.NET Framework 4

**.NET Framework Class Library Overview**

The .NET Framework includes classes, interfaces, and value types that expedite and optimize the development process and provide access to system functionality. To facilitate interoperability between languages, most .NET Framework types are CLS-compliant and can therefore be used from any programming language whose compiler conforms to the common language specification (CLS).

The .NET Framework types are the foundation on which .NET applications, components, and controls are built. The .NET Framework includes types that perform the following functions:

* Represent base data types and exceptions.
* Encapsulate data structures.
* Perform I/O.
* Access information about loaded types.
* Invoke .NET Framework security checks.
* Provide data access, rich client-side GUI, and server-controlled, client-side GUI.

The .NET Framework provides a rich set of interfaces, as well as abstract and concrete (non-abstract) classes. You can use the concrete classes as is or, in many cases, derive your own classes from them. To use the functionality of an interface, you can either create a class that implements the interface or derive a class from one of the .NET Framework classes that implements the interface.

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifNaming Conventions

.NET Framework types use a dot syntax naming scheme that connotes a hierarchy. This technique groups related types into namespaces so they can be searched and referenced more easily. The first part of the full name — up to the rightmost dot — is the namespace name. The last part of the name is the type name. For example, **System.Collections.ArrayList** represents the **ArrayList** type, which belongs to the **System.Collections** namespace. The types in **System.Collections** can be used to manipulate collections of objects.

This naming scheme makes it easy for library developers extending the .NET Framework to create hierarchical groups of types and name them in a consistent, informative manner. It also allows types to be unambiguously identified by their full name (that is, by their namespace and type name), which prevents type name collisions. It is expected that library developers will use the following guideline when creating names for their namespaces:

*CompanyName*.*TechnologyName*

For example, the namespace Microsoft.Word conforms to this guideline.

The use of naming patterns to group related types into namespaces is a very useful way to build and document class libraries. However, this naming scheme has no effect on visibility, member access, inheritance, security, or binding. A namespace can be partitioned across multiple assemblies and a single assembly can contain types from multiple namespaces. The assembly provides the formal structure for versioning, deployment, security, loading, and visibility in the common language runtime.

For more information on namespaces and type names, see [Common Type System](http://msdn.microsoft.com/en-us/library/zcx1eb1e.aspx).

Description: http://i.msdn.microsoft.com/Global/Images/clear.gifSystem Namespace

The [System](http://msdn.microsoft.com/en-us/library/system.aspx) namespace is the root namespace for fundamental types in the .NET Framework. This namespace includes classes that represent the base data types used by all applications: [Object](http://msdn.microsoft.com/en-us/library/system.object.aspx) (the root of the inheritance hierarchy), [Byte](http://msdn.microsoft.com/en-us/library/system.byte.aspx), [Char](http://msdn.microsoft.com/en-us/library/system.char.aspx), [Array](http://msdn.microsoft.com/en-us/library/system.array.aspx), [Int32](http://msdn.microsoft.com/en-us/library/system.int32.aspx), [String](http://msdn.microsoft.com/en-us/library/system.string.aspx), and so on. Many of these types correspond to the primitive data types that your programming language uses. When you write code using .NET Framework types, you can use your language's corresponding keyword when a .NET Framework base data type is expected.

In addition to the base data types, the [System](http://msdn.microsoft.com/en-us/library/system.aspx) namespace contains over 100 classes, ranging from classes that handle exceptions to classes that deal with core runtime concepts, such as application domains and the garbage collector. The [System](http://msdn.microsoft.com/en-us/library/system.aspx) namespace also contains many second-level namespaces.

For more information about namespaces, browse the [.NET Framework Class Library](http://go.microsoft.com/fwlink/?LinkID=217856). The reference documentation provides a brief overview of each namespace as well as a formal description of each type and its members.