

# Multilingual Flash Applications

Viswanath Parameswaran  
FastCurve Pte. Ltd.  
vnath@pacific.net.sg

## Abstract

Flash has simplified deployment of content on multiple platforms and browsers. Now with the ability to handle Unicode, it takes web applications into the next level. We will be looking at ways for creating multilingual content and a sample multilingual application using FlashMX.

**CR Categories:** H.3.5 Online Information Services, H.5.1 Multimedia Information Systems, K.8.1 Application Packages

**Keywords:** Web-based services, Graphics

## 1 Introduction

Currently Character encoding schemes, Browsers and Platforms abound. For the application developer this becomes a nightmare. A single delivery medium, which is cross platform and cross browser with support for Unicode is what Flash MX offers.

Multilingual capabilities of Flash boil down to two important aspects of manipulating text strings within Flash. The ability to bring in and to display different Character Sets.

This happens in two distinct phases. First when the movie is being created in the authoring environment and second when it is packaged as an swf file. In the authoring environment Flash depends on the Operating System to map character sets and to embed corresponding font information.

Once an swf movie, text strings are encoded in UTF-8. The swf movie plays within a Flash 6 player that can store characters both in UTF-8 and UTF-16 format. This enables us to bring in Unicode text dynamically

## 2 Single Byte Content

Russian, Greek, Turkish etc can be displayed using Single Byte encoding schemes. These characters are found in the upper ASCII range of the ASCII character set.

Usually content is brought into Flash either by

1. Cutting & pasting from a softcopy or
2. Dynamically loading from an external source or
3. By directly keying in the characters.

In the case of characters from different languages it would mean, different character sets and fonts. Since the Flash application environment depends on the Operating System, first we will have to change the language selection for the Operating System, before we cut and paste the content. In windows it would mean changing the system locale to the corresponding language and in Mac it would mean choosing the appropriate language.

Once the text is brought into Flash, you have to use the corresponding language fonts to display correctly. For static text fields, these fonts will be embedded.

## 3 Double Byte Content

The writing systems of the Far East use ideographs instead of alphabets. Each ideograph corresponds to a word. Chinese, Korean and Japanese languages use a minimum of 3000-15000 such ideographs.

Single Byte Character sets are totally inadequate to represent them. This resulted in the creation of 16 bit or Double Byte Character Sets (DBCS).

Just as we brought in Single Byte characters, we can change the system locale or languages selected in the Operating System to Chinese, Japanese or Korean and bring in these characters.

You can choose Unicode fonts or fonts in the native encoding to display these characters. Usually most of the Asian language fonts have an International name and a localized name in the local language. So you would find these names instead of the international name, once you switch to the corresponding system locale.

## 4 Dynamic Text

Dynamic text in Flash allows us to display multiple languages simultaneously. The external source should be encoded in Unicode (UTF-8 or UTF-16). Since Flash doesn't embed the entire Unicode fonts, it is better not to embed any fonts while showing multiple languages.

## 5 Flash Applications

Let us look at a sample application using these multilingual features.

### 5.1 Multilingual Dictionary

A sample Illustrated dictionary (English-Russian-Chinese Dictionary) created using Flash.

## 6 Future

Most of the display issues are due to the inability to embed specific ranges of Unicode fonts from Flash. This makes it difficult to embed fonts for Double Byte Characters or multiple Single Byte Characters together.

We can be looking at coming versions of Flash as well as third party developers to solve these issues in the future. To be a robust platform for multilingual application development Flash still has to go a long way, but for now even the small step in FlashMX is a big stride in the right direction.