Techniques to Use the 3ds Max .NET API enhancements

Kevin Vandecar

Principal Developer Consulting Engineer – M&E Autodesk Developer Network

GAME DEVELOPERS CONFERENCE 2012

SAN FRANCISCO, CA
MARCH 5-9, 2012
EXPO DATES: MARCH 7-9

bio: Kevin Vandecar

- @ Autodesk for over 18 years
  - all customization related
    - Experience with AutoCAD, AutoCAD Architecture, Revit
    - Five years as Software Engineer on AutoCAD Architecture API and Revit API
    - 18 months focusing on 3ds Max customization in ADN M&E Workgroup
  - Based in Manchester, New Hampshire, US
  - email: kevin.vandecar@autodesk.com

Autodesk Developer Network

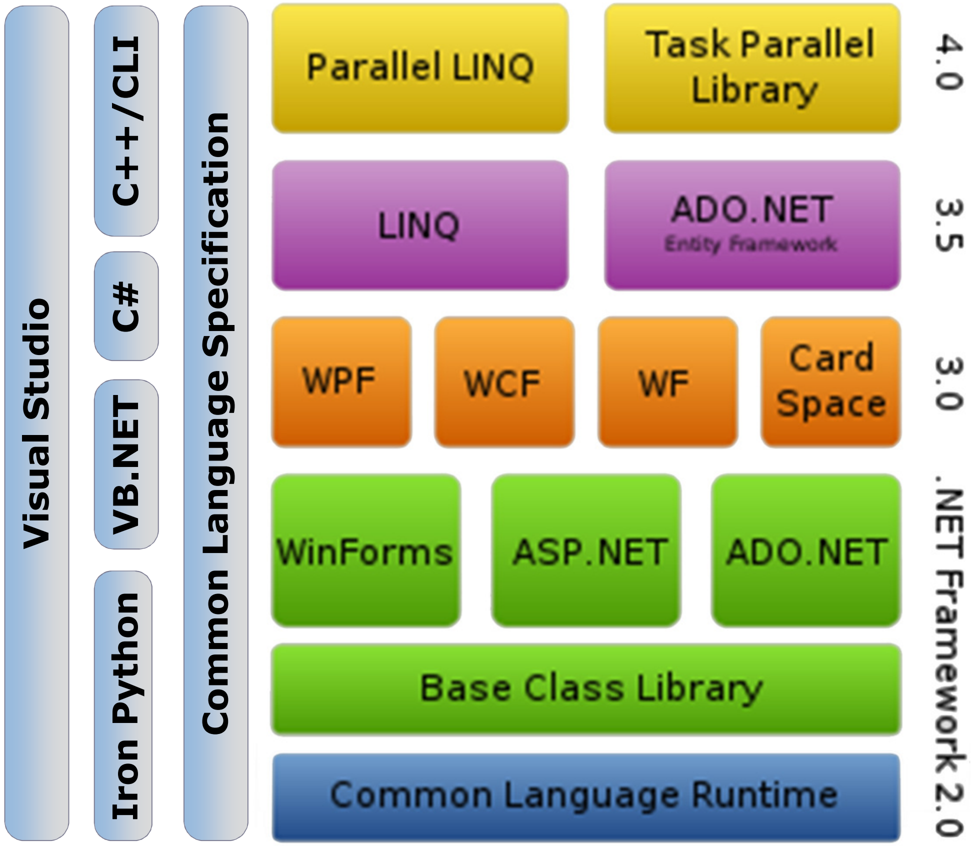
- Access to almost all Autodesk software and SDK's
  - Includes early access to beta software
- Members-only website with thousands of technical articles
- Unlimited technical support
- Product direction through conferences
- Marketing benefits
  - Exposure on autodesk.com
  - Promotional opportunities
- One to three free API training classes
  - Based on user level

www.autodesk.com/joinadn

What is .NET?

- Software architecture framework from Microsoft
  - Windows
  - Phone 7
  - Azure

- Programs execute in CLR
  - Programs are "managed"
  - Memory Management via Garbage Collection provided



Visual Studio

C++/CLI

C#

VB.NET

Iron Python

Common Language Specification

Parallel LINQ

Task Parallel Library

4.0

LINQ

ADO.NET Entity Framework

3.5

WPF

WCF

WF

Card Space

3.0

WinForms

ASP.NET

ADO.NET

Base Class Library

Common Language Runtime

.NET Framework 2.0

What is .NET?

- Power of C++ with simplicity of scripting
- Uses Simple Referencing
- Language choices using same APIs
- Reflection

What is .NET?

- Many modern Foundation Technologies
  - Windows Communication Foundation (WCF)
    - (ASP.NET, Azure, etc.)
  - Windows Presentation Foundation (WPF)
    - (UI Data Binding, Silverlight, View Model, etc.)
  - Windows Workflow Foundation (WF)

3ds Max .NET API

- .NET API capability has been part of the 3ds Max SDK for a number of years now, but mainly UI oriented
  - Supported through MAXScript
  - C++/CLI allows mixed-mode functionality

- New "Enhanced" version introduced in 2012
  - Subscription Advantage Pack
  - This new "enhanced" version is a new assembly called Autodesk.Max.dll
  - A 1-to-1 wrapper of the native C++ APIs

Getting Started - Learning

- The online Help for 3ds Max SDK now contains two distinct topics to help get started
    - "The Enhanced 3ds Max .NET SDK"
    - "Lesson 7: Writing .NET Plug-ins" – part of the new learning path documentation
    - See Christopher Diggins' blog for details

http://area.autodesk.com/blogs/chris/net_sdk_enhancements_in_the_3ds_max_2012_subscription_advantage_pack

Getting Started - Learning

- Use the C++ Reference and Programmers guide as main form of documentation.
- Keep in mind this is version ‘1’ ☺
- Use the Object Browser or Reflector
- Make sure to use the other assemblies as necessary, too.

Basics – Getting Started

- Create new managed “class library” project
- Reference and set the “Copy Local” property to **false** on all 3ds Max Assemblies
- Make sure to place (ie. build) the assembly to the <3dsMax>\bin\assemblies folder so it can be found and loaded
  - Note that 3ds Max will attempt to load any DLLs in this directory and it’s subdirectories

Two ways to be loaded/run

- Use Cui Action facility
  - easily hook into UI
- Assembly Loader
  - Allows loading and operations similar to Global Utility Plug-in (GUP)

Using Cui Action system

1. Derive from **`MaxCustomControls.CuiActionCommandAdapter`**

2. Implement necessary functions
   - Mostly strings to identify the action to 3ds Max
   - Implement **`Execute`** to run the code!

demo

Full Plug-in type using Autodesk.Max.DLL

1. Write a custom class derived from a plug-in base class in the Autodesk.Max.Plugins namespace
2. Write a custom class descriptor derived from Autodesk.Max.Plugins.ClassDesc2
3. Register the new class descriptor using the function IInterface.AddClass() from public static void AssemblyMain() of your assembly

demo

Using the “enhanced” functionality

- Autodesk.Max is the root namespace

- Use the IGlobal Interface, and Interface13

Using the “enhanced” functionality

- Functions with zero arguments and begin with “Get” are mapped to properties

- IGlobal contains Create method for types that can be instanced

Wrappers use C++ SDK techniques

- Class IDs are required to identify object types.
- Mixes unmanaged and managed ideas
    - For example IList is used to represent array

Caveats

- Nearly the entire C++ SDK is “wrapped”
  - however there are things that do not work exactly the same in .NET API
  - There are things that are not working
    - SystemNotificationCode.PostCloned notification works, but there is no way to marshal the INodeTab native type into managed code
    - The Texmap sample in the updated documentation does not work
    - Remember: version ‘1’ ☺

Caveats

- It is advised by engineering to NOT create a full plugin type.
  - This means you should not create a plugin derived from Autodesk.Max.Plugins.<types>
  - I have personally tested UtilityObj and GUP
    - However, using CuiCommandActionAdapter or AssemblyLoader is basically the same functionality.

Tips/Tricks

- Use Visual Studio tools to your advantage when learning.
    - Use Object Browser (or another tool like Reflector) to browse the assembly.
        - Object Browser has "custom" set that makes searching easy.
    - Use "Implement Abstract Class" to get stubbed methods.
    - Use refactor to rename elements

demo

Tips/Tricks

- When using CuiActionCommandAdapter
  - make a custom abstract base class that has your category and other requirements done
  - Then derive from it, and simply implement execute and ActionText

demo

Tips/Tricks

- Error check:
  - Try and Catch exceptions!
  - Check return values!
- Mix the older functionality with newer when appropriate
  - ManagedServices
  - CSharpUtilities

Some Examples

- Example 1
  - custom CuiActionCommandAdapter
  - IGlobal and IInterface
- Example 2
  - Host WPF window
  - Create scene geometry

Some Examples

- Example 3
  - Viewport notification
  - Scene iteration
- Example 5
  - Determines selected type
  - Iterates the Mesh
  - Creates a new face for each original

Demonstration

- Using CuiActionCommandAdapter with new functionality
  - MaxCustomControl.dll (for CuiAction)
  - Autodesk.Max.dll (to create and manipulate geometry)
- Managed Utility Plug-in
  - XAML based UI
  - Autodesk.Max.dll (to create and manipulate geometry)

Advanced Topics

- Using Mixed-Mode C++/CLI
  - Reasons include
    - Exposing .NET APIs for YOUR plug-in
    - Utilize the .NET Framework components
- Debugging caveats
  - Mixed-Mode debugging doesn’t work in x64 with VS2008, but works now in VS2010 when .NET Framework 4.0 is installed
  - Sometimes there can be strange behavior in wrappers

3ds Max Customization Resources

- Autodesk AREA http://area.autodesk.com/
    - http://area.autodesk.com/forum/autodesk-3ds-max
    - http://area.autodesk.com/blogs/chris

- http://www.gamedev.net
- http://www.cgsociety.org
- http://code.google.com/p/3ds-max-dev/

3ds Max Customization Resources

- Autodesk 3ds Max Developer Center
  - www.autodesk.com/develop3dsmax

- Autodesk Developer Network
  - www.autodesk.com/joinadn

Q&A

- Feel free to contact me later:
  - kevin.vandecar@autodesk.com