# Comparing the XML DOM and WinRT XML APIs

Processing XML data is a common activity for web applications and will likely be employed by many Windows Store apps. For apps written in HTML, CSS, and JavaScript, two APIs are available: the XML DOM API represented by the [*DOMParser*](http://msdn.microsoft.com/en-us/library/windows/apps/hh770805.aspx) object, and the WinRT API in the [*Windows.Data.Xml.Dom*](http://msdn.microsoft.com/en-us/library/windows/apps/windows.data.xml.dom.aspx) namespace. This post will make a brief comparison between the two.

*Note*: one key difference for any comparison with WinRT APIs is that HTML and native JavaScript APIs are available in the web context whereas WinRT is not. This comparison therefore only applies to pages running in the local context. In the web context you must use the HTML5/JS APIs.

It’s also helpful to note that overlaps such as these exist (a) because of the local and web contexts, and (b) to provide the capability to Windows Store apps written in other languages where an intrinsic API is not available.

**XML DOM API**

The [XML DOM API](http://www.w3schools.com/dom/default.asp) is already familiar to JavaScript developers, because the HTML DOM API is really just a specific flavor of the XML API. Functions like *getElementById*, for example, which every JavaScript developer can probably type in their sleep, is shared between both. The XML DOM API is compliant with the HTML5 [Web DOM Core](http://www.w3.org/TR/2010/WD-domcore-20101007/) spec.

This API is the most extensive where manipulating XML is concerned. Converting XML text to an XML document is done with a simple call to *DOMParser.parse*; the API is then available on that document object.

As far as drawbacks are concerned, this API is synchronous and entirely based on having the XML as a string. No provision is made for loading the XML from another source. It also does not support XSLT or XPath.

For details on the implementation of *DOMParser* for Windows Store apps, the documentation is [here](http://msdn.microsoft.com/en-us/library/windows/apps/hh770805.aspx).

**WinRT API: Windows.Data.Xml.Dom**

Generally speaking, the API available for an XML document in WinRT, [*Windows.Data.Xml.Dom*](http://msdn.microsoft.com/en-us/library/windows/apps/windows.data.xml.dom.aspx), is very similar to the XML DOM API. It is compliant with the W3C DOM Core specification.

In its design, many method names match the DOM versions, such as *getElementById*. That said, the WinRT API isn’t quite as extensive as the DOM API. Compare, for example, the XML DOM’s API on an [element](http://www.w3schools.com/Dom/dom_element.asp) to *IXmlElement*. Similarly, *IXmlText* in WinRT has a single method, *SplitText*, while the XML DOM [Text object](http://www.w3schools.com/Dom/dom_text.asp) has quite a few more.

The WinRT API provides other functionality, however. It provides XSLT and XPath support, and can, in addition to synchronous parsing of in-memory XML, asynchronously load XML from a file or remote URL. An XML document object obtained from WinRT can also be used with other WinRT APIs.

**Recommendations**

In the end, the specific differences between these APIs will determine what’s best for you:

1. **By default, use XML DOM API (DOMParser) in an app.** It is the web-friendliest API, works in the web compartment, and covers the majority of web scenarios.
2. **Use XML WinRT API where XML DOM API is insufficient.** It is necessary if you need to use XSLT or XPath, need to pass XML objects to WinRT APIs, require async loading of XML from a file or remote URL, or if compliance with the W3C DOM Core spec is required.