XML Support

# XML Standards Support in .NET

Although many programming languages and environments have provided XML support as an add-on, .NET’s support is integrated into the framework more tightly than most. The .NET development team decided to use XML extensively within the framework in order to meet its design goals. Accordingly, they built in XML support from the beginning.

The .NET Framework contains five main assemblies that implement the core XML standards. The below mentioned five assemblies, along with a description of the functionality contained in each.

| **Assembly** | **Description** |
| --- | --- |
| System.Xml | Basic XML input and output with XmlReader and XmlWriter, DOM with XmlNode and its subclasses, many XML utility classes |
| System.Xml.Schema | Constraint of XML via XML Schema with XmlSchemaObject and its subclasses |
| System.Xml.Serialization | Serialization to plain XML and SOAP with XmlSerializer |
| System.Xml.XPath | Navigation of XML via XPath with XPathDocument, XPathExpression, and XPathNavigator |
| System.Xml.Xsl | Transformation of XML documents via XSLT with XslTransform |

In addition, the System.Web.Services and System.Data assemblies contain classes that interact with the XML assemblies. The XML assemblies used internally in the .NET Framework are also available for use directly in your applications.

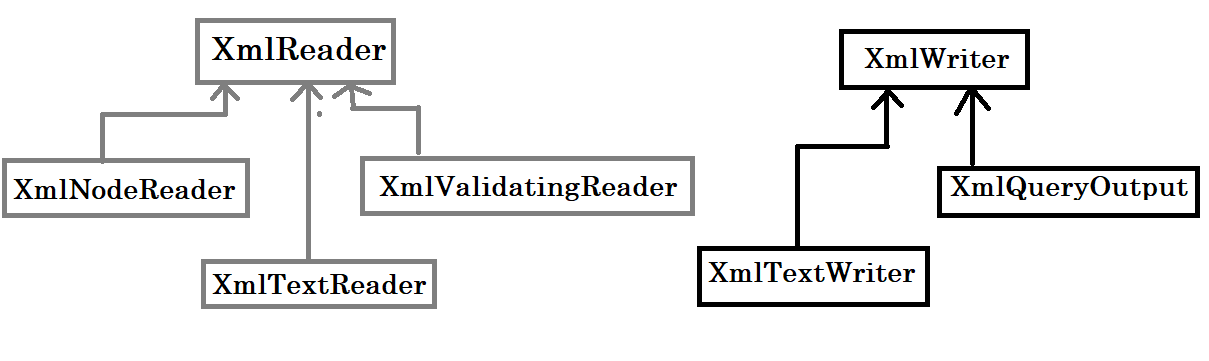
For example, the System.Data assembly handles database operations. Its DataSet class provides a mechanism to transmit database changes using XML. But you can also access the XML generated by the DataSet and manipulate it just as you would any XML file, using classes in the System.Xml namespace.

Besides the .NET Framework’s XML assemblies, there are several tools integrated into Visual Studio .NET and shipped with the .NET Framework SDK that can make your life easier when dealing with XML. These tools include xsd.exe, wsdl.exe, and disco.exe, among others.

# System.Xml Namespace and related Classes

|  |  |  |
| --- | --- | --- |
| **Sl.** | **Class Name** | **Description** |
| 1 | XmlReader | An abstract reader class that provides fast, non cached xml data XMLReader is forward only. |
| 2 | XmlWriter | An abstract writer class that provides fast, non cached xml data in stream or file format. |
| 3 | XmlTextReader | Extends XmlReader. Provides FF only Stream access to XML Data. |
| 4 | XmlTextWriter | Extends XmlWriter. Provides Forward only generation of XML Streams. |
| 5 | XmlNode | An abstract class that represents a single node in a XML document. Base class for several classes in Xml namespace. |
| 6 | XmlDocument | Extends XmlNode. This is the W3C DOM implementation. It represents a Tree representation of In memory of an XML document, enabling navigation and editing. |
| 7 | XmlDataDocument | Extends XmlDocument. This is document that can be loaded from XML data or from an relational data in an ADO.NET dataset. Allows the mixing of XML and relational data in the same view. |

# Reading and Writing Streamed XML



XmlTextReader and XmlTextWriter (Text based classes) work with either a stream based object from the System.IO or TextReader / TextWriter objects.

XmlNodeReader uses an XmlNode as its source instead of a stream.

# Using the DOM in .NET

# Using XPathNavigators

# XML and ADO.NET

# Serializing Objects in XML

# LINQ to XML and .NET

# Working with Different XML Objects

# Using LINQ to Query XML Documents

# More Query Techniques for XML Documents

1. Asynchronous Patterns -

<https://docs.microsoft.com/en-us/dotnet/standard/asynchronous-programming-patterns/>

1. Delegates and Async Programming

<https://www.c-sharpcorner.com/uploadfile/vendettamit/delegates-and-async-programming/>