How can your website speak multiple languages to your customers?

Summary: *Looking at a wider reach for your website? Do you think that a site that talks to your customers in their language will speak in a more personal manner? Yes, indeed! Here we discuss three different ways in which you could organize your site to support multiple languages.*

Does your web site attract visitors from around the globe? Do your potential customers speak different languages? How can you communicate to them? One of the most obvious ways is to speak their language.
Many web sites need to be developed in several languages. Multi Language web site development is a specialized skill. This is because it throws up various issues like:
Speed - How can you provide a customized user experience without decreasing the speed of the web application?
Translation - How can you translate all the content of the web site into the different languages you would like to offer?
Font Support - How do you make sure that your visitor's browsers support the fonts you will use.
Cost - How can you achieve this objective without multiplying the cost of developing the web site by the number of languages offered?
Here we talk about the different options available in offering multi-lingual support on a website. This article aims to explain the methods in detail. The three methods discussed in this article are:

Dynamic Content Generation
Site Replication
Selective Replication

Let us now discuss these three methods, and also discuss the advantages and disadvantages of each one here.

METHOD 1

Dynamic Content Generation

Although this method is a very complicated way of organizing your site to support different languages, it could be an option if you have only two languages, or even three to support on a fast server. It is also a good idea to use this option only if your site is not huge.
In this method all the text of the site is stored in a database. Every page carries a variable (a session variable or a query string) to identify which language the site is to be displayed in. Based on that, the content is pulled out from the respective tables for the language chosen, and displayed.
You might now be wondering, what about Graphics? You have two choices. If the amount of graphics that your site uses is very minimal, you could consider storing them in the database itself as blob fields. Another way is to simply open up a new table with the following structure:
Name English German French
Stored in this manner, you could give each image a name, and store only the relative paths to the different images in the database. When pulling it onto the client page, get the path and pull it out from the file system.

The messages can be stored in the database in a similar format, except instead of "Name" use a unique ID for each message. This message can then be called in the necessary pages of the site. You could also declare an array which you include in all pages, that contains all the messages. Please take care to keep the message number's constant once assigned because if the messages reshuffle it could be a tedious task to re-do all the messages on the site.
This method has many disadvantages. A few significant ones are:
There could be a performance degradation of the site if the amount of content of the site is huge. Editing the site would require you to either directly edit the content in the tables, or alternatively provide an admin panel to edit the content of each page on the site!
The load on the database is too high which could lead to lower performance. As you can see, this method is good for small websites that have less content and graphics. Providing for a complete administration panel for the content is a big thing in itself, and the reliability can never be guaranteed.

METHOD 2

Site Replication

This is one of the most commonly used methods on the web. In this approach the main site, which is in the default language of the website, resides in the root folder of the site. This basically is how a website is when it's a single-language site. When you want a site in German you would replicate the entire site into a directory, say German. The links in the German site should refer to the corresponding pages on the German site only. Now typing www.mysite.com would give the site in the default language, but www.mysite.com/german would give the German version of the site. On every page of the site you would have a select box with language choices. All this box does is to re-direct the user to the same page that sits on the chosen language site.
Do use proper tools when replicating the site. If you were to do it manually you will have to edit each other files on the site and correct the links on them to point to the pages on the language site. If you use a tool like Dreamweaver, for example, this task will be done automatically.
Now there are a few pages where the select box cannot be placed. These are pages that utilize what are called hidden form fields, which carry form information from page to page. Passing these over to another page would be a problem unless you have a mechanism to detect all the form variables and redirect to the same page on the other language site with the variables passed in the query string.
This method has a disadvantage too:
Any bug that is cleared on the main site needs to be cleared in all the other language sites. If you have 3 languages that you support, apart from the default language then this would increase work involved in any maintenance/bug-fixing/content-changing task 3-fold. You would have to make the change in the main language site, and then the change in each of the 3 other sites.
This does have a work-around. In your initial design of the site if you take care of code/content to be re-usable, this would not be an issue. All the language sites use the same includes so if there is any change in functionality all you would have to do is change the include file.

METHOD 3

Selective Replication

Of the three methods we discuss in this article, this is the most efficient one. Although difficult to set up the first time, the maintenance effort is lower than the other two methods discussed. This method is used by many major websites, including Microsoft, for multi-language support.
In Selective Replication we have the main site, which has no content or images whatsoever. The various images sit in various folders marked EN, GR, ES, etc depending on the languages. All the files that go into each of these directories have the same names. So, the English logo file name will be logo.gif, and so will the logo file for the other languages too.
The content (messages, JavaScript alerts, etc) have two places in which they can be stored. One way is to store each individual message as separate text files, or an alternative way is to make them sit in an array which is included in every ASP file and the message that needs to appear is called from the array. Each language has a separate array which resides in its directory. So the array include depends on the language that is chosen by the user. For information on how to create/use an array for this purpose, please see my article on "efficient use of arrays".
This method has no stress on the database. The database is designed to hold generic information applicable for all the languages.
The problem in this approach arrives only when the site is re-designed, the template changed and the content reworked. You will then have to re-create all the files in the language directories and change all the calls in the site files to include the newly created template files. Using text files for storing major content and storing all one/two line messages in an array or database tables could significantly drop time in maintenance of simple content changes.

CONCLUSION

In all the three steps discussed, bear in mind that the database needs to be able to handle Unicode characters. German characters like the §, etc need Unicode support to show up. By default, Windows installs with the Western European (ISO) encoding standard which supports all Unicode characters.
It would be a good idea to keep re-usability as priority one when designing the site. The more code/graphics/content you can make reusable for all the sites, the lesser the headache for maintenance and bug-fixing. Click here if you would like to see my article on Reusability.
You might also want to mix features of these three methods and derive a method suitable to your site. For example, you could use Selective Replication for the graphics and files and store all the content in the database using the Dynamic Content Generation method.