

This is Brendan Martin. I'm your AFCARS technical administrator. I want to give you a very, very short demo of probably the single most important tool you can use to build a good XML file that is namely validating your XML against the schema we provide.

First of all, I'm hoping you know what a schema is and where to find it. If you go to the AFCARS, technical bulletins page, go to technical bulletin, #21.

And it will offer you a download of the schema, XSD schema definition that is a zip file, and you'll need to download it, unzip it, and then get the dot XSD file which is a schema. Here's what a schema looks like. You're probably already familiar with it, hopefully.

A schema is a kind of XML that actually describes the requirements for building an XML file of that type. You're familiar by now with the sample XML that we've provided hopefully, and you'll see that the top-level element is called ACF data, that ACF is known as a namespace.

It's essentially describing this as an ACF type of document and don't be confused by the URL's. The software here is not actually going out to the website to get anything. It's a namespace has the requirement of being in URL form for historical reasons.

What I want to show you what to add to your XML, is this line here which points to a local copy of your schema. So, this schema location line is telling you for anything belonging to the HTTP www.acfhhs.gov namespace use the schema OOH XSD to validate that data.

The schema OHD can be, you can name your schema when you download it from the website. You can name it whatever you want. You can put it wherever you want in this case, since you don't see a path before the name, it means this is this local copy of the schema is in the same directory as the XML. If you wanted to put it in a different directory, you would path it.

So again, I want to just point to. Don't be confused by this stuff at the top. It's conceptually a little tricky to understand what a name space is, but this line here, the first line is defining the ACF namespace. This attribute XMLNS colon ACF equals this URL, that is saying we are declaring a namespace called ACF and its value is that URL. Once again, it's confusing because it looks like you're going to the web to get the schema, but that's actually not the case.

The other thing I need to show you here is that while you're seeing it on the screen, actually is that you need a decent XML editor to view XML, to edit it, to you know, use the indentation, the colors, etcetera. This this one you see on the screen right now is notepad++. I actually prefer this one over here which is called Visual Studio code. Both of these, you can download from the web. They're free, open source.

So, I'm going to demonstrate how to perform validation in this Visual Studio Code 1. In order to

implement a schema, here's that line again, where you point to the schema and, say, validate everything in the ACF namespace using this schema. To get it to do its magic, you need to get what's called a plug in for this software. I honestly don't remember how to do that. You Google plug-in validation plugin for Visual Studio code or whatever editor you were using, and it'll instruct you how to get that.

That's the software that actually performs the schema validation when you insert the line telling the editor where to find the schema, and again as I showed you a moment ago, the schema location we've specified that a local copy of the schema called Schema OOH XSD. OOH being out of home care, is going to be used to validate anything in this namespace.

Which is again into URL, but it's really a name space and that name space.
Is the ACF namespace. So, you can see the top level your data is is qualified as ACF colon data.

It's data, it's a data element of type ACF one of the reasons for namespaces is that is that in different kinds of XML, different files we use might use the same name for something, and to distinguish between, you know, the ACF's data element and other possible data elements you want to use that qualifier in namespace.

Anyway, when you have the plugin and when you have the schema specified and the schema is really there where it says it is and the schema is valid, you'll be able to perform validation and I have a simple example here as I put in an invalid value in the date of birth. Let me just show you.

The schema itself enforces valid values way down at the bottom. This is very kind of cryptic to read, but it's if you look at these, you'll you can kind of glean that they're saying you need to have 4 characters you need 4 digits, et cetera.

This very cryptic notation is called regular expression. It's notation that's common to many programming languages, and it tells you how to how to parse a string of characters, so those values are enforced by the schema and if we go back to the other editor you'll see that you know for date of birth you need an 8 digit field, invalid values, clearly invalid and when you have the plugin installed in your editor and you're pointing to the schema, you'll see how very legible and descriptive the validation errors are. They are also very visible, you get a red squiggly line, under everything that's in invalid. So, this is clearly an invalid value. If I make it a valid value, you will no longer see validation errors.

Likewise, if you have misnamed tags, you're going to see validation complaints. You'll even have possible ways of fixing the problem by simply by clicking the quick fix, etcetera. So that's about it. I don't have anything more to show you. This is what you're going to need to debug your own XML extracts. So, enjoy it. Thank you.