

1 System.Xml.XmlParserContext Class

```
2 [ILAsm]  
3 .class public XmlParserContext extends System.Object  
4 [C#]  
5 public class XmlParserContext
```

6 Assembly Info:

- 7 • *Name:* System.Xml
- 8 • *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- 9 • *Version:* 2.0.x.x
- 10 • *Attributes:*
 - 11 ○ CLSCompliantAttribute(true)

12 Summary

13 Provides all the context information required by instances of the
14 System.Xml.XmlTextReader class to parse an XML fragment.

15 Inherits From: System.Object

16

17 **Library:** XML

18

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

21

1
2 **XmlParserContext(System.Xml.XmlNameTable,**
3 **System.Xml.XmlNamespaceManager,**
4 **System.String, System.Xml.XmlSpace)**
5 **Constructor**

```
6 [ILAsm]  
7 public rtspecialname specialname instance void .ctor(class  
8 System.Xml.XmlNameTable nt, class System.Xml.XmlNamespaceManager nsMgr,  
9 string xmlLang, valuetype System.Xml.XmlSpace xmlSpace)  
  
10 [C#]  
11 public XmlParserContext(XmlNameTable nt, XmlNamespaceManager nsMgr, string  
12 xmlLang, XmlSpace xmlSpace)
```

13 **Summary**

14 Constructs and initializes a new instance of the System.Xml.XmlParserContext class
15 with the specified values.

16 **Parameters**

Parameter	Description
<i>nt</i>	The System.Xml.XmlNameTable to use. If <i>nt</i> is null, this defaults to the System.Xml.XmlNameTable used to construct <i>nsMgr</i> .
<i>nsMgr</i>	The System.Xml.XmlNamespaceManager to use for looking up namespace information, or null.
<i>xmlLang</i>	A System.String specifying the xml:lang scope.
<i>xmlSpace</i>	A System.Xml.XmlSpace value indicating the xml:space scope.

17
18 **Description**

19 This method is equivalent to System.Xml.XmlParserContext(*nt*, *nsMgr*, null, null,
20 null, null, System.String.Empty, *xmlLang*, *xmlSpace*, null) constructor.

21 **Exceptions**

Exception	Condition
-----------	-----------

System.Xml.XmlException	<i>nt</i> is not the same <code>System.Xml.XmlNameTable</code> used to construct <i>nsMgr</i> .
--------------------------------	---

1

2

1
2 **XmlParserContext(System.Xml.XmlNameTable,**
3 **System.Xml.XmlNamespaceManager,**
4 **System.String, System.Xml.XmlSpace,**
5 **System.Text.Encoding) Constructor**

```
6 [ILAsm]  
7 public rtspecialname specialname instance void .ctor(class  
8 System.Xml.XmlNameTable nt, class System.Xml.XmlNamespaceManager nsMgr,  
9 string xmlLang, valuetype System.Xml.XmlSpace xmlSpace, class  
10 System.Text.Encoding enc)  
  
11 [C#]  
12 public XmlParserContext(XmlNameTable nt, XmlNamespaceManager nsMgr, string  
13 xmlLang, XmlSpace xmlSpace, Encoding enc)
```

14 **Summary**

15 Constructs and initializes a new instance of the System.Xml.XmlParserContext class
16 with the specified values.

17 **Parameters**

Parameter	Description
<i>nt</i>	The System.Xml.XmlNameTable to use. If <i>nt</i> is null, this defaults to the System.Xml.XmlNameTable used to construct <i>nsMgr</i> .
<i>nsMgr</i>	The System.Xml.XmlNamespaceManager to use for looking up namespace information, or null.
<i>xmlLang</i>	A System.String specifying the xml:lang scope.
<i>xmlSpace</i>	A System.Xml.XmlSpace value indicating the xml:space scope.
<i>enc</i>	An instance of a class derived from the System.Text.Encoding class indicating the encoding to use.

18
19 **Description**

20 This method is equivalent to System.Xml.XmlParserContext(*nt*, *nsMgr*, null, null,
21 null, null, System.String.Empty, *xmlLang*, *xmlSpace*, *enc*).

22 **Exceptions**

Exception	Condition
System.Xml.XmlException	<i>nt</i> is not the same <code>System.Xml.XmlNameTable</code> used to construct <i>nsMgr</i> .

1

2

1
2 **XmlParserContext(System.Xml.XmlNameTable,**
3 **System.Xml.XmlNamespaceManager,**
4 **System.String, System.String, System.String,**
5 **System.String, System.String, System.String,**
6 **System.Xml.XmlSpace) Constructor**

```
7 [ILAsm]  
8 public rtspecialname specialname instance void .ctor(class  
9 System.Xml.XmlNameTable nt, class System.Xml.XmlNamespaceManager nsMgr,  
10 string docTypeName, string pubId, string sysId, string internalSubset,  
11 string baseURI, string xmlLang, valuetype System.Xml.XmlSpace xmlSpace)  
  
12 [C#]  
13 public XmlParserContext(XmlNameTable nt, XmlNamespaceManager nsMgr, string  
14 docTypeName, string pubId, string sysId, string internalSubset, string  
15 baseURI, string xmlLang, XmlSpace xmlSpace)
```

16 **Summary**

17 Constructs and initializes a new instance of the System.Xml.XmlParserContext class
18 with the specified values.

19 **Parameters**

Parameter	Description
<i>nt</i>	The System.Xml.XmlNameTable to use. If <i>nt</i> is null, this defaults to the System.Xml.XmlNameTable used to construct <i>nsMgr</i> .
<i>nsMgr</i>	The System.Xml.XmlNamespaceManager to use for looking up namespace information, or null.
<i>docTypeName</i>	A System.String specifying the name of the document type declaration.
<i>pubId</i>	A System.String specifying the public identifier.
<i>sysId</i>	A System.String specifying the system identifier.
<i>internalSubset</i>	A System.String specifying the internal DTD subset.
<i>baseURI</i>	A System.String specifying the base URI for the XML fragment (the location from which the fragment was loaded).

<i>xmlLang</i>	A <code>System.String</code> containing the <code>xml:lang</code> scope.
<i>xmlSpace</i>	A <code>System.Xml.XmlSpace</code> value indicating the <code>xml:space</code> scope.

1

2 **Description**

3 This method is equivalent to `System.Xml.XmlParserContext(nt, nsMgr, docTypeName,`
 4 `pubId, sysId, internalSubset, baseUri, xmlLang, xmlSpace, null)`.

5 **Exceptions**

Exception	Condition
System.Xml.XmlException	<i>nt</i> is not the same <code>System.Xml.XmlNameTable</code> used to construct <i>nsMgr</i> .

6

7

1
2 **XmlParserContext(System.Xml.XmlNameTable,**
3 **System.Xml.XmlNamespaceManager,**
4 **System.String, System.String, System.String,**
5 **System.String, System.String, System.String,**
6 **System.Xml.XmlSpace,**
7 **System.Text.Encoding) Constructor**

```
8 [ILAsm]  
9 public rtspecialname specialname instance void .ctor(class  
10 System.Xml.XmlNameTable nt, class System.Xml.XmlNamespaceManager nsMgr,  
11 string docTypeName, string pubId, string sysId, string internalSubset,  
12 string baseURI, string xmlLang, valuetype System.Xml.XmlSpace xmlSpace,  
13 class System.Text.Encoding enc)  
  
14 [C#]  
15 public XmlParserContext(XmlNameTable nt, XmlNamespaceManager nsMgr, string  
16 docTypeName, string pubId, string sysId, string internalSubset, string  
17 baseURI, string xmlLang, XmlSpace xmlSpace, Encoding enc)
```

18 **Summary**

19 Constructs and initializes a new instance of the System.Xml.XmlParserContext class
20 with the specified values.

21 **Parameters**

Parameter	Description
<i>nt</i>	The System.Xml.XmlNameTable to use. If <i>nt</i> is null, this defaults to the System.Xml.XmlNameTable used to construct <i>nsMgr</i> .
<i>nsMgr</i>	The System.Xml.XmlNamespaceManager to use for looking up namespace information, or null.
<i>docTypeName</i>	A System.String specifying the name of the document type declaration.
<i>pubId</i>	A System.String specifying the public identifier.
<i>sysId</i>	A System.String specifying the system identifier.
<i>internalSubset</i>	A System.String specifying the internal DTD subset.

<i>baseURI</i>	A <code>System.String</code> specifying the base URI for the XML fragment (the location from which the fragment was loaded).
<i>xmlLang</i>	A <code>System.String</code> specifying the <code>xml:lang</code> scope.
<i>xmlSpace</i>	A <code>System.Xml.XmlSpace</code> value indicating the <code>xml:space</code> scope.
<i>enc</i>	The <code>System.Text.Encoding</code> to use.

1

2 Description

3 The constructor sets `System.Xml.XmlParserContext.BaseURI` to *baseURI*,
 4 `System.Xml.XmlParserContext.DocTypeName` to *docTypeName*,
 5 `System.Xml.XmlParserContext.InternalSubset` to *internalSubset*,
 6 `System.Xml.XmlParserContext.PublicId` to *pubId*,
 7 `System.Xml.XmlParserContext.SystemId` to *sysId*, and
 8 `System.Xml.XmlParserContext.XmlLang` to *xmlLang*. If null is passed for any of these
 9 parameters, the corresponding property is set to `System.String.Empty`.

10

11 [Note: The DocumentType (DTD) information stored in this constructor is ignored when
 12 an instance of the class is passed to a `System.Xml.XmlTextReader`.
 13

13

14]

15 Exceptions

Exception	Condition
System.Xml.XmlException	<i>nt</i> is not the same <code>System.Xml.XmlNameTable</code> used to construct <i>nsMgr</i> .

16

17

1 XmlParserContext.BaseURI Property

```
2  [ILAsm]
3  .property string BaseURI { public hidebysig specialname instance string
4  get_BaseURI() public hidebysig specialname instance void
5  set_BaseURI(string value) }
6
7  [C#]
8  public string BaseURI { get; set; }
```

8 Summary

9 Gets or sets the base URI.

10 Property Value

11 A System.String specifying the base URI to use for resolving the DTD file.

12 Description

13 If an attempt is made to set this property to null, it is set to System.String.Empty.

14

15 [*Note:* A networked XML document is comprised of chunks of data aggregated using
16 various W3C standard inclusion mechanisms and therefore can contain nodes that come
17 from different places. The System.Xml.XmlParserContext.BaseURI property shows
18 where these nodes originated.

19

20]

21

1 XmlParserContext.DocTypeName Property

```
2 [ILAsm]  
3 .property string DocTypeName { public hidebysig specialname instance  
4 string get_DocTypeName() public hidebysig specialname instance void  
5 set_DocTypeName(string value) }  
  
6 [C#]  
7 public string DocTypeName { get; set; }
```

8 Summary

9 Gets or sets the name of the document type in a document type declaration.

10 Property Value

11 A `System.String` specifying the name of the document type.

12 Description

13 If an attempt is made to set this property to null, it is set to `System.String.Empty`.

14
15 *[Note:* A document type declaration is of the following form:

```
16  
17 <!DOCTYPE DocTypeName PUBLIC "PublicId" "SystemId" [InternalSubset]>
```

18
19 This property, along with `System.Xml.XmlParserContext.InternalSubset`,
20 `System.Xml.XmlParserContext.PublicId`, and
21 `System.Xml.XmlParserContext.SystemId` properties, provide all the document type
22 declaration information.

```
23  
24 ]
```

25

1 XmlParserContext.Encoding Property

```
2 [ILAsm]
3 .property class System.Text.Encoding Encoding { public hidebysig
4 specialname instance class System.Text.Encoding get_Encoding() public
5 hidebysig specialname instance void set_Encoding(class
6 System.Text.Encoding value) }
7
8 [C#]
9 public Encoding Encoding { get; set; }
```

9 Summary

10 Gets or sets the encoding type.

11 Property Value

12 A System.Text.Encoding indicating the encoding type.

13

1 XmlParserContext.InternalSubset Property

```
2 [ILAsm]  
3 .property string InternalSubset { public hidebysig specialname instance  
4 string get_InternalSubset() public hidebysig specialname instance void  
5 set_InternalSubset(string value) }  
  
6 [C#]  
7 public string InternalSubset { get; set; }
```

8 Summary

9 Gets or sets the internal subset in a document type declaration.

10 Property Value

11 A System.String specifying the internal subset.

12 Description

13 If an attempt is made to set this property to null, it is set to System.String.Empty.

14
15 [Note: A document type declaration is of the following form:

```
16  
17 <!DOCTYPE DocTypeName PUBLIC "PublicId" "SystemId" [InternalSubset]>
```

18
19 This property, along with System.Xml.XmlParserContext.DocTypeName,
20 System.Xml.XmlParserContext.PublicId, and
21 System.Xml.XmlParserContext.SystemId properties, provide all the document type
22 declaration information.

```
23  
24 ]
```

25

1 XmlParserContext.NamespaceManager

2 Property

```
3 [ILAsm]
4 .property class System.Xml.XmlNamespaceManager NamespaceManager { public
5 hidebysig specialname instance class System.Xml.XmlNamespaceManager
6 get_NamespaceManager() public hidebysig specialname instance void
7 set_NamespaceManager(class System.Xml.XmlNamespaceManager value) }

8 [C#]
9 public XmlNamespaceManager NamespaceManager { get; set; }
```

10 Summary

11 Gets or sets the System.Xml.XmlNamespaceManager used by the current instance.

12 Property Value

13 The System.Xml.XmlNamespaceManager used by the current instance.

14 Description

15 [*Note:* A System.Xml.XmlNamespaceManager defines the current namespace scope and
16 provides methods for looking up namespace information.

17]
18]

19

1 XmlParserContext.NameTable Property

```
2 [ILAsm]  
3 .property class System.Xml.XmlNameTable NameTable { public hidebysig  
4 specialname instance class System.Xml.XmlNameTable get_NameTable() public  
5 hidebysig specialname instance void set_NameTable(class  
6 System.Xml.XmlNameTable value) }  
  
7 [C#]  
8 public XmlNameTable NameTable { get; set; }
```

9 Summary

10 Gets or sets the `System.Xml.XmlNameTable` used by the current instance to look up
11 prefixes and namespace URIs.

12 Property Value

13 The `System.Xml.XmlNameTable` used by the current instance.

14

1 XmlParserContext.PublicId Property

```
2 [ILAsm]  
3 .property string PublicId { public hidebysig specialname instance string  
4 get_PublicId() public hidebysig specialname instance void  
5 set_PublicId(string value) }  
  
6 [C#]  
7 public string PublicId { get; set; }
```

8 Summary

9 Gets or sets the public identifier in a document type declaration.

10 Property Value

11 A System.String specifying the public identifier.

12 Description

13 If an attempt is made to set this property to null, it is set to System.String.Empty.

14
15 [Note: A document type declaration is of the following form:

16
17 <!DOCTYPE DocTypeName PUBLIC "*PublicId*" "*SystemId*" [InternalSubset]>

18
19 This property, along with System.Xml.XmlParserContext.DocTypeName,
20 System.Xml.XmlParserContext.InternalSubset, and
21 System.Xml.XmlParserContext.SystemId properties, provide all the document type
22 declaration information.

23
24]

25

1 XmlParserContext.SystemId Property

```
2 [ILAsm]  
3 .property string SystemId { public hidebysig specialname instance string  
4 get_SystemId() public hidebysig specialname instance void  
5 set_SystemId(string value) }  
  
6 [C#]  
7 public string SystemId { get; set; }
```

8 Summary

9 Gets or sets the system identifier in a document type declaration.

10 Property Value

11 A System.String specifying the system identifier.

12 Description

13 If an attempt is made to set this property to null, it is set to System.String.Empty.

14
15 [Note: A document type declaration is of the following form:

```
16  
17 <!DOCTYPE DocTypeName PUBLIC "PublicId" "SystemId" [InternalSubset]>
```

18
19 This property, along with System.Xml.XmlParserContext.DocTypeName,
20 System.Xml.XmlParserContext.InternalSubset, and
21 System.Xml.XmlParserContext.PublicId properties, provide all the document type
22 declaration information.

```
23  
24 ]
```

25

1 XmlParserContext.XmlLang Property

```
2 [ILAsm]  
3 .property string XmlLang { public hidebysig specialname instance string  
4 get_XmlLang() public hidebysig specialname instance void  
5 set_XmlLang(string value) }  
  
6 [C#]  
7 public string XmlLang { get; set; }
```

8 Summary

9 Gets or sets the current `xml:lang` scope.

10 Property Value

11 A `System.String` specifying the current `xml:lang` scope. If there is no `xml:lang` in
12 scope, `System.String.Empty` is returned.

13 Description

14 If an attempt is made to set this property to null, it is set to `System.String.Empty`.

15
16 [*Note:* The language attribute, `xml:lang`, specifies the language in which the content
17 and attribute values of the current element are written.

18
19 For details on valid `xml:lang` values, refer to section 2.12 of the W3C Extensible Markup
20 Language (XML) 1.0 recommendation.

21
22]

23

1 XmlParserContext.XmlSpace Property

```
2 [ILAsm]
3 .property valuetype System.Xml.XmlSpace XmlSpace { public hidebysig
4 specialname instance valuetype System.Xml.XmlSpace get_XmlSpace() public
5 hidebysig specialname instance void set_XmlSpace(valuetype
6 System.Xml.XmlSpace value) }
7 [C#]
8 public XmlSpace XmlSpace { get; set; }
```

9 Summary

10 Gets or sets the current `xml:space` scope.

11 Property Value

12 One of the members of the `System.Xml.XmlSpace` enumeration indicating the
13 `xml:space` scope.

14 Description

15 [*Note:* The white space attribute, `xml:space`, specifies how white space is handled in
16 the current element.

17]
18]

19