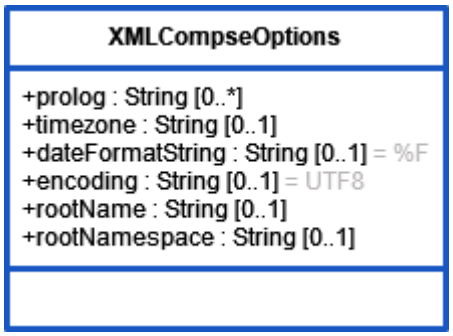
classToXML

Syntax	`set aBlob = anObject.classToXML()` `set aBlob = anObject.classToXML(options)`
Semantics	The operation takes any object (`anObject`) and tries to map it to an XML document. The document structure is given by the class definition of `anObject`. If this is not possible an error is raised (e.g. XML parser errors, invalid mappings, etc.). By default the following mapping rules apply: • Class attributes are mapped to XML attributes. • Association ends are mapped to XML elements. 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. A conversion with operation `classToXML()` always returns an object of type **Blob**. To display this data as a **String** you need to transcode it first (see transcodeToString() Operation).
Substitutables	`anObject` Target object can be any complex object. However, simple types and arrays are not supported, since they do not map naturally to a well formed XML document. `options` `xmlToClass()` offers an optional parameter of type **XMLComposeOptions**.



Its attributes are:

Attribute	Type	Description	Example
prolog	Array of String	The string values are inserted right before the root element of the generated document. This mechanism can be used to insert processing instructions (e.g. DTD and Schema references), comments, entities or any other prolog you may think of. However, be aware that using prolog arrays makes it easy to generate non-well-formed documents.	
timezone	String	Time zone string as specified on Time Zones. The timezone is used to print dateTime expressions. If no timezone is given, UTC is used. If "**`local`**" is used, the date/time is printed relative to the local timezone of the server, e.g. `2012-10-01T12:36:47.0+02:00` (the timezone of the server is UTC+02:00).	`"Australia/Melbourne"`, `"CET"`, `"Etc/GMT+10"`
dateFormatString	String	A format string to be used when printing DateTime values as `xs:date` (e.g. `%F` to print a date without timezone). The allowed formats can be found on Time Zones. If nothing is defined, the XSD standard is used.	`%F`
encoding	String	Encoding of the target xml. Default encoding is **UTF-8**. For a list of possible encodings refer to Charset Definitions.	`"UTF-8"`
rootName	String	Name of the generated XML root element. Use this tagged value to override the default behavior.	`TXTRAW01`
rootnamespace	String	Name of the namespace of the generated XML root element.	

Examples	`set xmlBlob = myAddress.classToXML();`

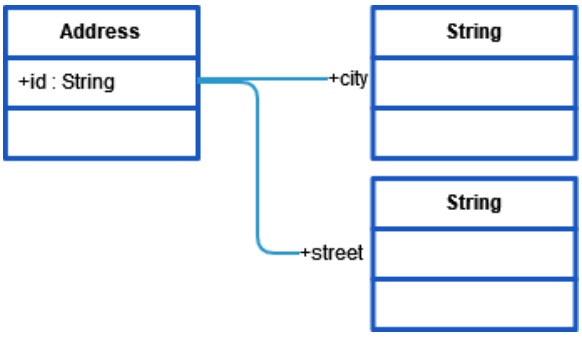
On this Page:

Related Pages:

- classToXMLFragment() Operation
- XML Serialization
- transcodeToString() Operation
- Constants and Names
  - Charset Definitions
  - Time Zones

XML Serialization Example

Assume you have an object **myAddress** of type **Address** .



The following action script serializes this object.

```
set xmlBlob = myAddress.classToXML();
```

The sample XML document below illustrates the mapping executed by `classToXML()`. The object **myAddress** of type **Address** is mapped to an XML document as depicted in the following XML document:

```
<myAddress id="myAddressID">
    <street>108, Kearny Avenue</street>
    <city>Newark</city>
</myAddress>
```

Note, that the XML element **myAddress** is of type **Address** . This type has the 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**.