STUDENT ACTIVITY 5.2 KEY

MTA Course: Web Development Fundamentals
Lesson name: Web Development Fundamentals 5.2
Topic: Configure projects and solutions and reference assemblies
File name: WebDevFund_SA_5.2_Key

Lesson Objective:

5.2: Configure projects and solutions and reference assemblies. *This objective may include but is not limited to:* local assemblies, shared assemblies (GAC), Web application projects, solutions; configuration files; AppSettings.

Resources, software, and additional files needed for this lesson:

- Use these sites for additional information:
  Video lessons and references:
  Introduction to Web Application Projects
  http://msdn.microsoft.com/en-us/library/aa730880%28VS.80%29.aspx
  http://msdn.microsoft.com/en-us/library/k4cbh4dh.aspx
  http://msdn.microsoft.com/en-us/library/610xe886.aspx

Directions to the student:

View the video *How Do I Determine Whether to Use a Web Site or a Web Application Project* (http://www.asp.net/learn/videos/video-410.aspx). Complete the chart below to demonstrate your understanding of the differences and similarities between Visual Studio® Web Site and Web Application Projects.

Attribute	Visual Studio Web Site Project type	Visual Studio Web Application Project type
Historical background	Roots in VS 2005 VS 2008 has both project types	Roots in VS 2003 VS 2008 has both project types
Creation process	Select a New Website option	Select a New Project option

Folder arrangement	Purely a folder-based project containing aspx pages and code A regular Windows folder arrangement in which all elements are fluid and dynamic	A structured VS project with project files (bin folder plus compiled assembly, etc.) Total control of code files and folder creation
Namespaces	No explicit name spaces created – have to add namespaces manually	Explicit namespace creation (in C#)
Classes	Can create a class library and reference them from the pages Can use different languages in different classes because everything	Can create a class library and reference them from the pages All classes are restricted to one language
Master Pages	Add via "New Item" to the Web Site project then the "Add New Item …Web Form" gives option to select the Master Page	Add a New Item to Web Application project then must add New Item → Web Content Form to see the choice to select the Master Page
Referencing a user control	More difficult to do Best through a class library	Easy to do because every entity is a class or object Easier to manipulate controls programmatically
Compiling	Folder based so everything is easily compiles dynamically Deployment is easy – just copy files to IIS Might have a lag in time to serve up the page because it is dynamically compiled when site is opened "Publishing" option enables pre-compilation and creates multiple assemblies representing a folder of the website	VS chooses the compiler based upon the language used in code within any given folder (only 1 language per folder) Creates a single assembly Dialogs offer considerable control over the compilation process Can also "Publish" which offers some additional granularity
Previewing site	Uses Local Development Server built into VS	Uses Local Development Server built into VS

When this is the best choice	Best when: Site is smaller More often updated Need to use classes in various languages	Best when: Precompiled code is most efficient Flexibility of total folder control is desired Many classes are included