# 9
# Localization

You may be wondering why a chapter on localization would be included in a book on ASP.NET architecture and design. As explained earlier, the primary goal of this book is to make the developer familiar with all of the major aspects—from architecture and design to implementation—so that he or she can develop and design medium to large-scale web applications. There are many tasks that go into the making of a scalable multilingual ASP.NET web application. Making sure that the application is capable of supporting different cultures and locales is one of them.

The fact is that most web applications do not focus only on users belonging to a specific region but cater to users spread across various countries and continents. It makes more business sense to launch applications in multiple countries and target users around the world. For this to be possible, we have to make sure that our application is usable by people from different countries and cultures, as not everyone is a native English speaker. This is why we need to globalize such applications and make our content multilingual and culture-specific. This chapter will make the book more complete and rounded so that a developer can handle basic architecture and design issues while globalizing his or her ASP.NET web application.

In this chapter, we will learn:

- What globalization and localization is
- The best practices to follow while implementing globalization
- What resource files are
- How resource files are published
- Using the custom resource provider model for database based localization

In short, this chapter will explain all of the important concepts regarding globalization and localization both practically and comprehensively, and provide a practical step-by-step approach to globalizing a web application in ASP.NET 3.5 by following certain best practices.

# Globalization and Localization

**Globalization** is defined as the process of developing a program or an application so that it is usable across multiple cultures and regions, irrespective of the language and regional differences. For example, say you have made a small inventory management program and you live in a region where English is the main language, such as the USA. Now, if you want to sell your program in a different country, say Germany, then you need to make sure that your program displays and accepts input in the German language as well, so that German users are able to use the same application.

**Localization** is the process of creating content, input, and output data, in a region-specific culture and language. Culture will determine date display format (such as, mm/dd/yyyy or dd/mm/yyyy), currency display format, and so on. The process by which we can make sure that our program will be localized is known as Internationalization or Globalization. In simpler terms, Globalization can be defined as the set of activities that will ensure that our program will be usable in regions with different languages and cultures. So globalization is the overall process whereby we change the code to support localized data, for example by using Resource files, and so on. Localization, on the other hand, is the process of using a particular culture and regional information so that the program uses the specific local languages and culture. This means translating text strings into a particular local language. This involves putting language-specific strings in the resource files. Globalization starts in the main construction phase, along with code development. Localization generally comes later.