


xmlToClass Blobs

Syntax	<pre>set anObject = aBlob.xmlToClass();</pre>	
Semantics	<p>The operation takes an XML buffer as blob (<code>aBlob</code>) and tries to map the XML document to <code>anObject</code>. If this is not possible, an error is raised (e.g. XML parser errors, invalid mappings, etc.).</p> <p>By default the following mapping rules apply:</p> <ul style="list-style-type: none">• Class attributes are mapped to XML attributes.• Association ends are mapped to XML elements. <p>These default rules can be overridden by using the stereotypes XMLElement, XMLAttribute, and XMLCharacters on class properties. More about these mapping rules, stereotypes and tagged values (e.g. for number and date & time formatting) can be found on Controlling the XML Serialization With Stereotypes.</p> <div> Frequently, an XML document is given as a string instead of a blob. In such cases, it is possible to apply <code>xmlToClass()</code> to a string as well. For details see xmlToClass.</div>	
Substitutables	aB lob	Can be any variable or object attribute having the type Blob .
	an Ob je ct	Target object, can be any object.
Error Codes	Find the related error codes on Log Errors of domain XMLLM .	
	XM LL M /3	The native XML parser reported a fatal error during parsing. Read more on Troubleshooting the XML Serialization for more information if you cannot resolve such errors.

On this Page:

- [XML Parsing Options \(Validation against a Schema\)](#)

Related Pages:

- [xmlToClass\(\) Operation for Strings](#)
- [XML Serialization](#)
- [Troubleshooting the XML Serialization](#)
- [Log Errors of domain XMLLM](#)

Examples

The action script below creates an object of type **Address**. An output object named **myAddress** of type **Address** needs to be defined.

```
create myAddress;
set myAddress = addressAsXMLDocument.xmlToClass();
```

Beneath, a sample XML document is shown to illustrate the executed mapping. The XML document is mapped to an instance of **Address** as shown in the class diagram.

XML Source	Target Class Structure
<pre><myAddress id="myAddressID"> <street>108, Kearny Avenue</street> <city>Newark</city> </myAddress></pre>	<pre>classDiagram class Address { +id : String +city : String +street : String } class String1 class String2 Address --> String1 : +city Address --> String2 : +street</pre>

Note, that the XML element **myAddress** is mapped to the object **myAddress**, which is of type **Address**. This type has the UML attribute **id** which corresponds to the XML attribute **id**. Additionally, the XML elements **street** and **city** are mapped to the association ends **city** respectively **street**. Both are having the type **String**.

XML Parsing Options (Validation against a Schema)

`xmlToClass()` offers an optional parameter of type **XMLParseOptions**. It has various attributes to control schema and DTD location and validation:

XMLParseOptions
+disableDefaultEntityResolution : Boolean +entityExpansionLimit : Integer +externalNoNamespaceSchemaLocation : String +externalSchemaLocation : String +namespacePrefixes : Boolean +namespaces : Boolean +nonvalidatingLoadExternalDTD : Boolean +scannerName : String +standardURIconformant : Boolean +validation : Boolean +validationDynamic : Boolean +validationIdentityConstraintChecking : Boolean +validationSchema : Boolean +validationSchemaFullChecking : Boolean +validationSchemaSkipDTWValidation : Boolean

Be aware that by default, schemas are parsed, but documents are not validated against them. Set the **validation** attribute to **true** if you want to enforce validation beyond well-formedness.

For example, assume that you want to validate your document against a schema called **CustomerData.xsd** having the namespace **http://acme.com/customer**. In this case, you need to set the following parse options:

Option	Value
externalSchemaLocation	http://acme.com/customer


validation	true
------------	------

If the XML document refers a schema file (.xsd) with filename (and optional path), it is sufficient to upload the file to the Integration (Bridge) as a resource. It will be automatically loaded from there.

If the XML document refers no schema, or you would like to provide another than the referred one, use the **XMLParseOptions** structure and set **externalSchemaLocation** (or **externalNonamespaceSchemaLocation**) accordingly. As above, missing or relative paths will be redirected to the resource folder of the Integration (Bridge). The same applies also for validating against DTDs. Be aware that validation is turned off by default (see option **validation** below).

The following table lists all available XML options. Default values used when an option is not explicitly set are written in bold. The Runtime uses the Xerces parser internally, so you can find more information for all options on the Xerces home page by following the link in the Xerces column.

Parse Option	Description	Xerces Link	Values	
validation	Controls validation.	Xerces Documentation	true	Report all validation errors. The document must specify a grammar in this case. This option overrides nonvalidatingLoadExternalDTD .
			false	Do not report validation errors (default). If the document specifies a grammar, that grammar might be parsed but no validation of the document contents will be performed.
validationDynamic	Validate the document if a grammar is specified.	Xerces Documentation	true	The parser will validate the document only if a grammar is specified. (validation must be true).
			false	Validation is determined by the state of the validation option (default).
validationSchema	Control schema support.	Xerces Documentation	true	Enable the parser's schema support (default). To use this option, namespaces must also be turned on.
			false	Disable the parser's schema support.
validationSchemaFullChecking	Enable checking the schema grammar itself for additional errors that are time-consuming or memory intensive. It does not affect the level of checking performed on document instances that use schema grammars.	Xerces Documentation	true	Enable full schema constraint checking, including checks that may be time-consuming or memory intensive. Currently, particle unique attribution constraint checking and particle derivation restriction checking are controlled by this option.
			false	Disable full schema constraint checking (default).
nonvalidatingLoadExternalDTD	Controls loading an external DTD. This feature is ignored and DTD is always loaded when the option validation is true .	Xerces Documentation	true	Load external DTD.
			false	Ignore external DTD completely (default).
standardURIConformant	Controls standard URI checks.	Xerces Documentation	true	Force standard URI conformance. Malformed URIs will be rejected.
			false	Do not force standard URI conformance (default).
validationIdentityConstraintChecking	Controls entity constraint checking.	Xerces Documentation	true	Enable identity constraint checking (default).
			false	Disable identity constraint checking.
validationSchemaSkipDTDValidation	Controls usage of DTDs.	Xerces Documentation	true	When validationSchema is true the parser will ignore the DTD, except for entities.
			false	The parser will not ignore DTDs when validating (default).
disableDefaultEntityResolution	Controls entity resolution.	Xerces Documentation	true	The parser will not attempt to resolve the entity if the Runtime can't find it.
			false	The parser will attempt to resolve the entity on its own if the Runtime can't find it (default).
namespaces	Controls namespace processing.	Xerces Documentation	true	Perform namespace processing (default).

	If the validation option is set to true , then the document must contain a grammar that supports the use of namespaces.		fa lse	Do not perform namespace processing.
namespacePrefixes	Controls reporting of namespace prefixes.	Xerces Documentation	tr ue fa lse	Report the original prefixed names and attributes used for namespace declarations. Do not report attributes used for namespace declarations, and optionally do not report original prefixed names (default).
externalSchemaLocation	<p>The XML Schema Recommendation explicitly states that the inclusion of schemaLocation /nonamespaceSchemaLocation attributes in the instance document is only a hint; it does not mandate that these attributes must be used to locate schemas.</p> <p>Similar situation happens to <import> element in schema documents. This property allows the user to specify a list of schemas to use. If the targetnamespace of a schema specified using this method matches the targetnamespace of a schema occurring in the instance document in schemaLocation attribute, or if the targetnamespace matches the namespace attribute of <import> element, the schema specified by the user using this property will be used (i.e., the schemaLocation attribute in the instance document or on the <import> element will be effectively ignored).</p>	Xerces Documentation		The syntax is the same as for schemaLocation attributes in instance documents: e.g, " http://www.acme.com file_name.xsd ". The user can specify more than one XML Schema in the list.
externalNonamespaceSchemaLocation	The XML Schema Recommendation explicitly states that the inclusion of schemaLocation /nonamespaceSchemaLocation attributes in the instance document is only a hint; it does not mandate that these attributes must be used to locate schemas. This property allows the user to specify the no target namespace XML Schema Location externally. If specified, the instance document's nonamespaceSchemaLocation attribute will be effectively ignored.	Xerces Documentation		The syntax is the same as for the nonamespaceSchemaLocation attribute that may occur in an instance document: e.g. " file_name.xsd ".
scannerName	This property allows the user to specify the name of the XMLScanner to use for scanning XML documents.	Xerces Documentation		The recognized scanner names are:
			W F X M L S c a n n er	A scanner that performs well-formedness checking only.
			D G X M L S c a n n er	A scanner that handles XML documents with DTD grammar information.
			S G X M L S c a n n er	A scanner that handles XML documents with XML schema grammar information.
			I G X M L S c a n n er	A scanner that handles XML documents with DTD or/and XML schema grammar information (default).
entityExpansionLimit	To mitigate an entity expansion attack (aka "XML bomb" or "the billion laughs" attack) you use this tagged value to limit entity expansion to the specified level.			Any integer, no default. <div>  If using this tagged value, provide at least value 1. Otherwise the standard XML entities will not be parsed. </div>