OSTwoUsrManCPPWriteToXML

Writing SAML Objects to XML

Once you have SAML Objects, writing them to XML is done through a marshalling process. Each object knows how to marshall itself, so you simply invoke the `marshall()` method appropriate for your needs.

Marshalling Example

The following example shows the creation of a SAML 2 Subject, setting some information, and then marshalling it. The exception handling code has been omitted here for readability.

```
// Create the subject
Subject* subject = SubjectBuilder::buildSubject();

// Add an NameID and two SubjectConfirmation items - creation of these items is not shown
subject->setNameID(nameID);
subject->getSubjectConfirmations().push_back(subjectConfirmation1);
subject->getSubjectConfirmations().push_back(subjectConfirmation2);

// Marshall the Subject
DOMElement* subjectElement = subject->marshall();
```

Additional Marshalling Information

Objects can be marshalled against a DOMElement, which reuses the element's document and appends the resulting DOM to the element you pass in.

Alternatively, you can supply a DOMDocument to use, and the resulting DOM is rooted in the document. If you wish to **bind** the document to the SAML object, you can do so with the `setDocument()` method.

Finally, you can simply marshall with no parameters. If the object's DOM exists, it will be reused. If not, a document will be created and **bound** to the SAML object, and used to generate the resulting DOM.

You should take care to minimize document creation and cross-document use of marshalled objects because this can result in extra processing, memory use, and sometimes breaks signatures. But when necessary, the library can deal with document changes internally to a great extent.

Finally, be very careful about taking the results of marshalling and integrating them into DOM trees that are not known to or managed by the library. You can do this, but you should usually take care to leave document ownership outside the SAML objects and just free the document some other way. The SAML objects still need to be freed, but they will leave the DOM intact when deleted, which allows you to use the library to create the DOM but then divorce it from whatever other processing you're doing.

%COMMENT%