



- ▢ [MSDN Library](#)
- ↑ [Windows Development](#)
- ↑ [Data Access and Storage](#)
- ↑ [MSXML](#)
- ↑ [XML for MSXML](#)
- ↑ [XML Concepts \[MSXML\]](#)
- ▢ [XML Data Islands](#)

XML Data Islands



[This feature was implemented for MSXML 3.0. Later versions of MSXML might not support it.]

When XML was first gaining acceptance, there was an increasing need to be able to embed "islands" of data inside HTML pages. In Microsoft® Internet Explorer 5.0 and later, these data islands can be written in XML.

The following topics describe the syntax used for embedding these data islands within a page, and detail the object model exposed by the browser to enable them to be used. This method of embedding XML in HTML follows the note published by the World Wide Web Consortium (W3C) as the "XML in HTML Meeting Report." The W3C expects to evolve the HTML specification to include the capability of embedding XML in HTML documents.

Embedding an XML Data Island into an HTML Page

An XML data island can be embedded using one of the following methods.

- Using the Dynamic HTML (DHTML) <XML> element within the HTML document
- Overloading the HTML <SCRIPT> element

Using the XML Element Within the HTML Document

This syntax is valid for Internet Explorer 5.0.

There are two syntactically correct ways of using the <XML> element within the HTML document.

- The XML data can exist inline, surrounded by <XML></XML> start and end tags.

```
<XML ID="XMLID">
  <XMLDATA>
    <DATA>TEXT</DATA>
  </XMLDATA>
</XML>
```

[Copy](#)

- The <XML> element can have a SRC attribute, the value of which is the URL for an XML data source.

[Conv](#)

```
<XML SRC="http://localhost/xmlFile.xml"></XML>
```

The `<XML>` element is present in the HTML Document Object Model. It is in the `DHTML.all` collection and is seen by the browser as just a regular node. The XML data within the `<XML>` element can then be accessed by calling the `DHTML.XMLDocument` property on the `<XML>` element.

The `XMLDocument` property returns the root node of the XML within the `<XML>` element or the root node of the XML referenced by the value of the `SRC` attribute. From this root, the XML data island can be navigated using the XML Document Object Model (DOM). The following function returns the data from the data island with the ID of "XMLID".

J#[Copy](#)

```
function returnXMLData(){
    return document.all("XMLID").XMLDocument.
    nodeValue;
}
```

The `<XML>` element can also be referenced by ID alone. For example, the following function has the identical functionality as the preceding example.

J#[Copy](#)

```
function returnXMLData(){
    return XMLID.documentElement.text;
}
```

Because the `XMLDocument` property was not used, the [documentElement property](#) must be called to retrieve the root element of the XML.

Overloading the HTML `<SCRIPT>` Element

This syntax has been deprecated and is intended only for down-level cases.

There are three syntactically correct ways of overloading the HTML `<SCRIPT>` element.

- The `LANGUAGE` attribute can be given the value "XML".
`<SCRIPT LANGUAGE="XML">`
- The `TYPE` attribute can be given the value "text/xml".
`<SCRIPT TYPE="text/xml">`
- As with the `<XML>` element, a `SRC` attribute can be added, the value of which is the URL for an XML data source.

```
<SCRIPT LANGUAGE="XML"
SRC="http://localhost/xmlFile.xml"></SCRIPT>
```

The following HTML fragment illustrates how to embed data by overloading the <SCRIPT> element.

```
<SCRIPT ID="XMLID" LANGUAGE="XML">
  <XMLDATA>
    <DATA>TEXT</DATA>
  </XMLDATA>
</SCRIPT>
```

[Copy](#)

The <SCRIPT> element is present in the HTML page's object model. (It is in the DHTML all collection and is seen by the browser as a regular script node.) The XML data within the <SCRIPT> elements can be accessed by calling the XMLDocument property on the <SCRIPT> object.

The following script accesses the XML data island in the preceding HTML fragment, and returns the name of the root node of the XML data island.

J#

```
function returnIslandRootName(){
  var islandRoot = document.all("SCRIPT").
XMLDocument;
  return islandRoot.nodeName;
}
```

[Copy](#)**Note**

A tag that uses the name "XML" cannot be nested within an XML data island.