

Building Localized Applications with Microsoft .NET Framework and Visual Studio .NET

Achim Ruopp
International Program Manager
Microsoft Corp.

Agenda

- Definitions
- Resources:
 - Creating
 - Building
 - Using
- Localizing Windows Forms and strings
- Localizing ASP.NET applications

Localizability vs. Localization

- Localizability
 - The ability of a product and/or content (including text and non-text elements) to be adapted for any local market (locale).
- Localization
 - The process of adapting a product and/or content (including text and non-text elements) to meet the language, cultural, and political expectations and/or requirements of a specific local market (locale).

CultureInfo

- Neutral culture
 - Based on language
 - Resource only
 - No formatting
 - CurrentUICulture only
- Specific culture
 - Based on language & region
 - Resource & Formatting specifics
 - CurrentCulture & CurrentUICulture

Resource Model

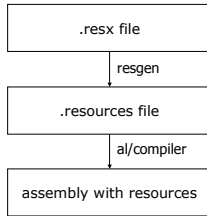
- .NET Framework uses a new resource model
- any serializable object can be a resource (e.g. also sound, images)
- resource model is extensible to new formats
- localization focuses on text, Windows Forms formats
- fully supported in the .NET Compact Framework for smart devices

Resource source formats

- text format (.txt)
 - simple name/value pairs
 - only suitable for string resources
- ResX XML format (.resx)
 - Simple, (almost) human-readable XML format
 - Can include arbitrary objects
 - Can be created with Visual Studio
 - Some samples in the .NET Framework SDK

Resource generation process

- .resx file
 - XML-based file
 - Describes resources
- .resources file
 - binary compiled file
- assembly
 - executable with default resources
 - resource-only satellite assembly (.resources.dll)



Naming convention and layout

- Naming pattern conventions
 - .resources files:
 - <myproject>.<xx-XX>.resources
 - Satellite assemblies
 - <myproject>.resources.dll
- Directory locations for satellite assemblies:
 - A subdirectory per culture
 - either neutral or specific cultures can be used
 - Myproject.dll
 - %fr%\myproject.resources.dll (neutral)
 - %de-DE%\myproject.resources.dll (specific)

Naming convention and layout

continued

- Installing satellite assemblies into the global assembly cache (GAC)
 - assemblies need to be strong-named (SN tool)
 - a strong name consists of a name, version number and culture combined with a digital signature
- Servicing the main assembly without revising the satellite assemblies – SatelliteContractVersionAttribute

System.Resources namespace

- ResourceManager
 - Provides access to culture-correct resources at runtime.
- ResourceWriter
 - Writes resources to an output stream or file
- ResourceReader
 - Reads resource name-value pairs from resources files and streams
- ResourceSet
 - Stores all resources localized for a particular culture

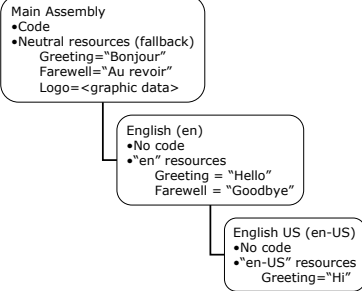
Loading Resources

- Creating a Resource Manager
 - The resource manager constructor indicates the file from which resources are to be loaded.
 - Several alternatives
 - Load from a loose .resources file
 - Load from this assembly
 - Load from another assembly
 - Load from a custom resource format if you write your own resource manager (e.g. a database)
 - See .NET SDK, Samples and Tutorials, ASP.NET QuickStarts, and How Do I ... samples

Loading Resources, continued

- Resource manager can be used to load both strings and objects:
 - RM.GetString("string", new CultureInfo(en-NZ))
 - RM.GetString("string2")
 - RM.GetObject("Button1.Cursor")
- Loads requested resources based on Thread.CurrentThread.CurrentUICulture
- Fallback hierarchy derived from RFC 1766

Resource Fallback



Localizability support in the Visual Studio .NET IDE

- Windows Forms provide localization support:
 - every form has a **Localizable** property
 - **Localizable** property set to **True**:
 - the project system automatically keeps track of different language versions of a form
 - builds the different language forms into satellite assemblies
- the resource format used for Windows Forms is the ResX format
- ResX resource template
 - can be used for string resources
 - is supported by the project system in the development environment

External localization process

- any XML localization tool can be used
- WinRes: tool for visual editing of Windows Forms
 - contained in .NET Framework SDK
 - does not require access to source
- 3rd party localization tools enabled for the new resource format:
 - ForeignDesk <http://www.foreigndesk.net> (Lionbridge)
 - Alchemy CATALYST <http://www.alchemysoftware.ie/>

Demo

- Creating a localized Windows Forms application with Visual Studio

ASP.NET And Localization

- ASP.NET has a flexible resource model
 - There is not "only one way" to organize localizable resources
 - Applications vary
 - Developers can choose
 - The following section shows a few alternatives
 - .NET Framework SDK has additional samples
 - ASP.NET QuickStarts, localization section, working with resources files.

ASP.NET Localization Options

- A: localizable resources included in source (like ASP)
- Copy of each page per language
 - Specify culture in page directive or web.config
 - Advantages:
 - Good for static content
 - Rapid initial development
 - Disadvantages:
 - High maintenance costs
 - Not suitable for dynamic content
 - Localizers and developers are working on same file.
- Not recommended

ASP.NET Localization Options

continued

B: Resources separated from source

- Use Resource Manager to load resources
- Resources could be in
 - Loose .resources files
 - Satellites below each main assembly
 - A separate (parallel) main assembly with satellites below it
- One .resources file or satellite per language
- Slightly different ResourceManager constructors for each case

ASP.NET Localization Options

continued

B: (continued)

- Advantages:
 - Can deploy additional languages incrementally, without redeploying core code
 - Lower maintenance costs, as there is one central app, not many parallel versions.
 - Disadvantages:
 - More files
 - Loose .resources can have locking issue.
 - No locking issue for satellites or parallel main assembly with satellites.
 - Shadow-copying done for EXEs and DLLs (main assemblies and satellites).
 - Samples: ASP.NET Quickstarts, Localization section
- Recommended: use satellite DLLs

Demo

- ASP.NET localized application

Summary

- .NET Framework provides built-in support for localization (and globalization)
- Visual Studio makes it easier to build localized applications
- ASP.NET builds on Framework support for international app development
 - Can separate code from resources
 - Can utilize .NET Frameworks classes
 - Developers can develop/host multi-cultural apps on a single server

References

- .NET Framework SDK:
 - Developing World-Ready Applications
 - Samples
 - SDK Tools
 - <http://msdn.microsoft.com/net>
 - <http://www.GotDotNet.com>
- Visual Studio .NET
 - Visual Studio .NET Visual Basic and Visual C# Globalizing and Localizing
 - <http://msdn.microsoft.com/vstudio/>
- Culture identifier syntax (RFC 1766)
<http://www.ietf.org/rfc/rfc1766.txt>

Questions?