Visual Studio Technical Articles

Visual Studio Class Designer

Summary: The Visual Studio Class Designer lets you visualize the structure of classes and their relationships, create new classes using a visual design environment, and easily refactor classes. This whitepaper walks you through some of these tasks. (7 printed pages)

Contents

Introduction

The Visual Studio Class Designer is a fully-functional, visual design environment for the Common Language Runtime. The Visual Studio Class Designer lets you visualize the structure of classes and other types, and through these visual representations edit their source code. Changes made to the class diagram are immediately reflected in code, and changes made to the code immediately affect the appearance of the designer. This synchronous relationship between designer and code makes it easy to create and configure complex CLR types visually.

The Class Designer contains features specifically designed to help you refactor your code as well as allow you to easily rename identifiers and override methods. You can automatically generate classes and structures, and implement interfaces by automatically generating stubs.

Finally, Class Designer also serves as a communication tool by letting you easily communicate areas of your code base to colleagues. Class diagrams can be printed to hard copy or saved as images for display in HTML pages or PowerPoint presentations.

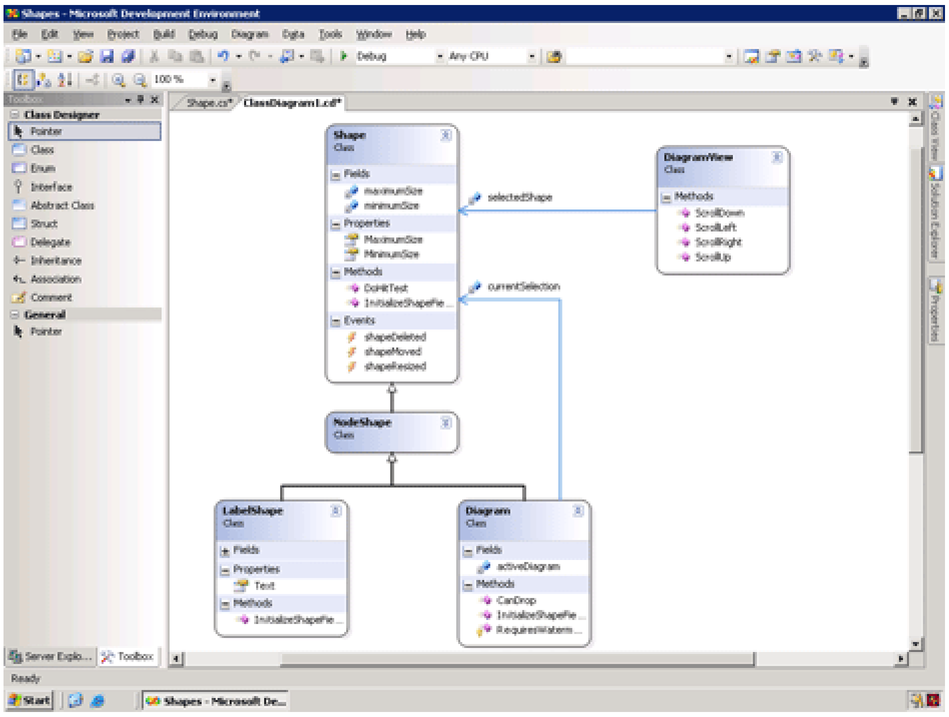


Figure 1. The Class Designer

Why Use a Visual Class Designer?

Software design is a difficult and complex task. Challenges occur at all points in the development cycle, from the early design phases, through code reviews, down to documentation of the final product. A visual class designer can be useful throughout the entire development cycle, such as in the following scenarios:

- Understanding existing code: Existing code bases can be complicated and confusing. With a visual class designer, you can graphically explore existing class hierarchies and get a feel for how classes relate to each other.
- Class Design: A visual class designer allows you to graphically create the high level design and implementation of your software.
- Reviewing and refactoring Code: A visual class designer is a powerful tool for code reviews and refactoring. Existing code diagrams can be annotated for review, and code can be refactored using a designer, saving time.
- Class diagrams for documentation: Class diagrams can be used to document the existing class hierarchies, graphically displaying inheritance trees. Class diagrams are also useful for communicating ideas with colleagues, through email or visual presentations.

Visual Studio Class Designer

The Visual Studio Class Designer is a visual code design tool that is an integrated design experience for the .NET Framework. The visual experience of Class Designer is closely tied to the common language runtime. CLR shapes such as classes, structures, and interfaces are represented by visually distinct shapes that indicate their identity. Furthermore, the terminology in the diagram is language specific-for example in Visual Basic, you might work with Public, Private and Friend access levels, whereas in C# they will be displayed as public, private, and internal. The close integration of the Class Designer with the CLR makes it an ideal tool for authoring classes using the .NET Framework.

The Visual Studio Class Designer is relevant throughout the entire development cycle, providing functionality for all the key scenarios mentioned above. For example:

- Understanding existing code: The Visual Studio Class Designer allows you to quickly and easily examine how classes are related to each other. Not only can you examine the inheritance hierarchy of existing code, but referenced types and .NET assemblies can be examined as well, allowing you to visually explore and become familiar existing types.
- Class Design: The Visual Studio Class Designer facilitates rapid design of classes and class hierarchies. Using familiar drag-and-drop functionality, you can diagram your classes while keeping in sync with the code editor. Changes made to the class diagram are immediately reflected in code, and vice versa. The class diagram always shows a live view of the code.
- Reviewing and refactoring Code: The Visual Studio Class Designer is a powerful tool for code reviews and refactoring. You can add comments to existing code diagrams for later action and the built-in refactoring features allow mundane tasks such as renaming a symbol or encapsulating fields in properties to be accomplished quickly and painlessly.
- Class diagrams for documentation: You can display existing class diagrams in a variety of ways, such as printing or saving as images for display in HTML pages or Microsoft PowerPoint presentations.

Creating Classes with the Class Designer

Class Designer makes it easy to create and configure classes in your project. The class diagram is actually a live view of your code. Changes to the diagram are automatically synchronized with the code, and vice versa. You can create a simple class by dragging it from the Toolbox to the Class Design surface. Once created in your project, you can open the code editor and add code directly to the new classes. Any changes you make will be echoed in the class diagram.

Once a class is created, you can add members using the Class Details window. To add a method, for example, you click **<add method>** in the Class Details window and type the name of the method. You can then indicate the return type, the access level, and add any comments about the method. Once a method is created, you can add parameters—to the method name in much the same way as you add methods-first by indicating the name of the parameter, then indicating the type, the modifier, and any comments. Properties, fields, and events are added the same way that methods are added. The method editing experience using tree control is very similar to the experience of typing in the code editor—the same keystrokes provide navigation through the cells, and IntelliSense help is available.

Implementing an Interface

The Class Designer makes it easy to implement interfaces in your classes. In fact, if your interface is displayed on the class designer surface, you can implement it using the same procedure you would use to inherit a class; by drawing an inheritance line from the class to the interface. If the interface is not displayed in Class Designer, however, it is still easy to implement. You simply drag the interface from the Class view onto the class that you want to implement. Method stubs are automatically generated for the methods defined in the interface. Once implemented, you can add the specific implementation code in the Code Editor.

Visualizing an Inheritance Hierarchy

You can use Class Designer to visualize inheritance hierarchies in a project. To show the base class of an inherited class, right-click the header area of the class and click **Show Base Class**. The base class appears on the diagram.

To show classes that inherit from an existing class, right-click the header area of the class and select **Show Derived Types**. The derived classes appear on the diagram, connected to the class by an inheritance line.

Integration with Visual Studio

The Class Designer integrates seamlessly with Visual Studio 2005 Team System. You can work with the Class Designer in the same space and in the same familiar manner as you would with the regular Visual Studio tools. When working in the Class Designer, the Toolbox is populated with Class Designer tools that are accessible via familiar drag-and-drop action, and the Class Details and Properties windows provide access to the type members. Drag-and-drop action is also available from standard Visual Studio tool windows such as Solution Explorer or the Class View. All changes made in the class designer are immediately synchronized with the corresponding code files. Those code files can then be opened and edited regularly using the Visual Studio code editor.

The Class Diagram

The class diagram is a live view of your code that is constantly updated with changes. The class diagram displays the classes of an existing project. Using a class diagram, you can visualize relationships between classes in a project, and edit and add members to individual classes. The class diagram file exists as a part of your project, is persisted, and can always be used to view your code graphically.

You can add a new class diagram to an existing project by selecting **New Item** from the **Project** menu, then choosing class diagram. The Class Diagram is not a compliable part of your project-rather it is a tool that aids the process of building and editing your classes. Once created, you can add existing classes to Class Designer by dragging them from the Class view to class design surface. You can also add new classes and other types by dragging them from the Toolbox to Class Designer.

Once added to Class Designer, classes and other types are represented by shapes that can be selected and manipulated. Selecting a shape in Class Designer causes its details to become visible in the Class Details window. The class diagram itself stores only visual information, and contains no information about the content of the code. Deleting the class diagram file will not result in any lost code.

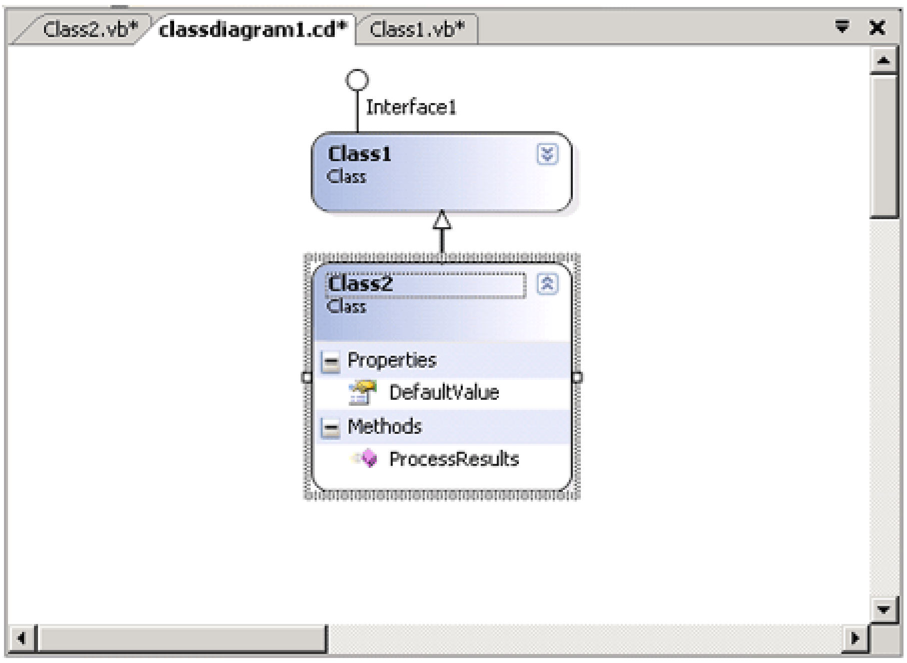


Figure 2. The Class Diagram

The Toolbox

When viewing a class diagram, you can add new members to the class diagram from the Toolbox. The Toolbox contains classes, structures, delegates, enums and other types that you can add to the class diagram. When a type is added from the Toolbox, the appropriate code and code files are added to the project.

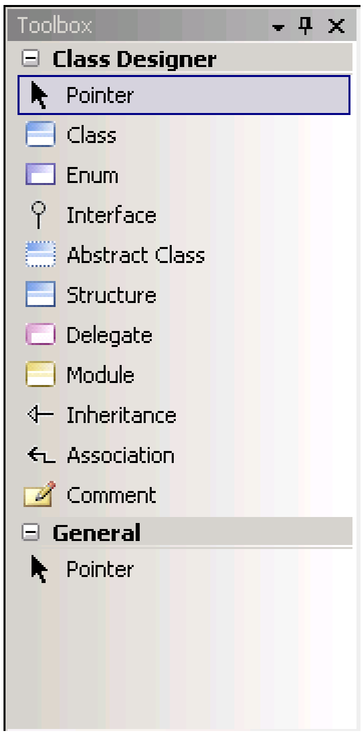
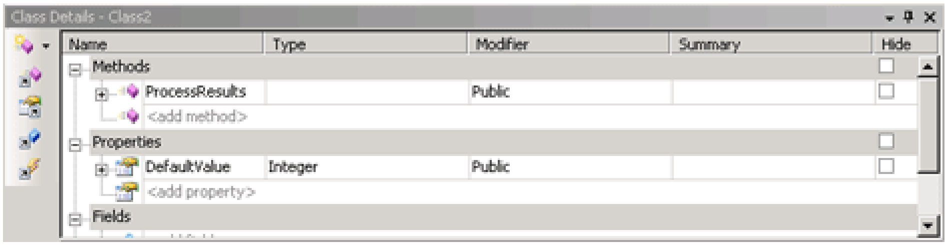


Figure 3. The Toolbox

The Class Details Window

The Class Details window displays the methods, properties, fields, and events that a type displayed in Class Designer possesses. You can use the Class Details window to rapidly edit the members of a class or structure in the class designer. For example, you can change the return type of a method, add parameters, or change the access level. All changes made in the Class Details window are immediately reflected in code.



Name	Type	Modifier	Summary	Hide
Methods				☐
ProcessResults		Public		☐
<add method>				
Properties				☐
DefaultValue	Integer	Public		☐
<add property>				
Fields				

Figure 4. The Class Details Window

Conclusion

Class Designer is a tool in Visual Studio that allows you to rapidly create and configure classes and interfaces in your project. This whitepaper has touched upon some of the most basic scenarios that Class Designer can be used for, such as understanding existing code, class design, reviewing and refactoring code, and using diagrams for documentation. You are encouraged to explore the capabilities of Class Designer further, and take advantage this new tool for software development