# Deploying ArcGIS Desktop and Engine Applications in .NET

*Don Kemlage*

*Mita Patel*

# Schedule

- **Today we will cover**
  - **General Development Issues**
  - **General Deployment Issues**
  - **Demonstration**
    - **(Create a deployment package in Visual Studio)**

*Please complete the session survey!*

# Development and Deployment Cycle

**Development machine**

**Deployment machine**

**Visual Studio**

**ArcGIS Engine / ArcGIS Desktop**

**Your custom application**

**setup.exe / setup.msi**

**Your custom application**
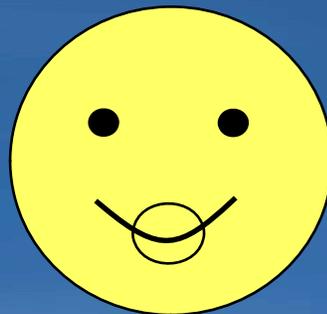
# General Development Issues

# Development Prerequisites

**ArcGIS Engine Applications**

- **ArcGIS Engine Runtime** (Engine, Engine with GeoDatabase Extensions)
- **ArcGIS Engine .NET SDK**

**ArcGIS Desktop Applications**

- **ArcGIS Desktop (ArcView, ArcEditor, ArcInfo)**
- **ArcGIS Desktop .NET SDK**

**Preferred**

- **ArcGIS Desktop**
- **ArcGIS Engine Runtime**
- **ArcGIS Desktop .NET SDK**
- **ArcGIS Engine .NET SDK**

# What do you get in the Engine and Desktop SDK's?

**ESRI's Visual Studio Integrated Development Environment Tools:**

- **ArcGIS Windows Forms Active X Controls**
- **Project and Item Templates**
- **ArcGIS Snippets (Snippet Finder and Snippet Editor)**
- **Add ArcGIS Reference Dialog**
- **Component Category Registrar Dialog**
- **License Initializer Dialog**
- **Toolbox Reference (GeoProcessing tools)**

**Developer resources**

- **HTML Help documentation**
- **Samples (C# and VB .NET)**
- **Object Model Diagrams**
- **Tools and Utilities (ESRI Object Browser, GUID Generator, etc.)**

**Engine Developers**

- **Rights to redistribute the Engine Runtime setup**

# ArcGIS Windows Forms Active X Controls
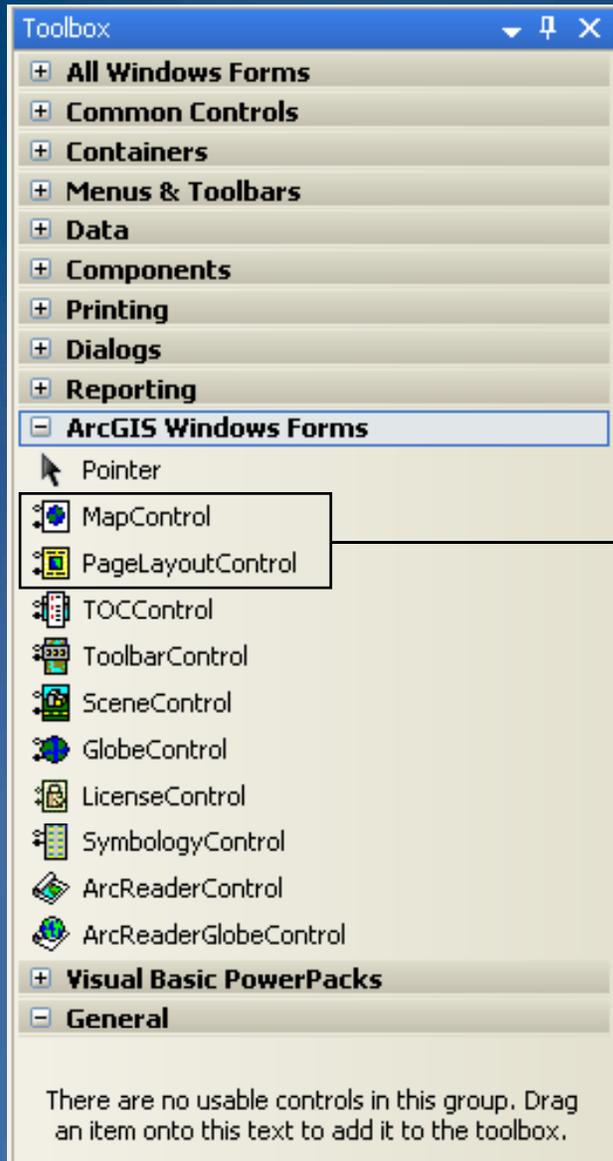


**Only controls available in Desktop**

# Application Types

- **An executable (.exe)**

.exe

- **A dynamic link library (.dll)**

.dll
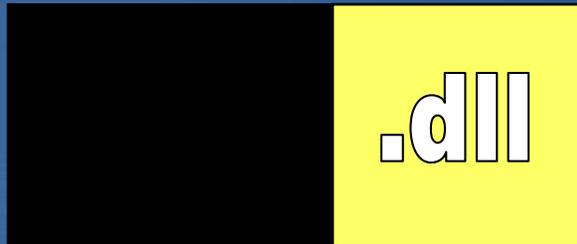
- **Hybrid: 1+ .exe's and 1+ .dll's**

.exe .dll .dll → .exe
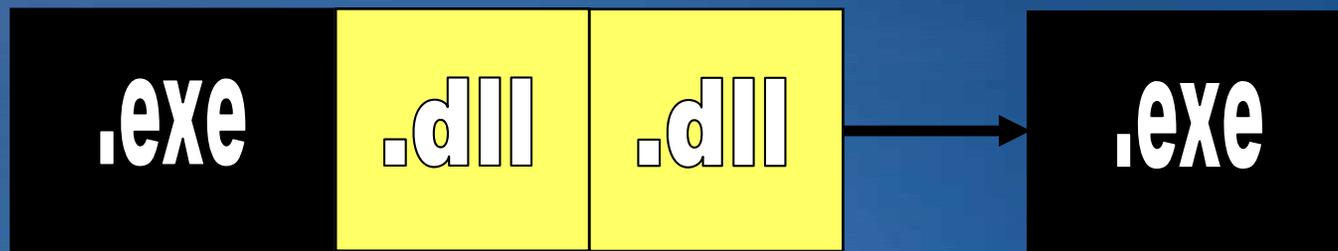
# Licensing Issues

- **Executables (.exe) require Copy Protection to ensure that the application is authorized**
  - ESRI uses .ecp files to unlock ArcObjects functionality. This is true for both Engine Single Use applications and Desktop Single Use and Concurrent Use (License Manager). Use ESRI's Software Authorization tool.

- **Dynamic Link Libraries (.dll) defer the applications authorization to the .exe**

- **Note: Once a deployment machine has been licensed with adequate ESRI Copy Protection, it can have an unlimited number of ArcObjects based applications installed on it for no additional ESRI royalties.**

**Licensing Issues cont.**

- As a developer of custom Engine and Desktop applications you CAN NOT redistribute the .ecp files.

- Exceptions: ESRI Business Partners with an Original Manufacturer Equipment (OEM) agreement and internal applications within an organization can unlock their applications functionality via code without providing an .ecp file and using the Software Authorization tool.

- Desktop and Engine licenses CAN NOT be used for server, Internet or Intranet applications. Use ArcGIS Server for these type of applications.

# Unlocking ArcObjects Authorization in Application .exe(s)

- **LicenseControl**

- **Use ESRI.ArGIS,System.AoInitialize**

- **Product codes available**
  - **Engine**
  - **EngineGeoDB**
  - **ArcView**
  - **ArcEditor**
  - **ArcInfo**



AoInitialize

IAoInitialize ○──── IAoInitialize : IUnknown

- CheckInExtension (in extensionCode: esriLicenseExtensionCode): esriLicenseStatus
- CheckOutExtension (in extensionCode: esriLicenseExtensionCode): esriLicenseStatus
- Initialize (in ProductCode: esriLicenseProductCode): esriLicenseStatus
- InitializedProduct: esriLicenseProductCode
- IsExtensionCheckedOut (in extensionCode: esriLicenseExtensionCode): Boolean
- IsExtensionCodeAvailable (in ProductCode: esriLicenseProductCode, in extensionCode: esriLicenseExtensionCode): esriLicenseStatus
- IsProductCodeAvailable (in ProductCode: esriLicenseProductCode): esriLicenseStatus
- Shutdown

```
[STAThread]
static void Main()
{
    //ESRI License Initializer generated code.
    m_pLicenseInitializer.InitializeApplication(new esriLicenseProductCode[]
        { esriLicenseProductCode.esriLicenseProductCodeEngine },
        new esriLicenseExtensi

    Application.Run(new frmMai

    //ESRI License Initializer
    //Do not make any call to                                    lication()
    m_pLicenseInitializer.ShutdownApplication();
}
```

- esriLicenseProductCodeArcEditor
- esriLicenseProductCodeArcInfo
- esriLicenseProductCodeArcServer
- esriLicenseProductCodeArcView
- esriLicenseProductCodeEngine
- esriLicenseProductCodeEngineGeoDB

# ArcObjects Authorization Rules

- **License initialization must occur before any ArcObjects code executes**

- **A product license can be checked out only once for the duration of the application**

- **Extension licenses can be checked in and out as necessary**

- **Cannot combine or mix product and extension licenses**
  - **ex. Engine Runtime must match Engine Extensions**

- **Tip: Always attempt to consume the lowest level of licensing possible**

# ArcObjects clean up for Application .exe(s)

- **You must clean up all ArcObjects**
  - Release memory
  - Release hooks to ESRI Assemblies

- **This will avoid the application crashing upon exit**

- **Use:**

**ESRI.ArcGIS.ADF.ComSupport.AOUnitialize.Shutdown()**
  - A good place is the Form's Dispose method

# General Deployment Issues

# What is a deployment?

- Files are copied onto a Target machine
- If necessary, Windows Registry values are added/updated

# What are the different deployment options?

- XCOPY and manual registration
- Batch files (.bat)
- Windows scripted
- Setup.msi and/or Setup.exe
  - Microsoft Windows Installer SDK
  - InstallShield
  - WiX Toolset
  - Wise for Windows Installer
  - NullSoft Scriptable Install System
  - Microsoft Visual Studio

# Setup .exe's -vs- .msi's

- .exe are just executables that can do anything you want

- .msi are Windows Installer files that do installations

- .exe's can call multiple .msi's (sometimes this is called a bootstrapper)

- An .msi can not call another .msi (it breaks the ability to use Add/Remove Programs)

# Deployment Scenarios

1. **End user will utilize ArcGIS Desktop**
   - Most likely .dll's
   - .exe's are possible but not common

2. **End user will utilize ArcGIS Engine Runtime**
   - Most likely an .exe
   - Possibly additional .dll's to enhance the .exe

# Deploying an application for ArcGIS Desktop (in most cases ArcGIS Desktop is already installed)

## Low Tech Way:

1. **Copy the .dll onto the deployment machine**
2. **Use regasm.exe to register the .dll**

   example: regasm "C:\test.dll" /codebase

   ms-help://ESRI.EDNv9.3/NET_Desktop/33903846-919f-48b4-a4b1-ef97680ddb7b.htm

## High Tech Way:

1. **Add an installer class to the VS class library project that uses System.Runtime.InteropServices.RegistrationServices**
2. **Create a Setup Project in VS to consume the .dll and installer class to create a setup.exe**

   ms-help://ESRI.EDNv9.3/NET_Desktop/0df20605-b457-42d6-b63c-341a3824474a.htm

**Note:**
**Add the ArcGIS customization to the GUI via the Customize dialog box**

# Deploying and ArcGIS Engine Application (usually ArcGIS Engine Runtime is not previously installed)

## Common Scenarios

#1  - User installs ArcGIS Engine Runtime (CD/web/network)

- User installs your custom application setup.msi

- User authorizes the ArcObjects functionality with an .ecp file

#2  - User installs you custom setup.exe (which contains the ArcGIS Engine Runtime .msi and your custom application)

- User authorizes the ArcObjects functionality with an .ecp file

#3  - User installs you custom setup.exe (which contains the ArcGIS Engine Runtime .msi, your custom application, and authorizes all ArcObjects functionality)

This option is only available to ESRI Business Partners with an OEM license agreement and organizations deploying internal applications

# Engine Runtime install options

- **Engine Runtime can be installed silently using msiexec commands**

    - **Basic UI Install**

        **msiexec.exe /i <setup location>\Setup.msi /qb**

    - **No UI Install**

        **msiexec.exe /i <setup location>\Setup.msi /qn**

    - **Installdir Parameter**

        **msiexec.exe /i <setup location>\Setup.msi /qn Installdir=c:\test**
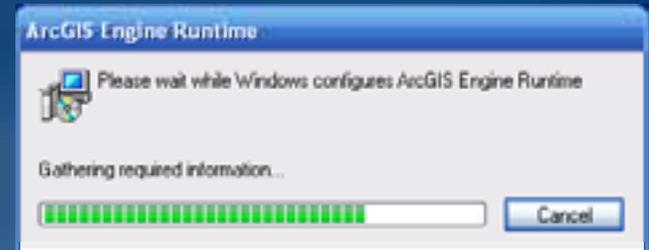
- **Help Documents:**

    **ms-help://ESRI.EDNv9.3/NET_Engine/88ddaf1f-7541-4ef9-a789-a05a726d1801.htm**
    **ms-help://ESRI.EDNv9.3/NET_Engine/fc7650e8-7cb8-43c8-84bc-f893b163b661.htm**
    **ms-help://ESRI.EDNv9.3/NET_Engine/81c23ac3-eeee-4be6-b11e-9b5829b83e59.htm**

# Service Packs

- **Application requires ArcGIS Engine Runtime and Service Pack**
  - Include Service pack file (msp) on CD

- **Launch within a batch file**

```
REM Set variables
SET MSI_PATH=<CDROM>\ArcEngine\Setup.msi
SET MSP_PATH=<CDROM>\ArcGISEngine92sp5.msp

REM Launch MSI Silently - NO UI
msiexec.exe /i "%MSI_PATH%" /qn
msiexec.exe /p "%MSP_PATH%" /qn
```

# Testing

- **Test on all supported platforms.**
- **Test on many configurations**
  - **Bare Operating System**
  - **Operating System, Office and Visual Studio**
- **Test Install**
  - **Test all launch conditions**
    - **Dependency on ArcGIS Engine Runtime 9.3**
  - **Diff the Installed files against source image**
  - **Test shortcuts**
- **Test Uninstall**
  - **Dll unregistration clean up**

# Resources

- **Platform SDK Documentation**
  - **http://www.microsoft.com/downloads/details.aspx?FamilyId=E15438AC-60BE-41BD-AA14-7F1E0F19CA0D&displaylang=en**
- **Appendix D of the Developer's Guide**
  - **Installing ArcGIS Engine Runtime on Windows, Solaris and Linux**
- **EDN website (http://edn.esri.com)**
- **Software Deployment websites**
  - **http://www.installsite.org/**
  - **http://msdn.microsoft.com/newsgroups/default.aspx?dg=microsoft.public.platformsdk.msi&lang=en&cr=US**

# Demonstration
## Create a deployment package in Visual Studio

# Authoring a Setup Program

1. **Create the setup template**

2. **Add the solution files to the template.**

3. **Launch Conditions**

4. **Prerequisites**

# The Setup Template

1. **Create the Setup IMAGE**

2. **Create the Setup Template**
   - **Product Information**
     **(Name, Version, ProductCode, Default Install location etc..)**

   - **Registry Keys**

   - **Features  (optional)**

# Add Solution files

1. **Add files**

2. **Register Dll's**

3. **Add shortcuts**

# Launch Conditions

- **Components required for the setup to run – such as .NET Framework, Engine Runtime, etc.**

- **Registry key's to check if Microsoft .NET Framework V2.0 is installed.**

  **HKLM\Software\Microsoft\.NETFramework\Policy\V2.01**
  - **50727**
- **Registry key to check if Engine Runtime is installed.**

  **HKLM\Software\ESRI\ArcGIS Engine Runtime**

  **Installdir**

  **RealVersion**

  **.NET_Support**

# Prerequisites

## Check for Pre-Requisites

- Installed Product, Features and Service Packs
- Registry keys
  - Engine Registry keys
    HKEY_LOCAL_MACHINE\Software\ESRI\ArcGIS Engine Runtime\RealVersion
    HKEY_LOCAL_MACHINE\SOFTWARE\ESRI\ArcGIS Engine Runtime\SPNumber

- Your setup.exe would install Engine Runtime if not found. (your Setup.msi would not run without it.)

# Demonstration

# Want to Learn More?
## *ESRI Training and Education Resources*

- ## Instructor-Led Training
  - **Developing Applications with ArcGIS Engine Using the Microsoft .NET Framework**
  - *http://www.esri.com/training*

# Additional Resources
## *Questions, answers and information…*

- ## *Tech Talk*
  - *Outside this room right now!*

- ## *ESRI Resource Centers*
  - PPTs, code and video

    **resources.esri.com**

# Session Attendees:

**Please turn in your session evaluations.**

*. . . Thank you*