System.Xml.XmlReader Class

```
[ILASM]
.class public abstract XmlReader extends System.Object

[C#]
public abstract class XmlReader
```

Assembly Info:

- *Name:* System.Xml
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- *Version:* 1.0.x.x
- *Attributes:*
  - CLSCompliantAttribute(true)

Type Attributes:

- DefaultMemberAttribute("Item") [*Note:* This attribute requires the RuntimeInfrastructure library.]

Summary

Represents a reader that provides non-cached, forward-only access to XML data.

Inherits From: System.Object

Library: XML

Thread Safety: All public static members of this type are safe for multithreaded operations. No instance members are guaranteed to be thread safe.

Description

This class provides forward-only, read-only access to a stream of XML data. This class enforces the rules of well-formed XML but does not perform data validation.

This class conforms to the W3C Extensible Markup Language (XML) 1.0 and the Namespaces in XML recommendations.

A given set of XML data is modeled as a tree of nodes. The different types of nodes are specified in the **System.Xml.XmlNodeType** enumeration. The reader is advanced to the next node using the **System.Xml.XmlReader.Read** method. The current node refers to the node on which the reader is positioned. The following table lists the node properties exposed for the current node.

Property	Description
AttributeCount	The number of attributes on the node.
BaseUri	The base URI of the node.
Depth	The depth of the node in the tree.
HasAttributes	Whether the node has attributes.
HasValue	Whether the node can have a text value.
IsDefault	Whether an **Attribute** node was generated from the default value defined in the DTD or schema.
IsEmptyElement	Whether an **Element** node is empty.
LocalName	The local name of the node.
Name	The qualified name of the node, equal to **Prefix**:**LocalName**.
NamespaceUri	The URI defining the namespace associated with the node.
NodeType	The **System.Xml.XmlNodeType** of the node.
Prefix	A shorthand reference to the namespace associated with the node.
QuoteChar	The quotation mark character used to enclose the value of an attribute.
Value	The text value of the node.
XmlLang	The `xml:lang` scope within which the node resides.

This class does not expand default attributes or general entities. Any general entities encountered are returned as a single empty **EntityReference** node.

This class checks that a Document Type Definition (DTD) is well-formed, but does not validate using the DTD.

To read strongly typed data, use the **System.Xml.XmlConvert** class.

This class throws a **System.Xml.XmlException** on XML parse errors. After an exception is thrown, the state of the reader is not predictable. For example, the reported node type may be different than the actual node type of the current node.

[*Note:* This class is **abstract** and implemented in the **System.Xml.XmlTextReader** class.]

XmlReader() Constructor

[ILASM]
```
family specialname instance void .ctor()
```

[C#]
```
protected XmlReader()
```

Summary

Constructs a new instance of the **System.Xml.XmlReader** class.

XmlReader.Close() Method

```
[ILASM]
.method public hidebysig virtual abstract void Close()

[C#]
public abstract void Close()
```

Summary

Changes the **System.Xml.XmlReader.ReadState** to **System.Xml.ReadState.Closed**.

Behaviors

This method releases any resources allocated by the current instance, changes the **System.Xml.XmlReader.ReadState** to **System.Xml.ReadState.Closed**, and calls the **Close** method of any underlying **System.IO.Stream** or **System.IO.TextReader** instance.

How and When to Override

This method must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.GetAttribute(System.Int32) Method

```
[ILASM]
.method public hidebysig virtual abstract string
GetAttribute(int32 i)

[C#]
public abstract string GetAttribute(int i)
```

Summary

Returns the value of the attribute with the specified index relative to the containing element.

Parameters

Parameter	Description
i	A **System.Int32** specifying the zero-based index of the attribute relative to the containing element.

Return Value

A **System.String** containing the value of the specified attribute.

Behaviors

This method does not move the reader.

How and When to Override

This method must be overridden in order to provide the functionality described above, as there is no default implementation.

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	*i* is less than 0, or greater than or equal to the **System.Xml.XmlReader.AttributeCount** of the containing element.

Example

For an example demonstrating this method, see **System.Xml.XmlTextReader.GetAttribute**(**System.String**, **System.String**).

XmlReader.GetAttribute(System.String, System.String) Method

```
[ILASM]
.method public hidebysig virtual abstract string
GetAttribute(string name, string namespaceURI)

[C#]
public abstract string GetAttribute(string name, string
namespaceURI)
```

Summary

Returns the value of the attribute with the specified local name and namespace URI.

Parameters

Parameter	Description
name	A **System.String** specifying the local name of the attribute.
namespaceURI	A **System.String** specifying the namespace URI of the attribute.

Return Value

A **System.String** containing the value of the specified attribute, or **null** if the attribute is not found.

Behaviors

This method does not move the reader.

How and When to Override

This method must be overridden in order to provide the functionality described above, as there is no default implementation.

Example

For an example demonstrating this method, see **System.Xml.XmlTextReader.GetAttribute**(**System.String**, **System.String**).

XmlReader.GetAttribute(System.String) Method

```
[ILASM]
.method public hidebysig virtual abstract string
GetAttribute(string name)

[C#]
public abstract string GetAttribute(string name)
```

Summary

Returns the value of the attribute with the specified qualified name.

Parameters

Parameter	Description
name	A **System.String** specifying the qualified name of the attribute.

Return Value

A **System.String** containing the value of the specified attribute, or **null** if the attribute is not found.

Behaviors

This method does not move the reader.

How and When to Override

This method must be overridden in order to provide the functionality described above, as there is no default implementation.

Example

For an example demonstrating this method, see **System.Xml.XmlTextReader.GetAttribute**(**System.String**, **System.String**).

XmlReader.IsName(System.String) Method

```
[ILASM]
.method public hidebysig static bool IsName(string str)

[C#]
public static bool IsName(string str)
```

Summary

Determines whether the specified string is a valid XML name.

Parameters

Parameter	Description
str	A **System.String** specifying the name to validate.

Return Value

A **System.Boolean** where **true** indicates the name is valid; otherwise, **false**.

Description

[*Note:* This method uses the W3C XML 1.0 Recommendation (http://www.w3.org/TR/2000/REC-xml-20001006#NT-Name) to determine whether the name is valid.]

XmlReader.IsNameToken(System.String) Method

```
[ILASM]
.method public hidebysig static bool IsNameToken(string
str)

[C#]
public static bool IsNameToken(string str)
```

Summary

Determines whether the specified string is a valid XML name token (Nmtoken).

Parameters

Parameter	Description
str	A **System.String** specifying the name to validate.

Return Value

A **System.Boolean** where **true** indicates the name is valid; otherwise **false**.

Description

[*Note:* This method uses the W3C XML 1.0 Recommendation (http://www.w3.org/TR/2000/REC-xml-20001006#NT-Nmtoken) to determine whether the name token is valid.]

XmlReader.IsStartElement(System.String, System.String) Method

```
[ILASM]
.method public hidebysig virtual bool IsStartElement(string
localname, string ns)

[C#]
public virtual bool IsStartElement(string localname, string
ns)
```

Summary

Determines if a node containing content is an **Element** node with the specified local name and namespace URI.

Parameters

Parameter	Description
localname	A **System.String** specifying the local name of an element.
ns	A **System.String** specifying the namespace URI associated with the element.

Return Value

A **System.Boolean** where **true** indicates the node is an **Element** node with the specified local name and namespace URI; **false** otherwise.

Behaviors

As described above.

Default

This method calls the **System.Xml.XmlReader.MoveToContent** method, which determines whether the current node can contain content and, if not, moves the reader to the next content node or the end of the input stream. When the reader is positioned on a content node, the node is checked to determine if it is an **Element** node with **System.Xml.XmlReader.LocalName** and **System.Xml.XmlReader.NamespaceURI** properties equal to *localname* and *ns*, respectively.

How and When to Override

Override this method to customize the behavior of this method in types derived from the **System.Xml.XmlReader** class.

Usage

Use this method to determine whether the node returned by the **System.Xml.XmlReader.MoveToContent** method is an **Element** node with the specified local name and namespace URI.

Exceptions

Exception	Condition
System.Xml.XmlException	An error occurred while parsing the XML.

XmlReader.IsStartElement(System.String) Method

```
[ILASM]
.method public hidebysig virtual bool IsStartElement(string
name)

[C#]
public virtual bool IsStartElement(string name)
```

Summary

Determines if a node containing content is an **Element** node with the specified qualified name.

Parameters

Parameter	Description
name	A **System.String** specifying the qualified name of an element.

Return Value

A **System.Boolean** where **true** indicates the node is an **Element** node with the specified name; **false** otherwise.

Behaviors

As described above.

Default

This method calls the **System.Xml.XmlReader.MoveToContent** method, which determines whether the current node can contain content and, if not, moves the reader to the next content node or the end of the input stream. When the reader is positioned on a content node, the node is checked to determine if it is an **Element** node with a **System.Xml.XmlReader.Name** property equal to *name*.

How and When to Override

Override this method to customize the behavior of this method in types derived from the **System.Xml.XmlReader** class.

Usage

Use this method to determine whether the node returned by the **System.Xml.XmlReader.MoveToContent** method is an **Element** node with the specified name.

Exceptions

Exception	Condition
System.Xml.XmlException	An error occurred while parsing the XML.

XmlReader.IsStartElement() Method

```
[ILASM]
.method public hidebysig virtual bool IsStartElement()

[C#]
public virtual bool IsStartElement()
```

Summary

Determines if a node containing content is an **Element** node.

Return Value

A **System.Boolean** where **true** indicates the node is an **Element** node; **false** otherwise.

Behaviors

As described above.

Default

This method calls the **System.Xml.XmlReader.MoveToContent** method, which determines whether the current node can contain content and, if not, moves the reader to the next content node or the end of the input stream. When the reader is positioned on a content node, the node is checked to determine if it is an **Element** node.

How and When to Override

Override this method to customize the behavior of this method in types derived from the **System.Xml.XmlReader** class.

Usage

Use this method to determine whether the node returned by the **System.Xml.XmlReader.MoveToContent** method is an **Element** node.

Exceptions

Exception	Condition
System.Xml.XmlException	An error occurred while parsing the XML.

XmlReader.LookupNamespace(System.String) Method

```
[ILASM]
.method public hidebysig virtual abstract string
LookupNamespace(string prefix)

[C#]
public abstract string LookupNamespace(string prefix)
```

Summary

Resolves a namespace prefix in the scope of the current element.

Parameters

Parameter	Description
prefix	A **System.String** specifying the prefix whose namespace URI is to be resolved. To return the default namespace, specify **System.String.Empty**.

Return Value

A **System.String** containing the namespace URI to which the prefix maps. If *prefix* is not in **System.Xml.XmlReader.NameTable** or no matching namespace is found, **null** is returned.

Behaviors

As described above.

How and When to Override

This method must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.MoveToAttribute(System.Int32) Method

```
[ILASM]
.method public hidebysig virtual abstract void
MoveToAttribute(int32 i)

[C#]
public abstract void MoveToAttribute(int i)
```

Summary

Moves the position of the current instance to the attribute with the specified index relative to the containing element.

Parameters

Parameter	Description
i	A **System.Int32** specifying the zero-based index of the attribute relative to the containing element.

Behaviors

After calling this method, the **System.Xml.XmlReader.Name**, **System.Xml.XmlReader.NamespaceURI**, and **System.Xml.XmlReader.Prefix** properties reflect the properties of current attribute.

How and When to Override

This method must be overridden in order to provide the functionality described above, as there is no default implementation.

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	*i* is less than 0 or greater than or equal to the **System.Xml.XmlReader.AttributeCount** of the containing element.

XmlReader.MoveToAttribute(System.String, System.String) Method

```
[ILASM]
.method public hidebysig virtual abstract bool
MoveToAttribute(string name, string ns)

[C#]
public abstract bool MoveToAttribute(string name, string
ns)
```

Summary

Moves the position of the current instance to the attribute with the specified local name and namespace URI.

Parameters

Parameter	Description
name	A **System.String** specifying the local name of the attribute.
ns	A **System.String** specifying the namespace URI of the attribute.

Return Value

A **System.Boolean** where **true** indicates the attribute was found; otherwise, **false**. If **false**, the position of the current instance does not change.

Behaviors

After calling this method, the **System.Xml.XmlReader.Name**, **System.Xml.XmlReader.NamespaceURI**, and **System.Xml.XmlReader.Prefix** properties reflect the properties of current attribute.

How and When to Override

This method must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.MoveToAttribute(System.String) Method

```
[ILASM]
.method public hidebysig virtual abstract bool
MoveToAttribute(string name)

[C#]
public abstract bool MoveToAttribute(string name)
```

Summary

Moves the position of the current instance to the attribute with the specified qualified name.

Parameters

Parameter	Description
name	A **System.String** specifying the qualified name of the attribute.

Return Value

A **System.Boolean** where **true** indicates the attribute was found; otherwise, **false**. If **false**, the reader's position does not change.

Behaviors

After calling this method, the **System.Xml.XmlReader.Name**, **System.Xml.XmlReader.NamespaceURI**, and **System.Xml.XmlReader.Prefix** properties reflect the properties of current attribute.

How and When to Override

This method must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.MoveToContent() Method

```
[ILASM]
.method public hidebysig virtual valuetype
System.Xml.XmlNodeType MoveToContent()

[C#]
public virtual XmlNodeType MoveToContent()
```

Summary

Determines whether the current node can contain content and, if not, moves the position of the current instance to the next content node or the end of the input stream.

Return Value

The **System.Xml.XmlNodeType** of the content node, or **System.Xml.XmlNodeType.None** if the position of the reader has reached the end of the input stream.

Description

[*Note:* The following members of **System.Xml.XmlNodeType** can contain content: **Attribute**, **CDATA**, **Element**, **EndElement**, **EntityReference**, **EndEntity**, and **Text**.]

Behaviors

As described above.

Default

If the current node is an **Attribute** node, this method moves the position of the reader back to the **Element** node that owns the attribute.

How and When to Override

Override this method to customize the behavior of this method in types derived from the **System.Xml.XmlReader** class.

Usage

Use this method to determine whether the current node can contain content and, if not, move the position of the reader to the next content node.

Exceptions

Exception	Condition
System.Xml.XmlException	An error occurred while parsing the XML.

XmlReader.MoveToElement() Method

```
[ILASM]
.method public hidebysig virtual abstract bool
MoveToElement()

[C#]
public abstract bool MoveToElement()
```

Summary

Moves the position of the current instance to the node that contains the current **Attribute** node.

Return Value

A **System.Boolean** where **true** indicates the position of the reader was moved; **false** indicates the reader was not positioned on an **Attribute** node and therefore the position of the reader was not moved.

Description

[*Note:* The **DocumentType**, **Element**, and **XmlDeclaration** members of **System.Xml.XmlNodeType** can contain attributes.]

Behaviors

As described above.

How and When to Override

This method must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.MoveToFirstAttribute() Method

```
[ILASM]
.method public hidebysig virtual abstract bool
MoveToFirstAttribute()

[C#]
public abstract bool MoveToFirstAttribute()
```

Summary

Moves the position of the current instance to the first attribute associated with the current node.

Return Value

A **System.Boolean** where **true** indicates the current node contains at least one attribute; otherwise, **false**.

Behaviors

If **System.Xml.XmlReader.AttributeCount** is non-zero, the position of the reader moves to the first attribute; otherwise, the position of the reader does not change.

How and When to Override

This method must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.MoveToNextAttribute() Method

```
[ILASM]
.method public hidebysig virtual abstract bool
MoveToNextAttribute()

[C#]
public abstract bool MoveToNextAttribute()
```

Summary

Moves the position of the current instance to the next attribute associated with the current node.

Return Value

A **System.Boolean** where **true** indicates the position of the reader moved to the next attribute; **false** if there were no more attributes.

Behaviors

As described above.

How and When to Override

This method must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.Read() Method

```
[ILASM]
.method public hidebysig virtual abstract bool Read()

[C#]
public abstract bool Read()
```

Summary

Moves the position of the current instance to the next node in the stream, exposing its properties.

Return Value

A **System.Boolean** where **true** indicates the node was read successfully, and **false** indicates there were no more nodes to read.

Behaviors

As described above.

How and When to Override

This method must be overridden in order to provide the functionality as described herein, as there is no default implementation.

Usage

When a reader is first created and initialized, there is no information available. Calling this method is required to read the first node.

Exceptions

Exception	Condition
System.Xml.XmlException	An error occurred while parsing the XML.

XmlReader.ReadAttributeValue() Method

```
[ILASM]
.method public hidebysig virtual abstract bool
ReadAttributeValue()

[C#]
public abstract bool ReadAttributeValue()
```

Summary

Parses an attribute value into one or more **Text**, **EntityReference**, and **EndEntity** nodes.

Return Value

A **System.Boolean** where **true** indicates the attribute value was parsed, and **false** indicates the reader was not positioned on an attribute node or all the attribute values have been read.

Description

[*Note:* To parse an **EntityReference** node, call the **System.Xml.XmlReader.ResolveEntity** method. After the node is parsed into child nodes, call the **System.Xml.XmlReader.ReadAttributeValue** method again to read the value of the entity.

The **System.Xml.XmlReader.Depth** of an attribute value node is one plus the depth of the attribute node. When general entity references are stepped into or out of, the **System.Xml.XmlReader.Depth** increments or decrements by one, respectively.]

Behaviors

As described above.

How and When to Override

Implementations that cannot expand general entities should return general entities as a single empty (**System.Xml.XmlReader.Value** equals **System.String.Empty**) **EntityReference** node.

Usage

Use this method after calling **System.Xml.XmlReader.MoveToAttribute** to read through the **Text**, **EntityReference,** or **EndEntity** nodes that make up the

attribute value. Call the **System.Xml.XmlReader.ResolveEntity** method to resolve the **EntityReference** nodes.

XmlReader.ReadElementString(System.String, System.String) Method

```
[ILASM]
.method public hidebysig virtual string
ReadElementString(string localname, string ns)

[C#]
public virtual string ReadElementString(string localname,
string ns)
```

Summary

Reads the contents of a text-only element with the specified local name and namespace URI.

Parameters

Parameter	Description
localname	A **System.String** specifying the local name of an element.
ns	A **System.String** specifying the namespace URI associated with the element.

Return Value

A **System.String** containing the contents of the element.

Behaviors

As described above.

Default

This method calls the **System.Xml.XmlReader.MoveToContent** method. If the returned node is an **Element** node, this method compares the **System.Xml.XmlReader.LocalName** and **System.Xml.XmlReader.NamespaceURI** properties of the node to *localname* and *ns*, respectively. If they are equal, this method calls the **System.Xml.XmlReader.ReadString** method to read the contents of the element.

How and When to Override

Override this method to customize the behavior of this method in types derived from the **System.Xml.XmlReader** class.

Usage

Use this method to read the contents of a text-only element with the specified local name and namespace URI.

Exceptions

Exception	Condition
System.Xml.XmlException	The node is not an **Element** node, the **System.Xml.XmlReader.LocalName** property of the **Element** node does not equal *localname*, or the **System.Xml.XmlReader.NamespaceURI** property of the **Element** node does not equal *ns*, the element does not contain a simple text value, or an error occurred while parsing the XML.

XmlReader.ReadElementString(System.String) Method

```
[ILASM]
.method public hidebysig virtual string
ReadElementString(string name)

[C#]
public virtual string ReadElementString(string name)
```

Summary

Reads the contents of a text-only element with the specified qualified name.

Parameters

Parameter	Description
name	A **System.String** specifying the qualified name of an element.

Return Value

A **System.String** containing the contents of the element.

Behaviors

As described above.

Default

This method calls the **System.Xml.XmlReader.MoveToContent** method and, if the returned node is an **Element** node, compares the **System.Xml.XmlReader.Name** property of the node to *name*. If they are equal, this method calls the **System.Xml.XmlReader.ReadString** method to read the contents of the element.

How and When to Override

Override this method to customize the behavior of this method in types derived from the **System.Xml.XmlReader** class.

Usage

Use this method to read the contents of a text-only element with the specified qualified name.

Exceptions

Exception	Condition
System.Xml.XmlException	The node is not an **Element** node, the **System.Xml.XmlReader.Name** property of the **Element** node does not equal *name*, the element does not contain a simple text value, or an error occurred while parsing the XML.

XmlReader.ReadElementString() Method

```
[ILASM]
.method public hidebysig virtual string ReadElementString()

[C#]
public virtual string ReadElementString()
```

Summary

Reads the contents of a text-only element.

Return Value

A **System.String** containing the contents of the element.

Behaviors

As described above.

Default

This method calls the **System.Xml.XmlReader.MoveToContent** method and, if the returned node is an **Element** node, calls the **System.Xml.XmlReader.ReadString** method to read the contents.

How and When to Override

Override this method to customize the behavior of this method in types derived from the **System.Xml.XmlReader** class.

Usage

Use this method to read the contents of a text-only element.

Exceptions

Exception	Condition
System.Xml.XmlException	The node is not an **Element** node, the element does not contain a simple text value, or an error occurred while parsing the XML.

XmlReader.ReadEndElement() Method

```
[ILASM]
.method public hidebysig virtual void ReadEndElement()

[C#]
public virtual void ReadEndElement()
```

Summary

Reads an **EndElement** node and advances the reader to the next node.

Behaviors

As described above.

Default

This method calls the **System.Xml.XmlReader.MoveToContent** method, which determines whether the current node can contain content and, if not, moves the reader to the next content node or the end of the input stream. The node the reader ends up positioned on is checked to determine if it is an **EndElement** node. If so, the node is read and the reader is moved to the next node.

How and When to Override

Override this method to customize the behavior of this method in types derived from the **System.Xml.XmlReader** class.

Usage

Use this method to read an **EndElement** node and advance the reader to the next node.

Exceptions

Exception	Condition
System.Xml.XmlException	The node is not an **EndElement** node or an error occurred while parsing the XML.

XmlReader.ReadInnerXml() Method

```
[ILASM]
.method public hidebysig virtual abstract string
ReadInnerXml()

[C#]
public abstract string ReadInnerXml()
```

Summary

Reads the contents of the current node, including child nodes and markup.

Return Value

A **System.String** containing the XML content, or **System.String.Empty** if the current node is neither an element nor attribute, or has no child nodes.

Behaviors

The current node and corresponding end node are not returned.

If the current node is an element, after the call to this method, the reader is positioned after the corresponding end element.

If the current node is an attribute, the position of the reader is not changed.

[*Note:* For a comparison between this method and the **System.Xml.XmlReader.ReadOuterXml** method, see **System.Xml.XmlTextReader.ReadInnerXml**.]

How and When to Override

This method must be overridden in order to provide the functionality described above, as there is no default implementation.

Exceptions

Exception	Condition
System.Xml.XmlException	The XML was not well-formed, or an error occurred while parsing the XML.

XmlReader.ReadOuterXml() Method

```
[ILASM]
.method public hidebysig virtual abstract string
ReadOuterXml()

[C#]
public abstract string ReadOuterXml()
```

Summary

Reads the current node and its contents, including child nodes and markup.

Return Value

A **System.String** containing the XML content, or **System.String.Empty** if the current node is neither an element nor attribute.

Behaviors

The current node and corresponding end node are returned.

If the current node is an element, after the call to this method, the reader is positioned after the corresponding end element.

If the current node is an attribute, the position of the reader is not changed.

[*Note:* For a comparison between this method and the **System.Xml.XmlReader.ReadOuterXml** method, see **System.Xml.XmlTextReader.ReadInnerXml**.]

How and When to Override

This method must be overridden in order to provide the functionality described above, as there is no default implementation.

Exceptions

Exception	Condition
System.Xml.XmlException	The XML was not well-formed, or an error occurred while parsing the XML.

XmlReader.ReadStartElement(System.String, System.String) Method

```
[ILASM]
.method public hidebysig virtual void
ReadStartElement(string localname, string ns)

[C#]
public virtual void ReadStartElement(string localname,
string ns)
```

Summary

Reads an **Element** node with the specified local name and namespace URI and advances the reader to the next node.

Parameters

Parameter	Description
localname	A **System.String** specifying the local name of an element.
ns	A **System.String** specifying the namespace URI associated with the element.

Behaviors

As described above.

Default

This method calls the **System.Xml.XmlReader.MoveToContent** method. If the returned node is an **Element** node, this method compares the **System.Xml.XmlReader.LocalName** and **System.Xml.XmlReader.NamespaceURI** properties of the node to *localname* and *ns*, respectively. If they are equal, this method calls the **System.Xml.XmlReader.Read** method to read the element and move to the next node.

How and When to Override

Override this method to customize the behavior of this method in types derived from the **System.Xml.XmlReader** class.

Usage

Use this method to read an **Element** node with the specified local name and namespace URI, and advance the reader to the next node.

Exceptions

Exception	Condition
System.Xml.XmlException	The node is not an **Element** node, the **System.Xml.XmlReader.LocalName** property of the **Element** node does not equal *localname*, the **System.Xml.XmlReader.NamespaceURI** property of the **Element** node does not equal *ns*, or an error occurred while parsing the XML.

XmlReader.ReadStartElement(System.String) Method

```
[ILASM]
.method public hidebysig virtual void
ReadStartElement(string name)

[C#]
public virtual void ReadStartElement(string name)
```

Summary

Reads an **Element** node with the specified qualified name and advances the reader to the next node.

Parameters

Parameter	Description
name	A **System.String** specifying the qualified name of an element.

Behaviors

As described above.

Default

This method calls the **System.Xml.XmlReader.MoveToContent** method and, if the returned node is an **Element** node, compares the **System.Xml.XmlReader.Name** property of the node to *name*. If they are equal, this method calls the **System.Xml.XmlReader.Read** method to read the element and move to the next node.

How and When to Override

Override this method to customize the behavior of this method in types derived from the **System.Xml.XmlReader** class.

Usage

Use this method to read an **Element** node with the specified qualified name, and advance the reader to the next node.

Exceptions

Exception	Condition
System.Xml.XmlException	The node is not an **Element** node, the **System.Xml.XmlReader.Name** property of the **Element** node does not equal *name*, or an error occurred while parsing the XML.

XmlReader.ReadStartElement() Method

```
[ILASM]
.method public hidebysig virtual void ReadStartElement()

[C#]
public virtual void ReadStartElement()
```

Summary

Reads an **Element** node and advances the reader to the next node.

Behaviors

As described above.

Default

This method calls the **System.Xml.XmlReader.MoveToContent** method, which determines whether the current node can contain content and, if not, moves the reader to the next content node or the end of the input stream. The node the reader ends up positioned on is checked to determine if it is an **Element** node. If so, the node is read and the reader is moved to the next node.

How and When to Override

Override this method to customize the behavior of this method in types derived from the **System.Xml.XmlReader** class.

Usage

Use this method to read an **Element** node and advance the reader to the next node.

Exceptions

Exception	Condition
System.Xml.XmlException	The node is not an **Element** node or an error occurred while parsing the XML.

XmlReader.ReadString() Method

```
[ILASM]
.method public hidebysig virtual abstract string
ReadString()

[C#]
public abstract string ReadString()
```

Summary

Reads the contents of an element or text node as a string.

Return Value

A **System.String** containing the contents of the **Element** or **Text** node, or **System.String.Empty** if the reader is positioned on any other type of node.

Behaviors

If positioned on an **Element** node, this method concatenates all **Text**, **SignificantWhitespace**, **Whitespace**, and **CDATA** node types, and returns the concatenated data as the element content. If none of these node types exist, **System.String.Empty** is returned. Concatenation stops when any markup is encountered, which can occur in a mixed content model or when an element end tag is read.

If positioned on an element **Text** node, this method performs the same concatenation from the **Text** node to the element end tag. If the reader is positioned on an attribute **Text** node, this method has the same functionality as if the reader were position on the element start tag.

How and When to Override

This method must be overridden in order to provide the functionality described above, as there is no default implementation.

Exceptions

Exception	Condition
System.Xml.XmlException	An error occurred while parsing the XML.

XmlReader.ResolveEntity() Method

```
[ILASM]
.method public hidebysig virtual abstract void
ResolveEntity()

[C#]
public abstract void ResolveEntity()
```

Summary

Resolves the entity reference for **EntityReference** nodes.

Behaviors

This method parses the entity reference into child nodes. When the parsing is finished a new **System.Xml.XmlNodeType.EndEntity** node is placed in the stream to close the **EntityReference** scope. To step into the entity after this method has been called, call the **System.Xml.XmlReader.ReadAttributeValue** method if the entity is part of an attribute value, or the **System.Xml.XmlReader.Read** method if the entity is part of element content.

If this method is not called, the parser moves to the next node past the entity (child nodes are bypassed).

How and When to Override

This method must be overridden in order to provide the functionality as described in the Behaviors and Usage sections, as there is no default implementation.

This method is required to throw an exception for implementations that do not support schema or DTD information. In this case, the **System.Xml.XmlReader.CanResolveEntity** property is required to return **false**.

Usage

Use this method to resolve the entity reference for **EntityReference** nodes. Before calling this method, determine whether the reader can resolve an entity by checking the **System.Xml.XmlReader.CanResolveEntity** property.

Exceptions

Exception	Condition
System.InvalidOperationException	The reader is not positioned on a

	System.Xml.XmlNodeType.EntityReference node.

1
2
3

XmlReader.Skip() Method

```
[ILASM]
.method public hidebysig virtual void Skip()

[C#]
public virtual void Skip()
```

Summary

Skips over the current element and moves the position of the current instance to the next node in the stream.

Behaviors

If the reader is positioned on a non-empty **Element** node (**System.Xml.XmlReader.IsEmptyElement** equals **false**), the position of the reader is moved to the node following the corresponding **EndElement** node. The properties of the nodes that are skipped over are not exposed. If the reader is positioned on any other node type, the position of the reader is moved to the next node, in this case behaving like the **System.Xml.XmlReader.Read** method.

Default

This method calls the **System.Xml.XmlReader.MoveToElement** method before skipping to the next node.

How and When to Override

Override this method to customize the behavior of this method in types derived from the **System.Xml.XmlReader** class.

Usage

Use this method to skip over the current node.

Exceptions

Exception	Condition
System.Xml.XmlException	The XML was not well-formed, or an error occurred while parsing the XML.

XmlReader.AttributeCount Property

```
[ILASM]
.property int32 AttributeCount { public hidebysig virtual
abstract specialname int32 get_AttributeCount() }

[C#]
public abstract int AttributeCount { get; }
```

Summary

Gets the number of attributes on the current node.

Property Value

A **System.Int32** containing the number of attributes on the current node, or zero if the current node does not support attributes.

Description

[*Note:* This property is only relevant to the **DocumentType**, **Element**, and **XmlDeclaration** node types of the **System.Xml.XmlNodeType** enumeration. Other node types do not have attributes.]

Behaviors

As described above.

This property is read-only.

How and When to Override

This property must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.BaseURI Property

```
[ILASM]
.property string BaseURI { public hidebysig virtual
abstract specialname string get_BaseURI() }

[C#]
public abstract string BaseURI { get; }
```

Summary

Gets the base Uniform Resource Identifier (URI) of the current node.

Property Value

The base URI of the current node.

Description

[*Note:* A networked XML document is comprised of chunks of data aggregated using various W3C standard inclusion mechanisms and therefore contains nodes that come from different places. DTD entities are an example of this, but this is not limited to DTDs. The base URI tells where these nodes come from. If there is no base URI for the nodes being returned (for example, they were parsed from an in-memory string), **System.String.Empty** is returned.]

Behaviors

As described above.

This property is read-only.

How and When to Override

This property must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.CanResolveEntity Property

```
[ILASM]
.property bool CanResolveEntity { public hidebysig virtual
specialname bool get_CanResolveEntity() }

[C#]
public virtual bool CanResolveEntity { get; }
```

Summary

Gets a value indicating whether this reader can parse and resolve entities.

Property Value

A **System.Boolean** equal to **false**.

Behaviors

This property returns **true** to indicate the reader can parse and resolve entities; otherwise, **false**.

This property is read-only.

Default

This property always returns **false**.

How and When to Override

Override this property to return **true** for implementations that support schema or DTD information.

Usage

Use this property to determine whether the reader can parse and resolve entities.

XmlReader.Depth Property

```
[ILASM]
.property int32 Depth { public hidebysig virtual abstract
specialname int32 get_Depth() }

[C#]
public abstract int Depth { get; }
```

Summary

Gets the depth of the current node in the XML document.

Property Value

A **System.Int32** containing the depth of the current node in the XML document.

Behaviors

As described above.

This property is read-only.

How and When to Override

This property must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.EOF Property

```
[ILASM]
.property bool EOF { public hidebysig virtual abstract
specialname bool get_EOF() }

[C#]
public abstract bool EOF { get; }
```

Summary

Gets a value indicating whether the **System.Xml.XmlReader.ReadState** is **System.Xml.ReadState.EndOfFile**, signifying the reader is positioned at the end of the stream.

Property Value

A **System.Boolean** where **true** indicates the reader is positioned at the end of the stream; otherwise, **false**.

Behaviors

As described above.

This property is read-only.

How and When to Override

This property must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.HasAttributes Property

```
[ILASM]
.property bool HasAttributes { public hidebysig virtual
specialname bool get_HasAttributes() }

[C#]
public virtual bool HasAttributes { get; }
```

Summary

Gets a value indicating whether the current node has any attributes.

Property Value

A **System.Boolean** where **true** indicates the current node has attributes; otherwise, **false**.

Behaviors

As described above.

This property is read-only.

Default

This property returns **true** if the **System.Xml.XmlReader.AttributeCount** property of the current node is greater than zero.

How and When to Override

Override this property to customize the behavior of this property in types derived from the **System.Xml.XmlReader** class.

Usage

Use this property to determine whether the current node has any attributes.

XmlReader.HasValue Property

```
[ILASM]
.property bool HasValue { public hidebysig virtual abstract
specialname bool get_HasValue() }

[C#]
public abstract bool HasValue { get; }
```

Summary

Gets a value indicating whether the current node can have an associated text value.

Property Value

A **System.Boolean** where **true** indicates the node on which the reader is currently positioned can have an associated text value; otherwise, **false**.

Description

[*Note:* The following members of the **System.Xml.XmlNodeType** enumeration can have an associated value: **Attribute**, **CDATA**, **Comment**, **DocumentType**, **ProcessingInstruction**, **SignificantWhitespace**, **Text**, **Whitespace**, and **XmlDeclaration**.]

Behaviors

As described above.

This property is read-only.

How and When to Override

This property must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.IsDefault Property

```
[ILASM]
.property bool IsDefault { public hidebysig virtual
abstract specialname bool get_IsDefault() }

[C#]
public abstract bool IsDefault { get; }
```

Summary

Gets a value indicating whether the current node is an attribute that was generated from the default value defined in the DTD or schema.

Property Value

A **System.Boolean** where **true** indicates the current node is an attribute whose value was generated from the default value defined in the DTD or schema; **false** indicates the attribute value was explicitly set.

Behaviors

As described above.

This property is read-only.

How and When to Override

This property should return **false** for implementations that do not support schema or DTD information.

XmlReader.IsEmptyElement Property

```
[ILASM]
.property bool IsEmptyElement { public hidebysig virtual
abstract specialname bool get_IsEmptyElement() }

[C#]
public abstract bool IsEmptyElement { get; }
```

Summary

Gets a value indicating whether the current node is an empty element (for example, `<MyElement />`).

Property Value

A **System.Boolean** where **true** indicates the current node is an element (**System.Xml.XmlReader.NodeType** equals **System.Xml.XmlNodeType.Element**) that ends with "/>", otherwise, **false**.

Behaviors

A corresponding **EndElement** node is not generated for empty elements.

This property is read-only.

How and When to Override

This property must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.Item Property

```
[ILASM]
.property string Item[int32 i] { public hidebysig virtual
abstract specialname string get_Item(int32 i) }

[C#]
public abstract string this[int i] { get; }
```

Summary

Retrieves the value of the attribute with the specified index relative to the containing element.

Parameters

Parameter	Description
i	A **System.Int32** specifying the zero-based index of the attribute relative to the containing element.

Property Value

A **System.String** containing the value of the attribute.

Behaviors

This property does not move the reader.

How and When to Override

This property must be overridden in order to provide the functionality described above, as there is no default implementation.

Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	*i* is less than 0 or greater than or equal to the **System.Xml.XmlReader.AttributeCount** of the containing element.

XmlReader.Item Property

```
[ILASM]
.property string Item[string name] { public hidebysig
virtual abstract specialname string get_Item(string name) }

[C#]
public abstract string this[string name] { get; }
```

Summary

Retrieves the value of the attribute with the specified qualified name.

Parameters

Parameter	Description
name	A **System.String** specifying the qualified name of the attribute.

Property Value

A **System.String** containing the value of the specified attribute, or **null** if the attribute is not found.

Behaviors

This property does not move the reader.

If the reader is positioned on a **DocumentType** node, this method can be used to get the PUBLIC and SYSTEM literals.

How and When to Override

This property must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.Item Property

```
[ILASM]
.property string Item[string name, string namespaceURI] {
public hidebysig virtual abstract specialname string
get_Item(string name, string namespaceURI) }

[C#]
public abstract string this[string name, string
namespaceURI] { get; }
```

Summary

Retrieves the value of the attribute with the specified local name and namespace URI.

Parameters

Parameter	Description
name	A **System.String** specifying the local name of the attribute.
namespaceURI	A **System.String** specifying the namespace URI of the attribute.

Property Value

A **System.String** containing the value of the specified attribute, or **null** if the attribute is not found.

Behaviors

This property does not move the reader.

How and When to Override

This property must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.LocalName Property

```
[ILASM]
.property string LocalName { public hidebysig virtual
abstract specialname string get_LocalName() }

[C#]
public abstract string LocalName { get; }
```

Summary

Gets the local name of the current node.

Property Value

A **System.String** containing the local name of the current node or, for node types that do not have a name (like **Text**, **Comment**, and so on), **System.String.Empty**.

Behaviors

As described above.

How and When to Override

This property must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.Name Property

```
[ILASM]
.property string Name { public hidebysig virtual abstract
specialname string get_Name() }

[C#]
public abstract string Name { get; }
```

Summary

Gets the qualified name of the current node.

Property Value

A **System.String** containing the qualified name of the current node or, for node types that do not have a name (like **Text**, **Comment**, and so on), **System.String.Empty**.

Behaviors

The qualified name is equivalent to the **System.Xml.XmlReader.LocalName** prefixed with **System.Xml.XmlReader.Prefix** and the ':' character. For example, **System.Xml.XmlReader.Name** is "bk:book" for the element <bk:book>.

The name returned is dependent on the **System.Xml.XmlReader.NodeType** of the node. The following node types return the listed values. All other node types return an empty string.

Node Type	Name
Attribute	The name of the attribute.
DocumentType	The document type name.
Element	The tag name.
EntityReference	The name of the entity referenced.
ProcessingInstruction	The target of the processing instruction.
XmlDeclaration	The literal string "xml".

This property is read-only.

How and When to Override

This property must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.NamespaceURI Property

```
[ILASM]
.property string NamespaceURI { public hidebysig virtual
abstract specialname string get_NamespaceURI() }

[C#]
public abstract string NamespaceURI { get; }
```

Summary

Gets the namespace URI associated with the node on which the reader is positioned.

Property Value

A **System.String** containing the namespace URI of the current node or, if no namespace URI is associated with the current node, **System.String.Empty**.

Behaviors

This property is relevant to **Element** and **Attribute** nodes only.

Namespaces conform to the W3C "Namespaces in XML" recommendation, REC-xml-names-19990114.

This property is read-only.

How and When to Override

This property must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.NameTable Property

```
[ILASM]
.property class System.Xml.XmlNameTable NameTable { public
hidebysig virtual abstract specialname class
System.Xml.XmlNameTable get_NameTable() }

[C#]
public abstract XmlNameTable NameTable { get; }
```

Summary

Gets the name table used by the current instance to store and look up element and attribute names, prefixes, and namespaces.

Property Value

The **System.Xml.XmlNameTable** used by the current instance.

Behaviors

Element and attribute names, prefixes, and namespaces are stored as individual **System.String** objects when a document is read.

This property is read-only.

How and When to Override

This property must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.NodeType Property

```
[ILASM]
.property valuetype System.Xml.XmlNodeType NodeType {
public hidebysig virtual abstract specialname valuetype
System.Xml.XmlNodeType get_NodeType() }

[C#]
public abstract XmlNodeType NodeType { get; }
```

Summary

Gets the type of the current node.

Property Value

One of the members of the **System.Xml.XmlNodeType** enumeration representing the type of the current node.

Behaviors

This property does not return the following **System.Xml.XmlNodeType** members: **Document**, **DocumentFragment**, **Entity**, **EndEntity**, and **Notation**.

This property is read-only.

How and When to Override

This property must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.Prefix Property

```
[ILASM]
.property string Prefix { public hidebysig virtual abstract
specialname string get_Prefix() }

[C#]
public abstract string Prefix { get; }
```

Summary

Gets the namespace prefix associated with the current node.

Property Value

A **System.String** containing the namespace prefix associated with the current node.

Description

[*Note:* A namespace prefix is used as a reference for a namespace URI and is defined in an element declaration. For example, `<someElement xmlns:bk="someURL">`, defines a prefix name "bk".]

Behaviors

As described above.

This property is read-only.

How and When to Override

This property must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.QuoteChar Property

```
[ILASM]
.property valuetype System.Char QuoteChar { public
hidebysig virtual abstract specialname valuetype
System.Char get_QuoteChar() }

[C#]
public abstract char QuoteChar { get; }
```

Summary

Gets the quotation mark character used to enclose the value of an attribute.

Property Value

A **System.Char** specifying the quotation mark character (" or ') used to enclose the value of an attribute.

Behaviors

As described above.

This property is read-only.

How and When to Override

This property must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.ReadState Property

```
[ILASM]
.property valuetype System.Xml.ReadState ReadState { public
hidebysig virtual abstract specialname valuetype
System.Xml.ReadState get_ReadState() }

[C#]
public abstract ReadState ReadState { get; }
```

Summary

Gets the read state of the reader.

Property Value

One of the members of the **System.Xml.ReadState** enumeration.

Behaviors

As described above.

This property is read-only.

How and When to Override

This property must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.Value Property

```
[ILASM]
.property string Value { public hidebysig virtual abstract
specialname string get_Value() }

[C#]
public abstract string Value { get; }
```

Summary

Gets the text value of the current node.

Property Value

A **System.String** containing the text value of the current node.

Behaviors

The value returned depends on the **System.Xml.XmlReader.NodeType**. The following table lists node types that have a value to return. All other node types return **System.String.Empty**.

Node Type	Value
Attribute	The value of the attribute.
CDATA	The content of the CDATA section.
Comment	The content of the comment.
DocumentType	The internal subset.
ProcessingInstruction	The entire content, excluding the target.
SignificantWhitespace	The white space between markup in a mixed content model, or in the scope of `xml:space = "preserve"`.
Text	The content of the text node.
Whitespace	The white space between markup.
XmlDeclaration	The content of the declaration.

This property is read-only.

How and When to Override

This property must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.XmlLang Property

```
[ILASM]
.property string XmlLang { public hidebysig virtual
abstract specialname string get_XmlLang() }

[C#]
public abstract string XmlLang { get; }
```

Summary

Gets the current `xml:lang` scope.

Property Value

A **System.String** containing the current `xml:lang` scope.

Behaviors

As described above.

This property is read-only.

How and When to Override

This property must be overridden in order to provide the functionality described above, as there is no default implementation.

XmlReader.XmlSpace Property

```
[ILASM]
.property valuetype System.Xml.XmlSpace XmlSpace { public
hidebysig virtual abstract specialname valuetype
System.Xml.XmlSpace get_XmlSpace() }

[C#]
public abstract XmlSpace XmlSpace { get; }
```

Summary

Gets the current `xml:space` scope.

Property Value

One of the members of the **System.Xml.XmlSpace** enumeration. If no `xml:space` scope exists, this property defaults to **System.Xml.XmlSpace.None**.

Behaviors

As described above.

This property is read-only.

How and When to Override

This property must be overridden in order to provide the functionality described above, as there is no default implementation.