**What is New in Visual C#**

Updated: July 2008

 What's New in Visual C# 2008 Service Pack 1

### C# Language

|  |  |
| --- | --- |
| **Feature** | **Description** |
| Various Compiler Improvements | Many improvements have been made to the C# compiler to remove inconsistencies with the language specification. Some of these improvements are breaking changes, but others are just software updates or enhancements. For more information about the breaking changes, see [Visual C# 2008 Breaking Changes](http://msdn.microsoft.com/en-us/library/cc713578.aspx). For more information about the other bugs fixed in Service Pack 1, see [List of changes and fixed issues for Visual Studio 2008 Service Pack 1 Beta for Visual C#](http://go.microsoft.com/fwlink/?LinkId=121489). |

### Visual C# IDE

|  |  |
| --- | --- |
| **Feature** | **Description** |
| Live Semantic Errors | Visual C# 2008 Service Pack 1 introduces Live Semantic Errors, a new feature that provides a more complete set of error information about your code. This feature detects expression-level errors that were previously reported only after a build. It highlights the errors with red wavy underlines as you write code. For more information about wavy underlines, see [Editing Code (Visual C#)](http://msdn.microsoft.com/en-us/library/ms228282.aspx). |
| LINQ Query Debugging | Debugging of LINQ queries is greatly enhanced in Service Pack 1.   * Range variables can now be viewed in the Watch window and in DataTips when you step through individual lines in a query expression. * The debugger no longer hides anonymous type names, and it allows them to appear in casts and instantiations. This enables temporary instances of anonymous types to be created during a debugging session. Casts to and from anonymous types are evaluated correctly. * The strings generated for Add to Watch have been improved and generate expressions that can be evaluated more often than in the original release version of Visual Studio 2008.   For more information, see [Sree's ventures in code space](http://go.microsoft.com/fwlink/?LinkId=121575) on the Microsoft Web site. |
| Task List Comments Are Displayed for All Files in Solution | In previous releases of Visual Studio, the task list was populated only with task comments in open files. In Visual Studio 2008 Service Pack 1, the C# integrated development environment (IDE) displays the task comments in all open and closed files in the solution. For more information, see [How to: Create Task List Comments](http://msdn.microsoft.com/en-us/library/zce12xx2.aspx). |
| Rename Refactoring in Windows Presentation Foundation Projects | Visual Studio 2008 Service Pack 1 enables you to use the rename refactoring feature to rename references that are defined in XAML. For more information about rename refactoring, see [Rename](http://msdn.microsoft.com/en-us/library/6kxxabwd.aspx). |

### LINQ

|  |  |
| --- | --- |
| **Feature** | **Description** |
| Improvements to the System.Linq.Enumerable.Cast<T> Method | Visual C# 2008 Service Pack 1 improves the performance of the Enumerable.Cast<T> method by disabling the primitive value conversions and the explicitly defined user conversions. An example of a primitive value conversion is a conversion from the [int](http://msdn.microsoft.com/en-us/library/5kzh1b5w.aspx) data type to the [long](http://msdn.microsoft.com/en-us/library/ctetwysk.aspx) data type. |
| Improvements in the Performance of System.Linq.Enumerable.Where<TSource> and System.Linq.Enumerable.Select<TSource, TResult> Methods | Visual C# 2008 Service Pack 1 improves the performance of the Where standard query operator and of the Select standard query operator. |

 What's New in the Original Release Version of Visual C# 2008

### C# 3.0 Language

The C# 3.0 language and compiler introduce several new language features. These new language constructs are useful individually in various contexts, and collectively for doing Language-Integrated Query (LINQ). For more information about LINQ, see [The LINQ Project](http://go.microsoft.com/fwlink/?LinkId=71829).

The following table lists the new C# 3.0 language features:

|  |  |
| --- | --- |
| **Feature** | **Description** |
| [Implicitly Typed Local Variables and Arrays](http://msdn.microsoft.com/en-us/library/bb384061.aspx) | When used with local variables, the [var](http://msdn.microsoft.com/en-us/library/bb383973.aspx) keyword instructs the compiler to infer the type of the variable or the array elements from the expression on the right side of the initialization statement. |
| [Object Initializers](http://msdn.microsoft.com/en-us/library/bb384062.aspx) | Enables object initialization without explicit calls to a constructor. |
| [Collection Initializers](http://msdn.microsoft.com/en-us/library/bb384062.aspx) | Enables initialization of collections with an initialization list rather than specific calls to **Add** or another method. |
| [Extension Methods](http://msdn.microsoft.com/en-us/library/bb383977.aspx) | Extend existing classes by using static methods that can be invoked by using instance method syntax. |
| [Anonymous Types](http://msdn.microsoft.com/en-us/library/bb397696.aspx) | Enables on-the-fly creation of unnamed structured types that can be added to collections and accessed by using var. |
| [Lambda Expressions](http://msdn.microsoft.com/en-us/library/bb397687.aspx) | Enables inline expressions with input parameters that can be bound to delegates or expression trees. See also [Anonymous Functions (C# Programming Guide)](http://msdn.microsoft.com/en-us/library/bb882516.aspx). |
| [Query Keywords](http://msdn.microsoft.com/en-us/library/bb310804.aspx) | Keywords that specify clauses in a query expression:   * from clause(s) * where clause (optional) * ordering clauses (optional) * join clause (optional) * select or group clause * into clause (optional) |
| [Auto-Implemented Properties](http://msdn.microsoft.com/en-us/library/bb384054.aspx) | Enables declaration of properties by using simplified syntax. |
| [Partial Method Definitions](http://msdn.microsoft.com/en-us/library/wa80x488.aspx) | Partial types may now contain partial methods. |

### C# 3.0 Compiler

[/win32Manifest](http://msdn.microsoft.com/en-us/library/bb545961.aspx) and [/noWin32Manifest](http://msdn.microsoft.com/en-us/library/bb513864.aspx) compiler switches.

These new compiler switches are used to specify requested execution levels for programs running on Windows Vista.

### Visual C# IDE

|  |  |
| --- | --- |
| **Feature** | **Description** |
| Multi-targeting | Visual Studio 2008 enables you to specify a version of the .NET Framework for your project, .NET Framework 2.0, 3.0, or 3.5. The .NET Framework target of an application is the version of the .NET Framework that is required on a computer to enable the application to run on that computer. For more information, see [Targeting a Specific .NET Framework](http://msdn.microsoft.com/en-us/library/bb398195.aspx). |
| New Project Types and Templates | Several new project templates are provided for Windows Presentation Foundation, Windows Communication Foundation, and Web projects. For more information, see [Project Templates in Visual C# Editions](http://msdn.microsoft.com/en-us/library/ms236405.aspx) and [Default Project Templates in Visual Studio](http://msdn.microsoft.com/en-us/library/0fyc0azh.aspx). |
| IntelliSense support for C# 3.0 | The Visual C# code editor provides statement completion and Quick Info to support the following new language constructs in C# 3.0:   * Implicitly Typed Local Variables * Query Expressions * Extension Methods * Object/Collection Initializers * Anonymous Types * Lambda Expressions * Partial Methods   For more information, see [Visual C# IntelliSense](http://msdn.microsoft.com/en-us/library/43f44291.aspx). |
| Refactoring Support for C# 3.0 | The refactoring features, Rename, Signature Change, Extract Method, and Promote Local have been updated to support the following new language constructs:   * Query Expressions * Extension Methods * Lambda Expressions   In addition, refactoring provides new options and warnings to help prevent unintended consequences from a refactoring action. For more information, see [Refactoring](http://msdn.microsoft.com/en-us/library/719exd8s.aspx). |
| Code Formatting | The code editor supports formatting options for several new C# 3.0 language constructs including query expressions. For more information, see [Formatting, C#, Text Editor, Options Dialog Box](http://msdn.microsoft.com/en-us/library/03864tbz.aspx). |
| [Organizing Using Statements](http://msdn.microsoft.com/en-us/library/bb514114.aspx) | The Visual C# code editor provides an easy way to sort using and extern declarations and remove those that are not being used. |