

Article

[Istvan Hahn](#) · Jan 6, 2017 4m read

Atelier, The XML Contents Editor

This article is one of the series which introduces Eclipse to experienced Caché/ Ensemble/ HealthShare users. The goal is to open the perspective of a developer who was using Caché Studio for years and make Her/ Him see deeply into the Eclipse world – far beyond Atelier. In other words it is an Atelier (Eclipse IDE for InterSystems technology) beginner's guide. This time the topic is: editing XML files using Atelier.

Editing XML files? We do not do that in Studio. Do we? Yes, we do. Oh, I am sorry. No, we do not.

We do for example

- n HealthShare CDA/ SDA sample messages – pure XML
- n Ensemble production settings, DTL, BPL – codeless, single XML block class
- n HL7 Schema, ZEN, ZEN Reports, Installer Manifest, RESTful service mapper.

So yes, we do a lot of XML editing in our environment, and no we do not edit XML file but DTL, BPL, etc. "What Difference Does It Make?" A lot. When we edit for example a DTL, we expect from the editor that it "knows the DTL syntax". So there is no need to remember all tag and attribute names, but the editor offers the potential valid names, even values in the form of a code completion or IntelliSense. Also it is an expected feature that XML contents is validated on open/ save/ compile. Nothing is easier in the XML world. There are XML schema definitions (DTD or XSD) which provide the required information for the editor. The following lines are describing how Atelier can be extended with XML editing feature and provide the required XML schema definitions.

Install XML Editor

To add the XML editing you need to install the Eclipse XML Editors and Tools classic plug in by using the Atelier Install New Software menu.

Once you installed you can use the New menu to create XML file. Select New/ Other/ XML/ XML file wizard. For the folder I recommend to create a CSP subfolder, because that makes the synchronization relatively easy. Please remember that every file in your Atelier local workspace is subject of synchronization to a Caché server. The CSP subfolders are able to manage raw XML data files at server side.

Then the wizard asks how the XML file is created. Select the Create XML file from XML schema definition option. Now the question is: where does an XSD come from? Eclipse XML editor gives you two options. Either to define it per file or create catalogs. I highly recommend creating catalogs. You should go to Preferences/ XML/ XML Catalog and add your XSD or DTD files. Once you selected the proper XSD, the editor is ready with IntelliSense & validation. Have fun. Oh, I am sorry. I almost forgot to tell, if the XML content is not a complete file but an XData block within a Caché class, then obviously you need to copy-paste the XML data between the file and the XData block. I know, that there should be some tide integration, but let us leave some topic for tomorrow too.

Where does an XSD come from?

Are we done? Almost. Where do I find the relevant XSD files in Caché? Well, that is a very good question.

The good news is that it all depends on what contents we want to edit. Let us have a closer look.

The first example for XML files were the IHE messages, CDA/ SDA documents. Those XML files are used in demos, POCs or any private learning session. Since the messages, documents are complex, it is not too bad, that the editor offers all valid choices for tag and attribute names. The XSD file can be found either inside the HealthShare installation directory or on public web site of organization maintaining the structure (IHE, HL7).

Another good example is the SOAP header. Since header elements are very well defined in structure, it is always easy to find the XSD to a particular header part. Including many kinds of SAML tickets.

Of course Ensemble XML VDoc uses XSD. Just add the same XSD to the Atelier XML Catalog list, and you are able to edit/ validate test messages.

How about the Caché/ Ensemble/ HealthShare internal structures stored in XData block? Some has it's own XSD like Ensemble Production, BPL, DTL, Rule available as an XData block in Ens.Schema.* classes. With some copy/ paste and polishing on the XSD, it works just fine. The most you have to do is to properly complete the <schema> tag by adding the right targetNamespace attribute and xmlns:ns namespace for the local types defined inside the schema. Then you are ready to edit BPL/ DTL/ Rule from Atelier.

And the REST of our XData block structures? Such as REST resource maps, ZEN, ZEN Reports, Installer Manifest, etc. Well, the good news is that we can generate XML schema definition using %XML.Schema utility class. Whenever someone is done with generating the right XSD, the job is done (not talk about importing it to Atelier).

Epilogue

It looks like a hell lot of work. Indeed it is not. You can collect and load the most frequently used XSD files in few hours, and then you are on the way "to infinity and beyond".

If you can not refrain from sharing XSDs I am sure you are more than welcome at InterSystems Developers Community.

Happy XML editing.

[#Beginner](#) [#Development Environment](#) [#Atelier](#)

Source URL: <https://community.intersystems.com/post/atelier-xml-contents-editor>