Building Windows Applications with .NET

Allan Laframboise
Shelly Gill

Introduction

- Who are we?
- Who are you?
- What is your experience…
  - Developing with ArcGIS Desktop, Engine and Server
  - ArcGIS 8.x, 9.x and 9.2 beta
  - Programming VBA/VB 6.0, VC++, .NET and Java
  - C# vs VB.NET

Overview

- ArcGIS and .NET development
- Visual Studio .NET IDE integration
- How to build ArcGIS Desktop, Engine and Server application
- Version compatibility and migration
- .NET development tips

Why develop ArcGIS and .NET?

- Visual Studio development experience
  - Fast and easy to build applications
  - Unified IDE
  - Supports different languages (VC++, VB.NET, C#...)
- .NET API and framework is very rich
  - Desktop components, web services, mobile APIs…
  - Easier to program using managed code
- Standards-based architecture
  - XML, SOAP…
- .NET runtime is freely distributable!

Why develop ArcGIS and .NET?

- Provides full COM support
  - Program with any of the ArcObjects objects and interfaces
- Languages are object oriented
  - Easy to extend any given ArcObjects component
- Easy to build n-tier ArcGIS applications
  - Desktop components
  - Engine components
  - Web applications
  - Web services

ArcGIS and .NET support

- ArcGIS Desktop, Engine and Server install .NET assemblies
- Option to install .NET support during the installation
- Requires the .NET framework to be installed
  - ArcGIS 9.1 - .NET 1.1
  - ArcGIS 9.2 - .NET 2.0

ArcGIS .NET assemblies

- Most assemblies map to a COM type library
  - Provide interop layer between .NET and COM types
- Some assemblies are for ArcGIS Server
  - Not COM interop
  - Pure .NET implementations for ASP.NET
- Primary interop assemblies (PIA)
- Registered in the Global Assembly Cache (GAC)

ArcGIS and .NET IDE integration

- ArcGIS .NET SDK installs extensions to Visual Studio .NET 2003 and 2005
- What you can do:
  - Build ArcGIS solutions (applications)
  - Easily extend base classes
  - Build and register custom COM components
  - Implement ArcObjects interface
  - Implement licensing code in any application
  - Use add-ins for other common tasks
- Application plumbing is provided for you…

IDE integration extends Visual Studio

- New project templates
  - ArcGIS Desktop, Engine and Server
  - .NET and C#
- Project Wizard
  - Add references
  - Implement licensing
- New class templates
  - Add new ArcGIS Class – e.g. command, tool, layer and render
- Class Wizard
  - Implement interfaces
  - Select component categories
- Code snippets

Building Desktop applications with .NET

- Use the IDE integration to build a project
- Types of projects
  - Component based projects (dlls)
  - Stand-alone (exes)
- Most cases create a new class and implement an interface
- Use base classes where possible!

ArcGIS Desktop development example

- IDE integration in action

1. Build a project
2. Build a command
3. Build a custom layer
4. Debug and test in ArcMap

Building Engine applications with .NET

- Use the IDE integration to build a project
- Types of projects
  - Stand-alone (exes)
  - Component based projects (dlls)
- Most cases create a stand-alone 2D or 3D application
- Use controls and base classes where possible!

ArcGIS Engine development example

- IDE integration in action
  - Build a project
  - Use a previously built command
  - Use a previously built custom layer
  - Test and debug Engine application
  - Make changes
  - Save .MXD

Building Server applications with .NET

- Use the IDE integration to build a project
- Types of projects
  - ASP.NET web application
  - ASP.NET web service
- Most cases create a new web application
- Use templates, controls and convenience classes where possible!

ArcGIS Server development example

- IDE integration in action

1. Build a server object
2. Build a web project
3. Access data sources
4. Test and debug application in VS .NET

* Re-uses previously built .NET custom component

ArcGIS .NET development tips

- Version support
- Compatibility
- Migration

ArcGIS 9.2 and .NET support

- ArcGIS 9.2 is built on the .NET 2.0 framework
- Use VS 2005 for development
- Can not develop with VS 2003 and .NET 1.1
  - Format of the metadata in assemblies changed
  - Can not load .NET 2.0 framework

ArcGIS .NET version compatibility

- ArcGIS 9.2 policy files redirect 9.1 calls to 9.2
- Redirection is forward only
  - 9.1 -> 9.2
  - Sp1 -> sp2
- Must ensure end-users have a compatible version

ArcGIS 9.2 .NET changes

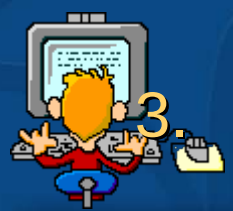
- Type libraries/assemblies
  - esriControlsCommands -> esriControls (all controls)
  - esri3DAnalyst -> esriAnimation
- ESRI.ArcGIS.Utility.dll has been deprecated
  - ESRI.ArcGIS.Utility.dll -> ESRI.ArcGIS.ADF.dll
  - CATIDs, BaseCommand and BaseTool have been duplicated
  - Only results in compiler warnings not errors
- New assemblies added
  - ESRI.ArcGIS.ADF.Web.xxx.dll

Forward compatibility

- How do you know if your application is forward compatible?

1. Check the migration documentation

2. Recompile your application on an ArcGIS 9.2 system with the .NET 2.0 framework
   - If there are no errors, the policy files will working
   - If there are errors, replace import statements and references…



3. Run the code converter in the IDE integration

Deploying ArcGIS 9.2 .NET applications

- Target machine requirements
  - .NET 2.0 runtime
  - ArcGIS 9.2 with *.NET support* option installed
  - Correct ArcGIS Service Pack (now distributable!)
- Use the VS setup project wizard
  - Reference existing project
  - Remove ESRI references
  - Add installer classes for category registration

Summary

- ArcGIS 9.2 Desktop, Engine and Server have strong support for .NET 2.0
- VS IDE integration will significantly improve ArcGIS application development and productivity
  - Project templates and wizards
  - New base classes…
- Policy files and code converter make forward deployment easier

Session Evaluations Reminder

Session Attendees:

Please turn in your session evaluations.

. . . Thank you