



Business To Manufacturing Markup Language

Batch Markup Language

Version 0500 – March 2011

Code Generation Notes



IMPORTANT: While the information, data, and standards provided in this publication were developed and are presented in good faith in accordance with a reasonable process that was subject to intellectual property and antitrust policies to benefit the industry as a whole, the publication is provided “as is” for information and guidance only, and there is no representation or warranty of any type or kind, including but not limited to warranties of merchantability or fitness for a particular purpose, and no warranty that use of the information, data, or standards will not infringe patent, copyright, trademark, trade secret, or other intellectual property rights of any party.

Table of Contents

Contents

Table of Contents	2
Code Generation Notes	2
WebMethods	2
MS Visual Studio	3
XSD.EXE	3
File Name Too Long	3
CodeXS	4
XMLBeans, JAXP, Xerces	4
BizTalk 2006	4

Code Generation Notes

B2MML and BatchML are used in a wide variety of code generation tools. Experience shows that *namespace and include/imports* are treated differently by different tools. The current model used in B2MML & BatchML was developed to accommodate as many of the more common tools as possible. It appears that no one *namespace and include/imports* model will work well with all tools. The WBF recognizes that some changes may be needed to the schemas for them to work in certain environments.

This document has been created to allow users to document changes to the schemas required for specific environments. This document will be updated as input is received for each release.

The following notes on code generation using B2MML V0500 and BatchML V0500 are from members of the WBF XML committee members.

WebMethods

In our environment, we use WebMethods and many developers use OxygenXML or XMLSpy.

One challenge we discovered is when developers attempt to extend files individually, the broader impact/use across the entire model is not well understood. To this end, in other functional areas, we have use a logical data model (UML equivalent) where we maintain business concepts and definitions can maintain the relationship between XMLSchemas and the model.

MS Visual Studio

For the test I only did a validity check with MS Visual Studio .net. Valid XSD's are a prerequisite for code-generation.

I have asked my colleagues and they are satisfied with the C#-code generation from the B2MML-xsd's using MS Visual Studio .net.

The only remark they have is that the "actioncode", in the data area of the message erroneously is being translated by MS Visual Studio in an array type.

They have to manually change it in the C#-code every time that the classes are generated, which is once per B2MML release.

The problem occurred in the ProcessProductionSchedule scheme and any other scheme that contained the actioncode.

XSD.EXE

I was able to generate classes in .NET using xsd.exe based on this blog:

<http://www.west-wind.com/WebLog/posts/10402.aspx>

re: XSD.EXE and included schemas?

by Lukash September 11, 2008 @ 2:51 am

I had the same kind of issue so I implemented a custom tool linked to XSD that just loops inside the XSD file and retrieve all the sub xsd file imported inot the imported or included sub xsd files... and then call xsd with the list of sub xsd files :

```
xsd.exe sub1.xsd sub2.xsd sub3.xsd main.xsd...
```

One of the main benefits of V0500 was the incorporation of the BatchML schemas. However, the list of schema dependencies also increased. Depending on future developments, the maximum file name length limitation could become an issue.

One annoying thing about xsd is that it doesn't allow you to specify the output file name, just the directory. Since all of the imported and included schemas have to be passed as command line parameters, the generated file name is a concatenation of the entire schema file name delimited with an underscore.

File Name Too Long

The filename becomes too long when using the xsd.exe tool.

There appears to be a work-around for this though, which I came across browsing Stackoverflow (<http://stackoverflow.com/questions/906093/xsd-exe-output-filename>).

To apply this method use the following method:

- 1) Use the AllSchema.xsd file included with the B2MML release.

- 2) Run the following command (pay attention to the path character in the last filename):

```
xsd AllSchema.xsd /c
```

The resulting file generated is named AllSchema.cs

The AllSchema.xsd file includes all of the B2MML schemas definitions.

The only purpose of the "AllSchemas.xsd" file is to get a friendly filename.

CodeXS

I still can't get the new schemas to generate using CodeXS, but as long as there is at least one tool that works with .NET, this is no longer a critical issue.

XMLBeans, JAXP, Xerces

I haven't noticed many (any?) comments on testing with Java tools. Our entire internal corporate web service infrastructure is based on Java.

My coworker has done testing using several Java tools. He was able to use the schemas with XMLBeans 2.3.0.

However, he had various issues using JAXP 1.3.x inside JDK 1.5, JAXP 1.4.x inside JDK 1.6, and Xerces j 2.9.1

BizTalk 2006

To make a valid BizTalk project remove the “include of

B2MML-V0500-CoreComponents.xsd

from

B2MML-V0500-CommonExtensions.xsd

This does not work in BizTalk because B2MML-V0500-CoreComponents.xsd is being included in different contexts (targetNamespace).

It's included in B2MML-V0500-CommonExtensions.xsd where it becomes part of <http://www.wbf.org/xml/B2MML-V0500-AllExtensions>. Then subsequently in B2MML-V0500-Common.xsd, and others, as part of <http://www.wbf.org/xml/B2MML-V0500>.

This is because it has no targetNamespace of its own so once it inherits *-AllExtension first, the subsequent schemas complain because the namespace is now different.

The **B2MML-V0500-CommonExtensions.xsd** file is designed to be user editable (it is where you put extensions to the common types). So this is one of the 2 files that users are encouraged to change as necessary.

This change does prevent the use of the core component types in extensions to the common types (Value, Result, Parameter, ...). However, this may have a minor impact

because most user extensions are in the complex types defined in the **B2MML-V0500-Extensions.xsd** file (ProductionSchedule, MaterialLot, MaterialActual, ...).