

# Проект RusXMMS: Прозрачная работа с кодировками

Сурен Чилингарян  
Karlsruhe, Germany  
Forschungszentrum Karlsruhe  
RusXMMS  
<http://RusXMMS.sf.net>

## Аннотация

Наверное, каждому русскоязычному пользователю знакомы проблемы с кодировками: ID3 заголовки, ZIP архивы, FTP сервера и ICQ – вот неполный список задач, в которых часто можно увидеть абракадабру вместо понятного текста. Проект RusXMMS призван помочь программисту сделать в его приложениях работу с кодировками прозрачной для пользователей. В настоящий момент в рамках проекта разработаны две библиотеки: библиотека автоматического определения кодировки и библиотека перекодировки. Библиотеки оптимизированы для работы с многоязычными списками, помимо перекодировки предоставляют возможность транслитерации и, в случае наличия широкополосного доступа к интернет, перевода. В докладе будут коротко освещены основные возможности библиотек проекта, а также будет рассказано, как с их помощью можно быстро добавить поддержку перекодировки в существующие приложения.

Не смотря на широкое распространение юникода, разработчики приложений до сих пор вынуждены сталкиваться с большим количеством разнообразных кодировок, используемых пользователями на различных платформах. Проект RusXMMS призван сделать работу с кодировками в многоязычной среде прозрачной для пользователей и программистов. Используя функциональность, предоставляемую библиотекой, возможно с помощью нескольких строчек кода добавить в ваше приложение поддержку автоматического определения и конвертации кодировок, а также, при желании пользователя, транслитерации и перевода. Нижеприведенные рисунки иллюстрируют возможности библиотеки на примере музыкального проигрывателя XMMS.



*Рисунок 1: Пример использования библиотеки в проигрывателях музыки. Сверху оригинальный XMMS, а снизу XMMS, использующий библиотеку перекодировки.*

PLAYLIST EDITOR	
1. Океан Ельзи - Я на небе был	3:16
2. Carl Michael Bellman - sows trot we gradually	4:49
3. Богусhevская Ирина - У Нас В Раю	4:03
4. Океан Ельзи - Когда тебя нет	23:49
5. Отбор Yann - когда пастух	19:22
6. Богусhevская Ирина - Наперегонки С Волной	4:29
7. Wu Li-lan layers of water	10:35
8. Wu Li-JEWELRY	4:58
9. Udo Jürgens - Париж, просто так только к шутке	3:53

Рисунок 2: Пример использования функции перевода.

Итак, текущая версия библиотеки поддерживает:

- Перекодировку заголовков, причем кодировки можно как жёстко задавать, так и использовать авто-определение. Встроенная поддержка авто-определения имеется для русского и украинского языков. Восточно-европейские языки поддерживаются при помощи библиотеки Encs. Китайский, Японский и Корейский при помощи – LibGUESS.
- Многоязычные списки, а так же авто-определение языков при помощи библиотеки Aspell.
- Транслитерацию и перевод при помощи библиотеки LibTranslate, которая в свою очередь использует онлайн переводчики. В частности, для перевода на русский и украинский языки используются Translate.ru и Pereklad.online.ua.
- Кэш перекодировок на основе базы данных BerkeleyDB, что позволяет существенно оптимизировать производительность в случае использования функции перевода.
- Конфигурационный файл, позволяющий пользователям фиксировать определенные кодировки, включать режим перевода, настраивать режим кэширования перекодировок и т.д.
- Для GTK приложений доступен пользовательский интерфейс. Локализация интерфейса возможна через конфигурационный файл.

## Программный интерфейс библиотеки

Поскольку библиотека предназначена для работы в многоязычной среде, то архитектура предусматривает работу не с конкретными кодировками, а с классами кодировок. Так, к примеру, в проигрыватели XMMS эти классы включают кодировки ID3 заголовков версии 1 и 2, кодировку списка проигрывания, кодировку названий файлов в списке проигрывания, кодировку файловой системы, и кодировку отображения. Для каждого класса задается: имя, тип, способ получения кодировки по умолчанию и флаги. Тип класса и флаги, позволяют варьировать функции библиотеки относительно класса в зависимости от его назначения. Так, к примеру, для класса, который используется для представления данных пользователю, можно использовать функцию перевода, но ее использование не желательно для классов, которые сохраняют информацию для последующего использования. Для класса задающего кодировку файлов, можно использовать поиск по файловой системе для авто-определения кодировки. Или, если мы хотим сохранять ID3 v.2 заголовки всегда в UTF-16, то можно объявить этот класс константным и запретить его изменения из конфигурационного файла.

В качестве кодировки по умолчанию обычно используется первая сконфигурированная кодировка текущего языка. Однако, для некоторых классов можно указать, что по умолчанию должна использоваться указанная юникодная кодировка, кодировка, заданная указанной переменной локали, или кодировка заданного «родительского» класса.

Итак, для того, чтобы добавить поддержку библиотек RusXMMS в ваше приложение, для начала требуется описать список классов, который предполагается использовать. Простейший пример:

```
static rcc_class classes[] = {
    { "id3", RCC_CLASS_STANDARD, NULL, NULL, "ID3 Encoding", 0 },
    { "fs", RCC_CLASS_FS, "LC_CTYPE", NULL, "FileSystem Encoding", 0 },
    { "out", RCC_CLASS_TRANSLATE_LOCALE, "LC_CTYPE", NULL, NULL, 0 },
    { "utf8", RCC_CLASS_STANDARD, "UTF-8", NULL, NULL, RCC_CLASS_FLAG_CONST },
    { NULL }
};
```

Далее требуется инициализировать библиотеку, создать рабочий контекст, инициализировать кэш перекодировок, и загрузить конфигурационный файл.

```
rccInit();
ctx = rccCreateContext(NULL, 0, 0, classes, 0);
rccInitDb4(ctx, NULL, 0);
rccLoad(ctx, "xmms");
```

Если предполагается использовать пользовательский интерфейс, его также необходимо инициализировать:

```
uictx = rccUiCreateContext(ctx)
```

В конце работы необходимо сохранить изменения в конфигурации и освободить использованные ресурсы:

```
rccUiFreeContext(uictx);
rccLoad(ctx, "xmms");
rccFree();
```

После того, как библиотека инициализирована можно выполнять перекодировку между классами при помощи команды *rccRecode*. Более сложный сценарий предусматривает использование двух функций: *rccFrom*, которая выполняет перекодировку во внутренний формат библиотеки, содержащий информацию о языке и кодировке, и *rccTo*, которая конвертирует из этого формата в указанный класс. Пример:

```
recoded = rccSizedRecode(ctx, RCC_CLASS_FROM, RCC_CLASS_TO, string, stringlen)
```

Для того, чтобы добавить настройки RusXMMS в пользовательский интерфейс необходимо воспользоваться функцией *rccUiGetPage*, которая, в случае GTK интерфейса, возвращает указатель на *GtkVBox*. В обработчике, вызываемом при применении новой конфигурации, необходимо вызвать функцию *rccUiUpdate*, которая изменит текущую конфигурацию в соответствии с текущими пользовательскими настройками. Примеры вызова функций:

```
vbox = (GtkWidget*)rccUiGetPage(uictx, NULL);
rccUiUpdate(uictx);
```