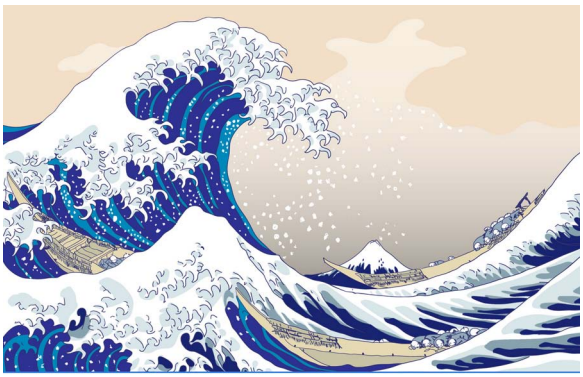
Catching the XML Wave

How I used BBjCustomObjects to Utilize XML

By Brian Hipple

As a BBx® Web services developer, I have longed for XML support in the BASIS Products Suite. BBj® does not directly support XML, but it does support embedded Java code and the many Java XML packages that create and manipulate XML documents. Therefore, the great news is that I, or any BASIS developer, can create BBj Custom Objects to simplify the rather complex interaction with Java's XML packages, and extend functionality not provided by the Java implementation. Any BBj application can then use these custom objects in a traditional fashion.



What is XML?

XML, or E**X**tensible **M**arkup **L**anguage, is a cross-platform, software- and hardware- independent tool for transmitting information. XML is much like the familiar HTML, **H**yper **T**ext **M**arkup **L**anguage, although XML's main function is to describe the data rather than format and display data as HTML does. Both languages use tags, which are keywords surround by < > to convey information for the data. The tags used in HTML documents are predefined and the creator of an HTML documents can only use tags that are defined in the HTML standard (like <p>, <h1>, etc.). In XML, the tags are not predefined; the document author must define them. **Figure 1** is an example of an XML document in the BASIS IDE XML Editor.

Why use XML?

XML technology allows developers to future-proof their applications and services, ensuring that the data they are managing today will easily adapt to future needs. With XML, developers can transform today's data into new data formats as they emerge or build new applications and services to accomplish new tasks with existing data. XML also serves as a common platform for transmitting and sharing data between disparate systems, allowing the rapid development of Web services that query, retrieve, and share data among many sources. (http://www.sitepoint.com/books/xml1/)

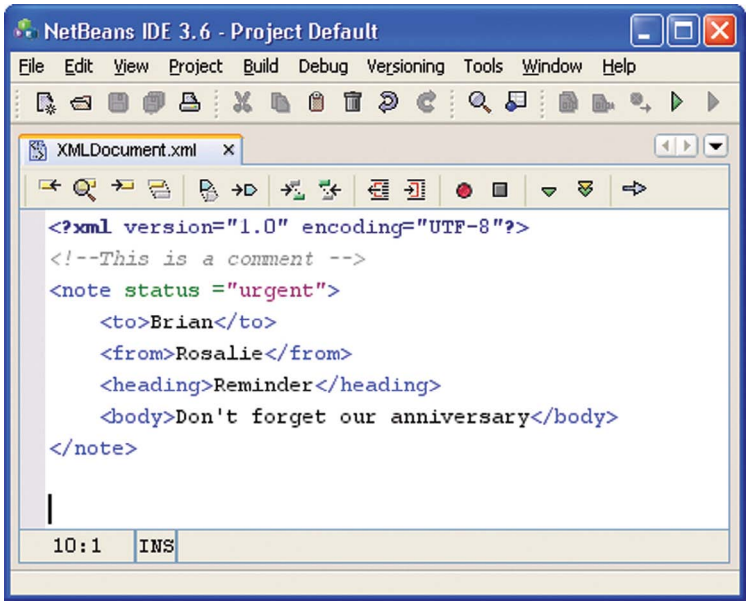


Figure 1. XML document

XML is not just a universal data format; it is also a universal library of tools that includes XSLT, XPath, XQuery, and DOM. Developers use these tools for transforming documents between otherwise incompatible formats, presenting data in particular styles and formats, querying data from data sources, and manipulating data in a hierarchical, tree-like form. XML standards are implemented in every major programming language, ensuring developers that they can always access their most important asset in the future – their data. (http://www.sitepoint.com/books/xml1/)

Why use BBj Custom Objects?

BBj version 6.0 introduced BBj Custom Objects that enable developers to make use of object-oriented programming. My decision to use a BBj Custom Object for my XML functionality enabled me to reap the benefits these objects provide: increased development resources, maintainability, readability, and reusability. I was able to streamline functionality into the most commonly-used methods; therefore, a BBj developer should be able to easily decipher the methods.

continued...

Reusability is a major advantage of using a BBj Custom Object. Once I thoroughly tested my object, any BBj application can now use this object to access XML documents with the assurance of quality. Incorporating this functionality separately into each application would have required a great deal more coding and testing.

The Custom Object - XMLDoc

The resultant BBj Custom Object name is XMLDoc, which includes multiple constructors that provide for the creation of an XMLDoc object from a new document, an existing document, a string, or from a specified URL. The XMLDoc methods provide many commonly used XML functions such as addRootElement(), addElement(), setAttribute(), addTextNode(), and addCommentNode(). Along with these methods are the combination of these methods for a macro-type interface which are not provided in the Java XML APIs, that includes addElementWithAttribute(), addElementWithAttributes(), and addElementWithTextNode(). Search and destroy methods getAllMatchingNodes() and removeAllMatchingNodes() will search or remove nodes based on criteria such as a "node name" and "type of node." Also, the output method writeToConsole() can write the XMLDoc to the BBj console for debugging purposes and writeToFile() writes the contents of the XMLDoc to a new file. **Figure 2** shows the XMLDoc source that employs these methods.

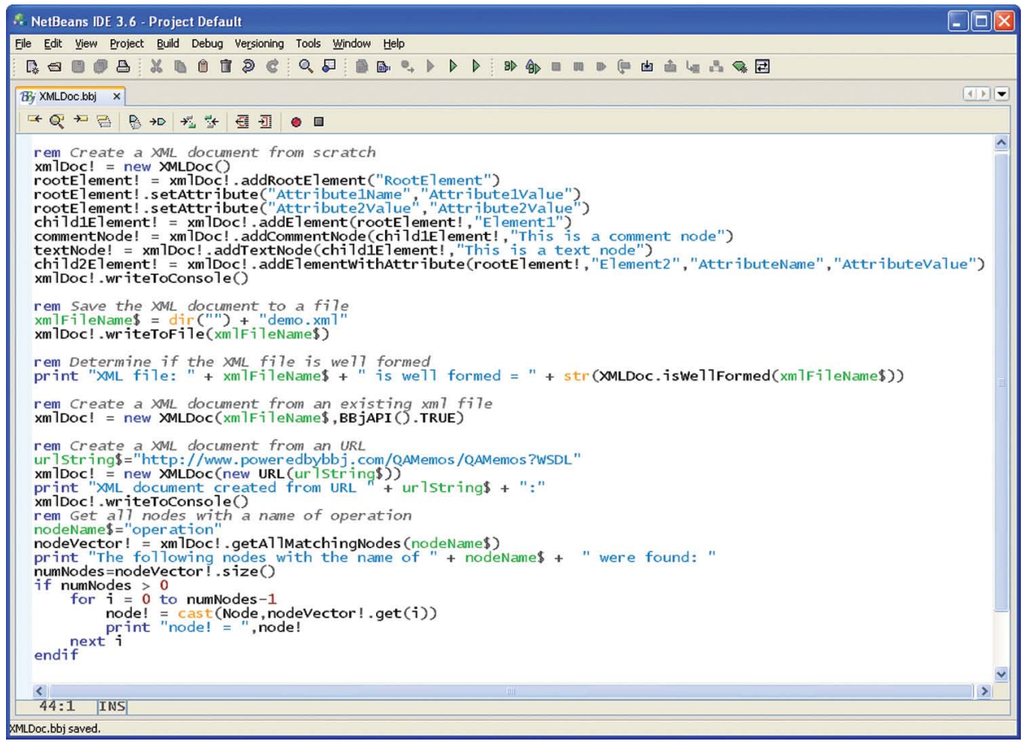


Figure 2. XMLDoc

Using the XMLDoc Object

To exercise the XMLDoc object in an application, I wrote a small BBx program named WeatherXMLDemo.bbj, which passes the URL for a weather site with parameter information that requests the seven-day forecast in XML format for a city when the user enters a zip code

This creates an instance of an XMLDoc that can then retrieve the forecast data and display this information in a grid. See WeatherDemo code in **Figure 3** and the results in **Figure 4**.

This program shows just how easy it is to retrieve information from the internet in XML - which many Web services provide - and use this data in your application. Simple applications like this weather retrieval program demonstrate the use of XML to transfer information over the web, but it does not end there. XML is highly adaptable and works well for applications with widely differing degrees of complexity.

continued...

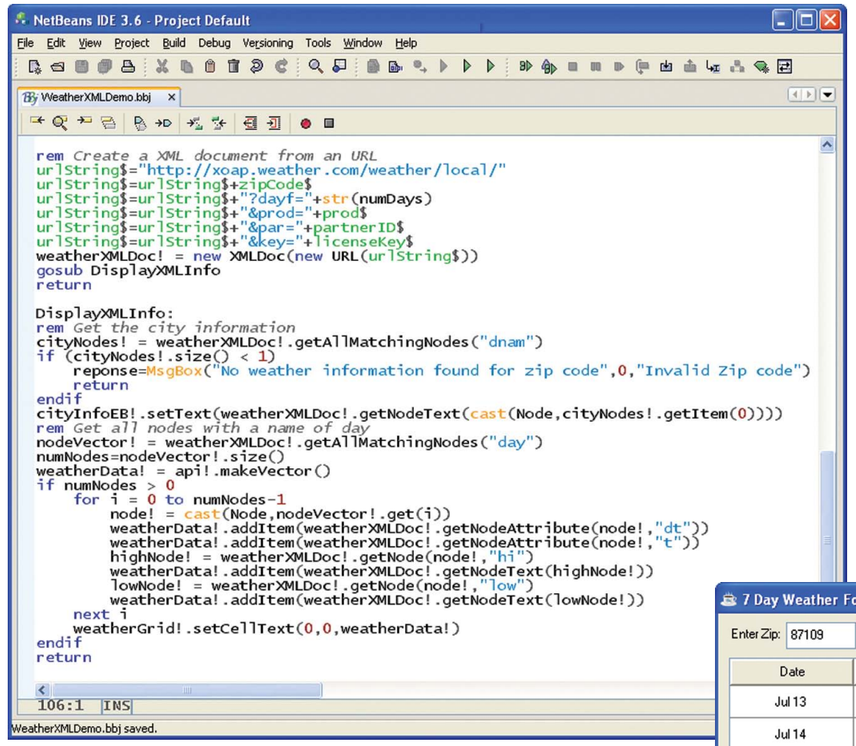


Figure 3. The WeatherXMLDemo source code

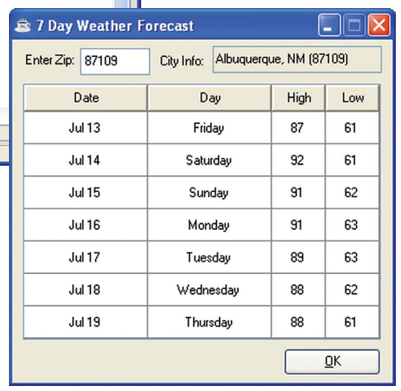


Figure 4. The WeatherXMLDemo application

Recently, I consulted with a BASIS customer on their Ford Motor Company Web service that transferred all the information in XML, including parameters for all the methods. It was evident to me they chose XML for future compatibility. For example, if one of the Web service methods should require more information, the XML specification will change for the method, but the method signature does not have to change. As a result, this customer would not need to make any changes to the client application.

Summary

It amazes me that XML has become the standard data format for transferring information over the internet in such a relatively short period or time. While the BBj Custom Object that I created provides basic XML functionality, my hope is that over time I, or another BBx developer, will take this object and extend its functionality. For example, we could incorporate the ability for a DTD or XML schema to describe the XML in detail and validate the XML, or the ability to use XML Namespaces to avoid element name conflicts by providing qualifications in the XML documents. Using Custom Objects in BBj with the strength of the Java XML API gives BBx developers the ability to harness this data standard and provide information that will thrive and prosper for many years to come. XML puts your data and application in the fast lane!

A Primer for Using BBj Custom Objects *By David Wallwork*
www.basis.com/advantage/mag-v10n1/primer.pdf

Applying Custom Objects to Existing Code *By Brian Hipple*
www.basis.com/advantage/mag-v10n1/custom.pdf

Download and run the sample referenced in this article at
www.basis.com/advantage/mag-v11n1/XML.zip