What's new in C# 3.0?

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 | When used with local variables, the var 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 | Enables object initialization without explicit calls to a constructor. |
| Collection Initializers | Enables initialization of collections with an initialization list rather than specific calls to **Add** or another method.  Contact c1 = new Contact { Name = "John", YearofBirth = 1980 }; |
| Extension Methods | Extend existing classes by using static methods that can be invoked by using instance method syntax.  The following is an example of a valid extension method:  public static class string  {  public static int ToInt32(this string s)  {  return Convert.ToInt32(s) ;  }  } |
| Anonymous Types | Enables on-the-fly creation of unnamed structured types that can be added to collections and accessed by using var.  var contact1 = new  {  id = "54321",  Name = "Wei-Meng Lee",  email = "weimenglee@learn2develop.net"  }; |
| Lambda Expressions | Enables inline expressions with input parameters that can be bound to delegates or expression trees. See also Anonymous Functions (C# Programming Guide).  Lambda expressions have the following format:  params => expression(s)  button1.Click += (object \_sender,EventArgs \_args) =>  {  Console.WriteLine("Button clicked!");  }; |
| Query Keywords | 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 | Enables declaration of properties by using simplified syntax. |
| Partial Method Definitions | Partial types may now contain partial methods.   which allows to separate method definition and implementation. |