ArcGIS 10.1 for Desktop Developer Migration Topics

Deep Dhanasekaran and John Hauck
Agenda

• ArcGIS as a System
• Changes at ArcGIS 10 and 10.1
  - SDK Requirements
  - .NET 4.0 support
  - 64 bit support
  - Publisher Policy Files
  - Product name changes
• Runtime Binding
• Component Registration
• New Customization Framework Options
• New ArcGIS Runtime
ArcGIS 10 — A Complete System

Easier
More Powerful
and Everywhere

- Discover
- Create
- Manage
- Visualize
- Analyze
- Collaborate
## Product Name Changes

### Changes in the `esriProductCode` enumeration

<table>
<thead>
<tr>
<th>10.0</th>
<th>10.1</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>esriProductCodeArcView</code></td>
<td><code>esriProductCodeBasic</code></td>
</tr>
<tr>
<td><code>esriProductCodeArcEditor</code></td>
<td><code>esriProductCodeStandard</code></td>
</tr>
<tr>
<td><code>esriProductCodeArcInfo</code></td>
<td><code>esriProductCodeAdvanced</code></td>
</tr>
</tbody>
</table>

### Changes in the `esriLicenseProductCode` enumeration

<table>
<thead>
<tr>
<th>10.0</th>
<th>10.1</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>esriLicenseProductCodeArcView</code></td>
<td><code>esriLicenseProductCodeBasic</code></td>
</tr>
<tr>
<td><code>esriLicenseProductCodeArcEditor</code></td>
<td><code>esriLicenseProductCodeStandard</code></td>
</tr>
<tr>
<td><code>esriLicenseProductCodeArcInfo</code></td>
<td><code>esriLicenseProductCodeAdvanced</code></td>
</tr>
</tbody>
</table>
ArcGIS for Desktop .NET SDK requirements

10.0

- Visual Studio 2008 sp1
- VS Express 2008
- Visual Studio 2010
- .NET Framework – 3.5 sp1
- ArcGIS Requirements
  - ArcGIS Engine Runtime
  - ArcObjects .NET SDK
  - ArcGIS Engine Developer Kit License
ArcGIS for Desktop .NET SDK requirements

10.1

- Visual Studio 2010
- Visual Studio 2010 Express
- .NET Framework – 3.5 sp1
- ArcGIS Requirements
  - ArcGIS Engine Runtime
  - ArcObjects SDK
  - ArcGIS Engine Developer Kit License
.NET Framework

- .NET Framework 3.5 is min requirement
  - Guaranteed to be installed with Engine or Desktop

- .NET Framework 4.0 is fully supported
  - Need to handle distribution
  - Desktop developer Add-ins install without Admin requirements

- Do NOT embed iterop types
64 Bit Support

- ArcGIS Desktop and Engine are 32 bit applications
  - Run as 32 bit applications on a 64 Bit OS
- For Standalone applications
  - Set platform to x86 in Visual Studio Configuration Manager
  - Default is “Any CPU”
- At Version 10 ArcGIS applications are Large Address Aware
  - On 64 Bit OS processes can take up to 4 gigabytes of RAM if available
No Publisher Policy Files at ArcGIS 10

- Policy files specify assembly redirection
  - Installed to the GAC
  - Included in previous versions of ArcGIS

- Desktop applications include an Application Configuration file
  - Redirects 9.3.x Assemblies to 10
  - There for migration assistance only
  - Recommended to recompile
Runtime Binding

What is Runtime Binding?

- At ArcGIS 10 each product has its own runtime
  - Products have separate install locations
  - Service pack products separately
  - Uninstall service packs
- ArcObjects must be pointed to a runtime to work
  - Before any other ArcObjects calls
  - Required for all standalone applications
  - Engine applications can bind to either Desktop or Engine Runtimes
- Remember – Binding is not Licensing
  - Still have to check out a license
Runtime Binding

How to bind to a runtime

- Add reference to:
  `ESRI.ArcGIS.Version`

- Bind using the `RuntimeManager` static class
  
  `ESRI.ArcGIS.RuntimeManager.Bind(ESRI.ArcGIS.ProductCode.Engine)`
Runtime Binding

When to bind to a runtime?

• **Before any other ArcObjects calls**
  - Preferably in your main method (C#)
  - Preferably in your application events StartUp method (VB)

• **Tip:**
  - The Assembly is called ESRI.ArcGIS.Version
  - The Namespace is ESRI.ArcGIS
  - Bind method returns a Boolean that you can use to handle binding errors
Runtime Binding
Binding and Deployment

- ArcGIS Engine is not required on the target machine
  - Engine applications work with either a Desktop or Engine Runtime

- Can mix and match Licensing and runtime for flexible Engine applications
  - Leverage Desktop license and runtime on clients machine
Component Registration
Component Registration

Understanding component registration

- Prior to 10, ArcGIS knew which customizations to load based on which categories a component was registered in.

- Example Custom ArcMap Command
  - Implement ICommand
  - Register COM component
  - Register command in MxCommands Category
  - ArcMap reads and loads all components from the MxCommands Category
Component Registration at ArcGIS 9.x

Understanding component registration

MxCommands

ArcMap.exe

My.dll

COM

ICommand
Component Registration

Understanding component registration at 10

• ArcGIS 10 no longer reads component categories from the registry

• Component Cat information is read from configuration files (*.ecfg)
Component Registration at ArcGIS 10

Understanding component registration at 10

ECFG
MxCommands
Registry
COM
ICommand
EsriRegasm
ArcMap.exe
My.dll
Component Registration

How to register custom components at 10.x

- Components are still COM based
  - Must register
- Use ESRIRegAsm.exe to register category information
  - Creates your *.ecfg file in the appropriate folder
    Engine:
    %CommonProgramFiles%\ArcGIS\Engine10.1\Configuration\CATID
    Desktop:
    %CommonProgramFiles%\ArcGIS\Desktop10.1\Configuration\CATID
Component Registration

Adding registration to existing projects

• Adding ESRIRegAsm.exe step to your .NET projects
  - Unload and Edit project
  - Add custom build step
  - This registers your component on build and un-registers on clean

• Visual C++ developers add custom Post-Build Event
  “$(CommonProgramFiles)\ArcGIS\bin\esriregasm.exe” /p:Desktop
  “$(TargetPath)”
Component Registration

How to deploy custom components

- To Deploy little change needs to be made
  - Option 1:
    - If you call RegSvr32.exe or RegAsm.exe
    - Call ESRIRegAsm.exe instead
  - Option 2:
    - Register like you always have with COM and copy a pre-generated *.ecfg file to the appropriate location

Engine:

%CommonProgramFiles%ArcGIS\Engine10.1\Configuration\CATID

Desktop:

%CommonProgramFiles%ArcGIS\Desktop10.1\Configuration\CATID
Migration

Tips for migration workflow

- ArcObjects SDK includes the Code Migration Analyzer
  - Provides warnings to assist with the migration process
- Update debug start action to new ArcGIS executable locations
- Fix any changes to Assemblies
- Add new component registration step
- Standalone applications add the Binding code
- Update Desktop License
Migrating Custom Components
New Customization Framework Options
New Customization Framework

Add-ins

• No COM registration required!
• Can use Java, .NET, or Python
• Create Add-in using a Wizard
  - Config.esriAddins.xml file and a class
  - New items can be created through wizard or through XML
• Compile Add-In
  - Creates an *.esriaddin file
  - Xcopy deployable
  - Installation is as simple as double-clicking on the Add-In
New Customization Framework

Add-ins

- Types of Add-In customizations
  - Buttons and Tools
  - Combo Boxes
  - Menus, Context menus. Multi-items
  - Toolbars and Tool palettes
  - Dockable windows
  - Extensions
  - Editor Extensions
New Customization Framework

Add-ins

• Add-ins are installed on a per user basis
• No administrative permissions required
  - Can be digitally signed
  - Installation can be controlled by systems administrator
    - Block all Add-ins, block unsigned Add-ins, specify administrator folder, only allow Esri Add-ins
• Add-In can be placed on a network share and referenced through Add-In Manager
  - All updates will automatically be picked up next time ArcGIS is loaded
Python Add-Ins

Enhanced Add-in framework to include Python
Demo Python Add-In
Python

Enhanced at ArcGIS 10.x

- Can be leveraged throughout the ArcGIS system
  - Same scripts can be run on
    - Desktop
    - Engine
    - Server
    - Windows/Linux
Python

Enhanced at ArcGIS 10

• ArcPy site package
  - Automation of GIS mapping functionality
    - Printing and Exporting
    - Manipulate Map Document and Layer files
    - Automate Map Book creation

• Automation of GIS analysis
  - Geoprocessing
  - Python API for Spatial Analyst
  - Python API for GeoStatistical Analyst
Python

Enhanced at ArcGIS 10.1

• ArcPy site package
  - Data Access Module (arcpy.da)
    - Edit Sessions, cursors, data conversion, versioning and replication workflows
  - Network Analyst module
    - Geoprocessing tools and helper classes and functions

• Automation of GIS analysis
  - Geoprocessing
  - Python API for Spatial Analyst
  - Python API for GeoStatistical Analyst
VBA Migration
VBA Migration

Status for 10.1

• Free VBA Compatibility Setup
  - Request License
  - No Support or SDK
  - Available to allow existing solutions to still work during migration

• Esri urges you to migrate VBA applications to a supported development environment

• Where should a VBA developer go?
VBA Migration

Why Migrate to Python?
- Simpler API
  - Perfect for Automation of Mapping and GIS analysis
- Functionality will increase over time
- Python can be leveraged from any development environment

Why might Python not be an option?
- Not ArcObjects replacement
  - Review ArcPy and make sure it does what you need
- Need to re-write code
  - Business logic will need to be changed significantly
VBA Migration

• Why Migrate to Add-ins?
  - A lot of code can be copy/pasted in a VB.NET application
  - Types of customizations are the same or similar
  - Simple deployment model
Migrating VBA to Add-ins
New ArcGIS Runtime

- New Architecture
- Easy to Deploy Solutions
  - Software, maps and data
- 64 bit Native Support
- Modern API
  - WPF and Java
- Reduced Memory and Disk Footprint
New ArcGIS Runtime

- Not a functional equivalent to Engine
- Not an ArcObjects replacement
- New APIs
  - Code will need to be re-written
  - Should leverage general patterns for data sharing and distribution
Questions?

• Please fill out the surveys.