

System.Xml.XmlNodeType Enum

```
[ILASM]
.class public sealed serializable XmlNodeType extends
System.Enum

[C#]
public enum XmlNodeType
```

Assembly Info:

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

Summary

Specifies the type of node.

Inherits From: System.Enum

Library: XML

Description

A given set of XML data is modeled as a tree of nodes. This enumeration specifies the different node types.

1 XmlNodeType.Attribute Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Xml.XmlNodeType Attribute = 2  
  
5 [C#]  
6 Attribute = 2
```

7 Summary

8 An attribute.

9
10 Example XML: id="123"

11
12 An **Attribute** node can have the following child node types: **Text** and
13 **EntityReference**. The **Attribute** node does not appear as the child
14 node of any other node type. It is not considered a child node of an
15 **Element**.

16

1 XmlNodeType.CDATA Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Xml.XmlNodeType CDATA = 4  
  
5 [C#]  
6 CDATA = 4
```

7 Summary

8 A CDATA section.

9

10 Example XML: <![CDATA[escaped text]]>

11

12 CDATA sections are used to escape blocks of text that would otherwise
13 be recognized as markup. A **CDATA** node cannot have any child
14 nodes. It can appear as the child of the **DocumentFragment**,
15 **EntityReference**, and **Element** nodes.

16

1 XmlNodeType.Comment Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Xml.XmlNodeType Comment = 8  
  
5 [C#]  
6 Comment = 8
```

7 Summary

8 A comment.

9

10 Example XML: <!-- comment -->

11

12 A **Comment** node cannot have any child nodes. It can appear as the
13 child of the **Document**, **DocumentFragment**, **Element**, and
14 **EntityReference** nodes.

15

1 XmlNodeType.Document Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Xml.XmlNodeType Document = 9  
  
5 [C#]  
6 Document = 9
```

7 Summary

8 A document object that, as the root of the document tree, provides
9 access to the entire XML document.

10
11 A **Document** node can have the following child node types:
12 **XmlDeclaration**, **Element** (maximum of one),
13 **ProcessingInstruction**, **Comment**, and **DocumentType**. It cannot
14 appear as the child of any node types.

15

1 XmlNodeType.DocumentFragment Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Xml.XmlNodeType DocumentFragment = 11  
  
5 [C#]  
6 DocumentFragment = 11
```

7 Summary

8 A document fragment.

9
10 The **DocumentFragment** node associates a node or sub-tree with a
11 document without actually being contained within the document. A
12 **DocumentFragment** node can have the following child node types:
13 **Element**, **ProcessingInstruction**, **Comment**, **Text**, **CDATA**, and
14 **EntityReference**. It cannot appear as the child of any node types.

15

1 XmlNodeType.DocumentType Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Xml.XmlNodeType DocumentType = 10  
  
5 [C#]  
6 DocumentType = 10
```

7 Summary

8 The document type declaration, indicated by the following tag.

9
10 Example XML: <!DOCTYPE...>

11
12 A **DocumentType** node can have the following child node types:
13 **Notation** and **Entity**. It can appear as the child of the **Document**
14 node.

15

1 XmlNodeType.Element Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Xml.XmlNodeType Element = 1  
  
5 [C#]  
6 Element = 1
```

7 Summary

8 An element.

9

10 Example XML: <name>

11

12 An **Element** node can have the following child node types: **Element**,
13 **Text**, **Comment**, **ProcessingInstruction**, **CDATA**, and
14 **EntityReference**. It can be the child of the **Document**,
15 **DocumentFragment**, **EntityReference**, and **Element** nodes.

16

1 XmlNodeType.EndElement Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Xml.XmlNodeType EndElement = 15  
  
5 [C#]  
6 EndElement = 15
```

7 Summary

8 An end element.

9

10 Example XML: </name>

11

12 Returned when **System.Xml.XmlReader** gets to the end of an
13 element.

14

1 XmlNodeType.EndEntity Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Xml.XmlNodeType EndEntity = 16  
  
5 [C#]  
6 EndEntity = 16
```

7 Summary

8 Returned when **System.Xml.XmlReader** gets to the end of the entity
9 replacement as a result of a call to
10 **System.Xml.XmlReader.ResolveEntity**.
11

1 XmlNodeType.Entity Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Xml.XmlNodeType Entity = 6  
  
5 [C#]  
6 Entity = 6
```

7 Summary

8 An entity declaration.

9

10 Example XML: <!ENTITY...>

11

12 An **Entity** node can have child nodes that represent the expanded
13 entity (for example, **Text** and **EntityReference** nodes). It can appear
14 as the child of the **DocumentType** node.

15

1 XmlNodeType.EntityReference Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Xml.XmlNodeType EntityReference = 5  
  
5 [C#]  
6 EntityReference = 5
```

7 Summary

8 A reference to an entity.

9

10 Example XML: #

11

12 An **EntityReference** node can have the following child node types:

13 **Element**, **ProcessingInstruction**, **Comment**, **Text**, **CDATA**, and

14 **EntityReference**. It can appear as the child of the **Attribute**,

15 **DocumentFragment**, **Element**, and **EntityReference** nodes.

16

1 XmlNodeType.None Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Xml.XmlNodeType None = 0  
  
5 [C#]  
6 None = 0
```

7 Summary

8 This is returned by the **System.Xml.XmlReader** if a read method has
9 not been called or if no more nodes are available to be read.

10

1 XmlNodeType.Notations Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Xml.XmlNodeType Notations = 12  
  
5 [C#]  
6 Notations = 12
```

7 Summary

8 A notation in the document type declaration.

9
10 Example XML: <!NOTATIONS...>

11
12 A **Notations** node cannot have any child nodes. It can appear as the
13 child of the **DocumentType** node.

14

1 XmlNodeType.ProcessingInstruction Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Xml.XmlNodeType ProcessingInstruction = 7  
  
5 [C#]  
6 ProcessingInstruction = 7
```

7 Summary

8 A processing instruction.

9
10 Example XML: <?pi test?>

11
12 A **ProcessingInstruction** node cannot have any child nodes. It can
13 appear as the child of the **Document**, **DocumentFragment**,
14 **Element**, and **EntityReference** nodes.

15

1 XmlNodeType.SignificantWhitespace Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Xml.XmlNodeType SignificantWhitespace = 14  
  
5 [C#]  
6 SignificantWhitespace = 14
```

7 Summary

8 White space between markup in a mixed content model or white space
9 within the `xml:space="preserve"` scope.

10

1 XmlNodeType.Text Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Xml.XmlNodeType Text = 3  
  
5 [C#]  
6 Text = 3
```

7 Summary

8 The text content of a node.

9
10 A **Text** node cannot have any child nodes. It can appear as the child
11 node of the **Attribute**, **DocumentFragment**, **Element**, and
12 **EntityReference** nodes.

13

1 XmlNodeType.Whitespace Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Xml.XmlNodeType Whitespace = 13  
  
5 [C#]  
6 Whitespace = 13
```

7 Summary

8 White space between markup.
9

1 XmlNodeType.XmlDeclaration Field

```
2 [ILASM]  
3 .field public static literal valuetype  
4 System.Xml.XmlNodeType XmlDeclaration = 17  
  
5 [C#]  
6 XmlDeclaration = 17
```

7 Summary

8 The XML declaration.

9
10 Example XML: <?xml version="1.0"?>

11
12 The **XmlDeclaration** node must be the first node in the document. It
13 cannot have children. It is a child of the **Document** node. It can have
14 attributes that provide version and encoding information.

15