строк
```

Чтобы загрузить определенную строку из **StringTable** нужно использовать функцию **LoadString**

. Один из параметров в вызове

LoadString

- индекс строки в таблице строк.

Связывание в приложении

Как и любой **.RES** файл, Вы можете связать файл ресурсов с Вашим приложением просто добавив следующую инструкцию в код Вашего приложения (после

implementation

).

```
{ $R StringTableLanguage.RES }
```

Как только файл ресурсов будет связан с Вашей программой, Вы сможете загружать ресурс из любого модуля, даже если Вы определили директиву **\$R** в другом модуле.

LoadString

Вот пример, как использовать функцию **LoadString** для загрузки строки из **StringTable**:

```
function GetString(const Index: integer) : string; var  buffer : array[0..255] of char;  Is :  
integer; begin  Result := "";  Is := LoadString(hInstance,          Index,  
buffer,          sizeof(buffer));  if Is < 0 then Result := buffer; end;
```