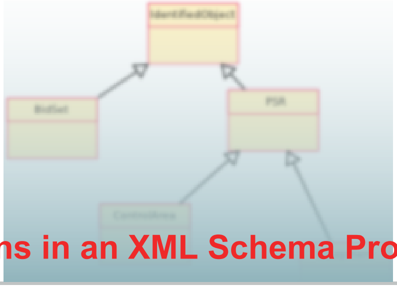

Why There Are so Many Top Level Definitions in an XML Schema Profile

Profiles for XML Schemas typically have a single concrete class that becomes the top level element definition. Everything else is nested below it and the CIMTool profile outline view shows this tree structure.

But in profiles built with CIMTool 1.9, we typically see all the classes at the top level of the outline view, not just the concrete class. What is going on and why the change?

Be assured, if you expand the concrete class you see the same structure as before. The difference is due to the use of named classes versus anonymous classes to express the profile.

Earlier version of CIMTool used to favour the creation of nested anonymous classes but 1.9 favours named classes (more about the reason for this change later).

What you see at the top level of the profile outline in any version of CIMTool are all the *named* classes. But older profiles are composed of mostly anonymous classes so you don't see them at the top level.

These top level profile class do not mean the generated XML schema will have lots of top level elements. Each profile class actually maps to an `xsd:complexType`. So the outline is showing you the complexTypes as they will appear in the schema.

You only get a top level element if a class is also marked "concrete". A concrete class in a profile can be can be recognised by its icon, the concrete checkbox on the stereotypes page and the concrete checkbox on the restriction page.

Why the change in CIMTool to favour named classes?

First, note that you can still create anonymous classes if you want. There is a button on add/remove for this, top right in the appropriate context. (See Associations and Anonymous Classes for details.) But by default CIMTool creates named profile classes.

Anonymous classes give you complete flexibility to profile the same CIM class a different way each time it appears. But they don't encourage or even allow you to reuse a profile class definition (or `xsd:complexType`) in more than one place.

(And incidentally they are hard to find in CIMTool's profile outline as they are buried in the hierarchy.)

Starting with 1.9 you get get all the flexibility you want with named classes. You can have more than one per CIM class and you can select which to use for each property on the restriction page.

At the end of the day none of this affects the "interpretation" of the generated XSD. Just whether the complexTypes are inline or referenced by name.

- Wiki Home
- Site News
- Download
- Recent Changes
- Source Code
- License
- How to ...
- Discussion