

Guidance on the maintenance of WMDR: using the model in EA with SVN and re-generating the XSD

Author: Dominic Lowe, Australian Bureau of Meteorology

First Version: 29 June 2015

Revised to add schema generation section: 28 March 2017

Circulation: WIGOS team

Table of Contents

Introduction	1
1. Install Enterprise Architect.....	2
2. Install a Subversion Client.....	3
2.1. Check the SVN installation	3
3. SVN Basic Concepts.....	4
4. Register to access the model repository.....	5
5. Use SVN to set up a working copy of the shared model repository	6
6. Set up Enterprise Architect to access the model repository	7
6.1. Set up a reference to the WMO SVN working copy.....	8
6.2. Setting up the ISO TC211 configuration.....	9
6.3. Prepare the EA project structure (optional step).	10
6.4. Fetch the ISO packages.	12
6.5. Get the OGC models	13
6.6. Get the WMO models	14
7. Final step: Update the models	15

Introduction

The WIGOS Observations Metadata model will be developed using SPARX Enterprise Architect.

- The team has agreed to use **Enterprise Architect version 12** in conjunction with Subversion.
- Various editions of this software are available.

- At least the **Professional** edition is needed to enable integration with Subversion.

This document explains how to:

- Install Enterprise Architect.
- Install the required version control software (Subversion, also known as SVN).
- Understand the basic concepts of SVN.
- Register to access the model repository.
- Use SVN to set up a working copy of the shared model repository.
- Set up Enterprise Architect to access the model repository.
- Fetch the model(s) from the repository.
- Fetch updates (get latest version) of the model(s) from the repository

The document then explains how to:

- Modify the model (e.g. add new attributes)
- Re-generate the XML Schema

It is assumed there is a primary model editor and that everyone else will just need to fetch updates from the repository to inspect the model.

Having multiple editors working on the model can create synchronisation and 'locking' problems that will need to be discussed in detail between editors involved to establish a good working practice. This is probably best discussed in detail with the editors involved.

The following steps should be followed to configure Enterprise Architect for use with the WMO Subversion repository.

1. Install Enterprise Architect

Enterprise Architect is a commercial tool that can be purchased here:

<http://www.sparxsystems.com.au/products/ea/purchase.html>

At least the 'Professional' version is required.

A fully functional 30 day trial can be downloaded from here:

www.sparxsystems.com.au/products/ea/trial.html

[As an aside there is a free EA reader available, however this doesn't connect to SVN and can't read models from XMI files (the format used to store the model in SVN) so using the reader only would require that we set up an arrangement involving the regular exchange and updating of large EAP project files which is not really desirable. The reader is therefore not recommended. If a group member is unable to acquire Enterprise Architect then the model could be occasionally exported in HTML for viewing by that member. But this is sub-optimal as it requires more work – so it is preferable that EA and SVN is used by all team members].

The install is a typical Windows installation process. It may require administrative rights.

2. Install a Subversion Client

The models are stored in a subversion *repository*. In order to interact with the repository you need a subversion client.

For the purpose of EA and SVN it is recommended that you install TortoiseSVN on Windows.

Again this is a straightforward Windows install, with one exception:

When installing it is important to check the option that installs the 'command line tools' as EA is dependent on these. By default they are not installed.

TortoiseSVN can be downloaded for free here (download the latest version)

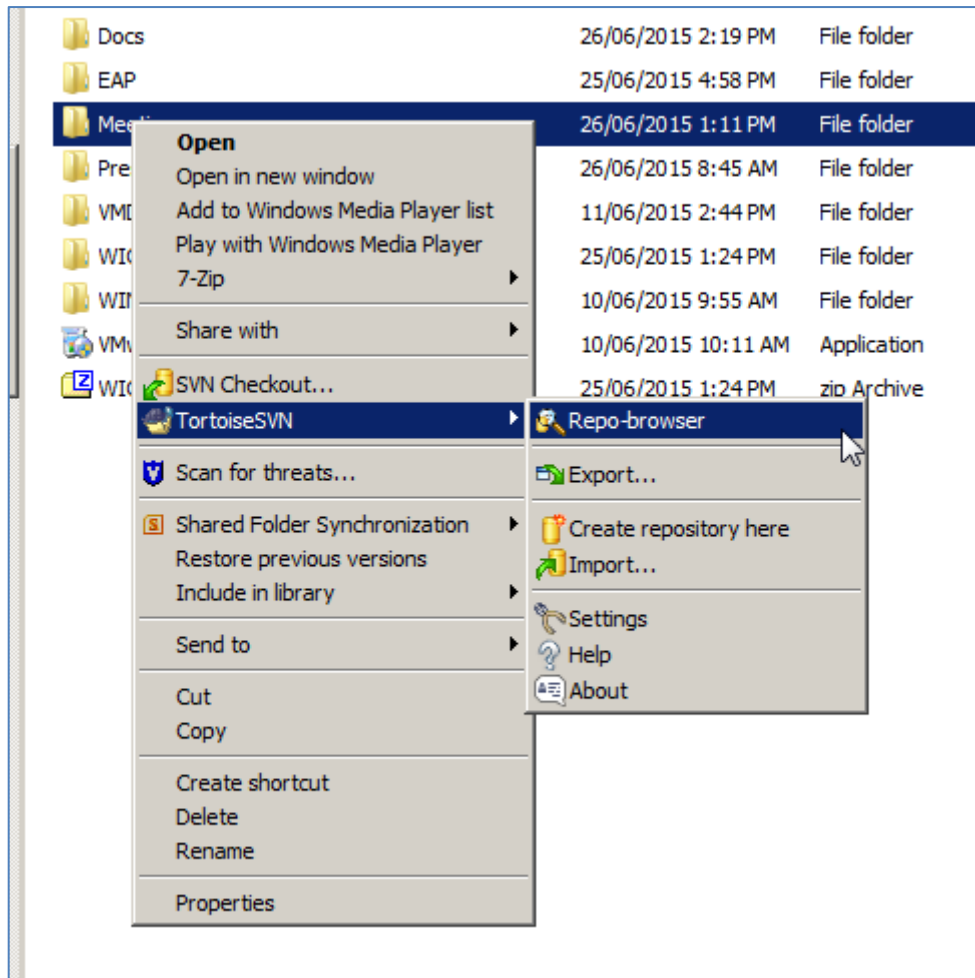
<http://tortoisesvn.net/downloads.html>

2.1. Check the SVN installation

To check the command line tools are installed go to the **Start/Run** menu and type '**cmd**'.

Then type '**svn help**' at the command line. You should see some output from subversion. If the command is not recognised then repeat the TortoiseSVN installation and ensure you select the command line tools.

After installation you should also see some new options in your right click menu (when right-clicking on a directory). See the below image.



3. SVN Basic Concepts

The basic concepts of SVN are outlined below. This isn't a complete explanation of how Subversion works but covers the key principles that are of concern to the group.

The repository:

There is a central subversion repository where the files (in our case the models) are stored.

Users of the repository can make changes to files in the repository.

The repository also stores history and change information so you can see and access the files as they were at any point in time.

We will use Enterprise Architect to make changes to the files in the repository and upload/download changes.

A Working Copy:

Using a subversion client (like TortoiseSVN) you can make a 'working copy' of the repository on your local machine. The working copy is a copy of the repository (or part of the repository).

This working copy enables you to upload and download files (and changes to files) to the central repository using an SVN client. [Note that after the initial configuration of the working copy we will be effectively be use Enterprise Architect as our svn client and won't use our actual svn client (TortoiseSVN) directly.]

Basic subversion workflow (forgetting about Enterprise architect for now)

Subversion typically used to manage software development projects and enable multiple developers to edit code. The normal basic workflow is as follows:

- You '**checkout**' a copy of the central repository to create a local copy.
- You now have a '**working copy**' of the files from the repository.
- In your working copy you then do some work...
 - You can edit files (e.g. change the text), perhaps '**add**' files, '**move**' files, '**rename**' files etc. (All in your working copy.)
- You then '**commit**' your changes to upload them to the central repository.
- Everybody who can access the central repository can now see your changes in the repository.
- Other users can do an '**update**' on their own working copies and they will download your changes.
- You also regularly (before every work session) do an '**update**' on your local working copy to see if anyone else has committed any changes.
 - If so those changes are downloaded to your working repository by the update command.
- Sometimes '**conflicts**' arise between different peoples '**commits**' and then you can use an SVN '**merge**' command to fix them.
 - **Merging** is hard and can get messy.
 - it is therefore best to '**commit**' your changes regularly and '**update**' regularly to avoid divergence and '**merging**' problems.

4. Register to access the model repository.

The development branch of the model repository is (as of March 2017) here:

<https://wmo.projecthut.com/svn/wmdr/branches/development/>

To access this you will need to register with the WMO SVN (contact Steve Foreman)

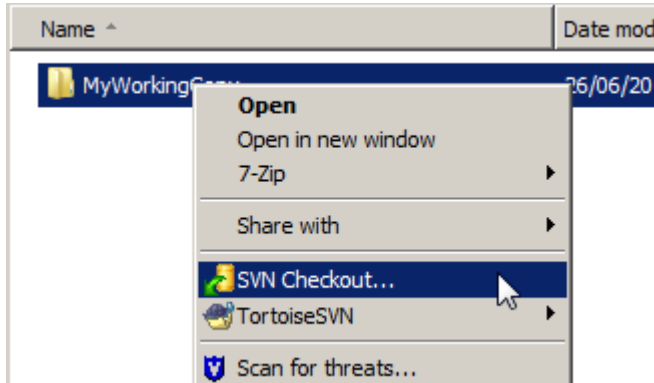
Once registered you can access the repository using your username and password.

5. Use SVN to set up a working copy of the shared model repository

Using TortoiseSVN we will set up a working copy of the shared model repository.

First create a new folder on your computer to hold the working copy (e.g. MyWorkingCopy).

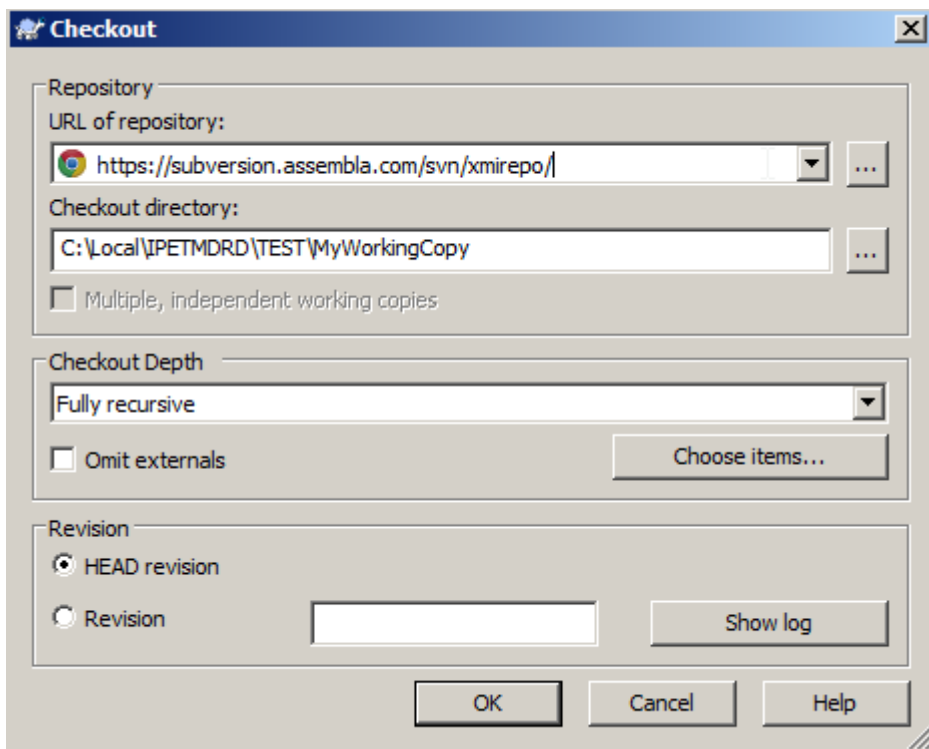
Then right click on that folder and select 'SVN Checkout'



In the dialog box enter the repository URL as shown below (leaving the other values as they are) then select 'OK'.

NOTE: Update 2017 – the URL of repository has changed since this screenshot was taken and should now read:

<https://wmo.projecthut.com/svn/wmdr/>



This should make a copy of the entire repository to your working copy as follows. (you may or may not see the hidden `.svn` folder depending on your settings).

 <code>.svn</code>	26/06/2015 3:41 PM	File folder
 <code>branches</code>	26/06/2015 3:41 PM	File folder
 <code>tags</code>	26/06/2015 3:41 PM	File folder
 <code>trunk</code>	26/06/2015 3:41 PM	File folder

You will see that there are 3 directories; branches, tags and trunk.

These folders are a typical subversion convention. There are various possible strategies for working in these structures but it was agreed (before the ad hoc meeting) that:

- **The trunk** is stable. It contains a stable copy of the files.
 - We won't actively work in the trunk
 - We will only migrate work there once complete (i.e. public version 1.0 will go in the trunk)
- **Branches** are to be used for work on new models/changes to existing models.
 - **Active work takes place in a branch called `./branches/development`**
 - When work done in branches is finally complete the work can be migrated into the trunk.
- **Tags** will be used to **version** the model at a particular time.
 - Tags are simply 'snapshots' or copies of the repository (or parts of the repository) at a particular time.
 - We use tags to version releases.
 - Once something is tagged it should not change.
 - We never work on tagged directories.

To summarise –

- We do the work in a branch.
- Only finished work is migrated to the trunk (when final).
- Versioned snapshots will be made available as tags.
 - We don't change these once tagged.

You should now have a 'working copy'. The next step is to configure EA to read from this working copy.

6. Set up Enterprise Architect to access the model repository

There are two main steps to configuring EA to work with an SVN repository:

- Set up a reference to the working copy for each SVN repository
- 'Get Package' from the working copy for each model you wish to import

6.1. Set up a reference to the WMO SVN working copy

Open a new EA project and save it (dismiss any wizards).

Now select the menu item *Project/Version Control/Version Control Settings*

Complete the dialog box as shown in this screenshot:

Version Control Settings

Model Settings

- ☐ This model is private (for Shared models, it is best to disable this check box)
- ☒ Save nested version controlled packages to stubs only (recommended)
- ☐ For all packages, create placeholders for external references

Configuration Details:

Unique ID:

Type: ☐ SCC ☐ CVS ☒ Subversion ☐ TFS

Working Copy Path:

Workstation Settings:

Subversion Exe Path:

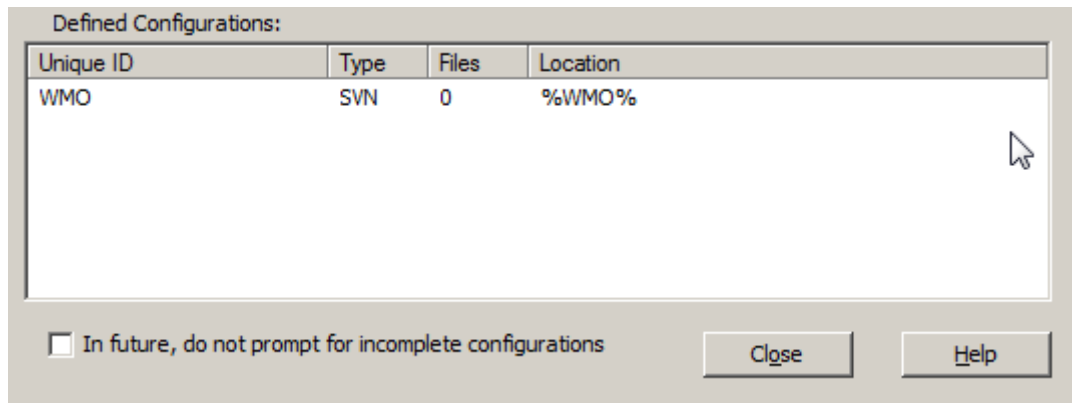
VC Time-Out Value: seconds

Defined Configurations:

Unique ID	Type	Files	Location
-----------	------	-------	----------

☐ In future, do not prompt for incomplete configurations

- The unique ID can be anything – the example uses 'WMO'.
- Select 'subversion' as the version control type.
- The working copy path should point to your SVN working copy (i.e. the folder that contains the branches, tags, trunk folders)
- The subversion path should be automatically completed. If not you need to find your svn.exe on the filesystem. (if there is no svn.exe you might have not installed the command line tools – so redo the TortoiseSVN installation).
- Then click 'save' and you should see the 'WMO' configuration saved in the list of 'Defined Configurations' as follows.



You will also need a configuration for the METCE model and the ISO TC211 models.

6.2. Setting up the METCE configuration

This is configured in the same way:

- SVN Checkout the WMO-ICAO models to a directory on your computer using TortoiseSVN.
 - <https://wmo.projecthut.com/svn/wmo-icao-models/>
- Configure an EA version control configuration for this folder using the Version Control Settings dialog box (as just done the WMO config). You will need to select 'New' to add the new ISO configuration. Call it something else like 'WMO-METCE'

6.3. Setting up the ISO TC211 configuration

This is configured in the same way:

- SVN Checkout the ISO models to a directory on your computer using TortoiseSVN.
 - <https://www.seegrid.csiro.au/mirrors/iso-harmonized-model>
- Configure an EA version control configuration for this folder using the Version Control Settings dialog box (as just done the WMO config). You will need to select 'New' to add the new ISO configuration. Call it something else like 'isotc211'

Your EA should now be configured something like this, with references to three SVN repositories:

Version Control Settings

Model Settings

☐ This model is private (for Shared models, it is best to disable this check box)

☒ Save nested version controlled packages to stubs only (recommended)

☐ For all packages, create placeholders for external references

Configuration Details:

Unique ID:

Type: ☐ SCC ☐ CVS ☒ Subversion ☐ TFS

Working Copy Path:

Workstation Settings:

Subversion Exe Path:

VC Time-Out Value: seconds

Defined Configurations:

Unique ID	Type	Files	Location
isotc211	SVN	106	%isotc211%
WMO	SVN	10	%WMO%

☐ In future, do not prompt for incomplete configurations

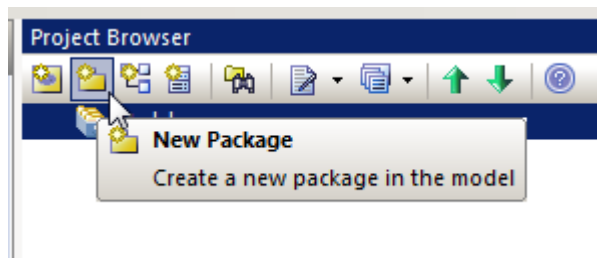
Note: Update 2017 – a third configuration should be shown for WMO-METCE (previously METCE and WMDR shared a subversion repository, now they have one each).

6.4.Prepare the EA project structure (optional step).

Now the configuration is set up you can fetch the models from the repository.

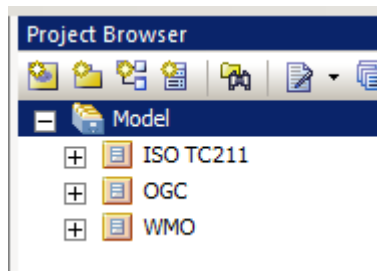
Although not essential it is recommended that you create separate empty packages in EA for the ISO Models, OGC Models and the WMO models. This will make it easier to navigate later.

Simply click on 'New Package' in the Project Browser to create each empty package:

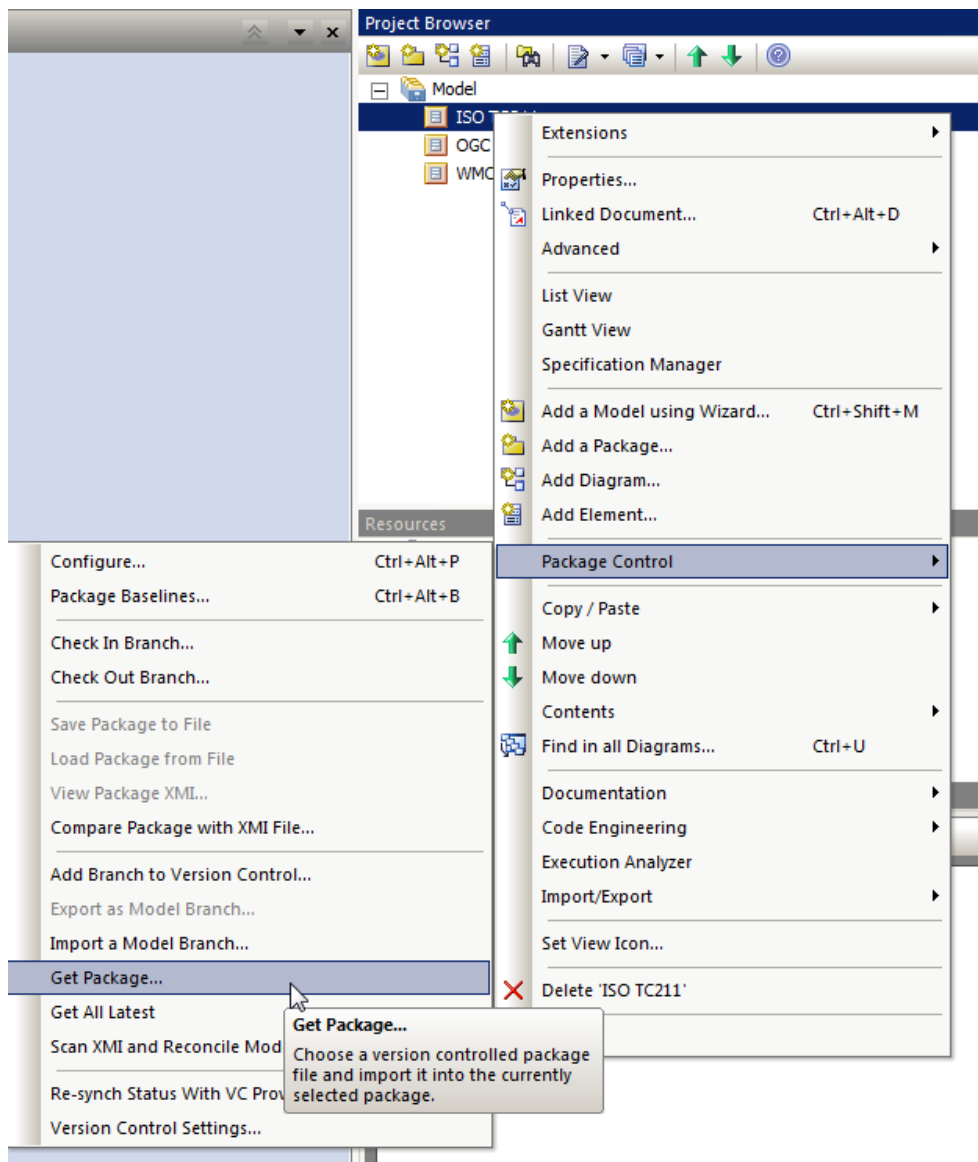


(note: if you can't see the Project Browser, select it from the View menu).

Your model should look like something this:



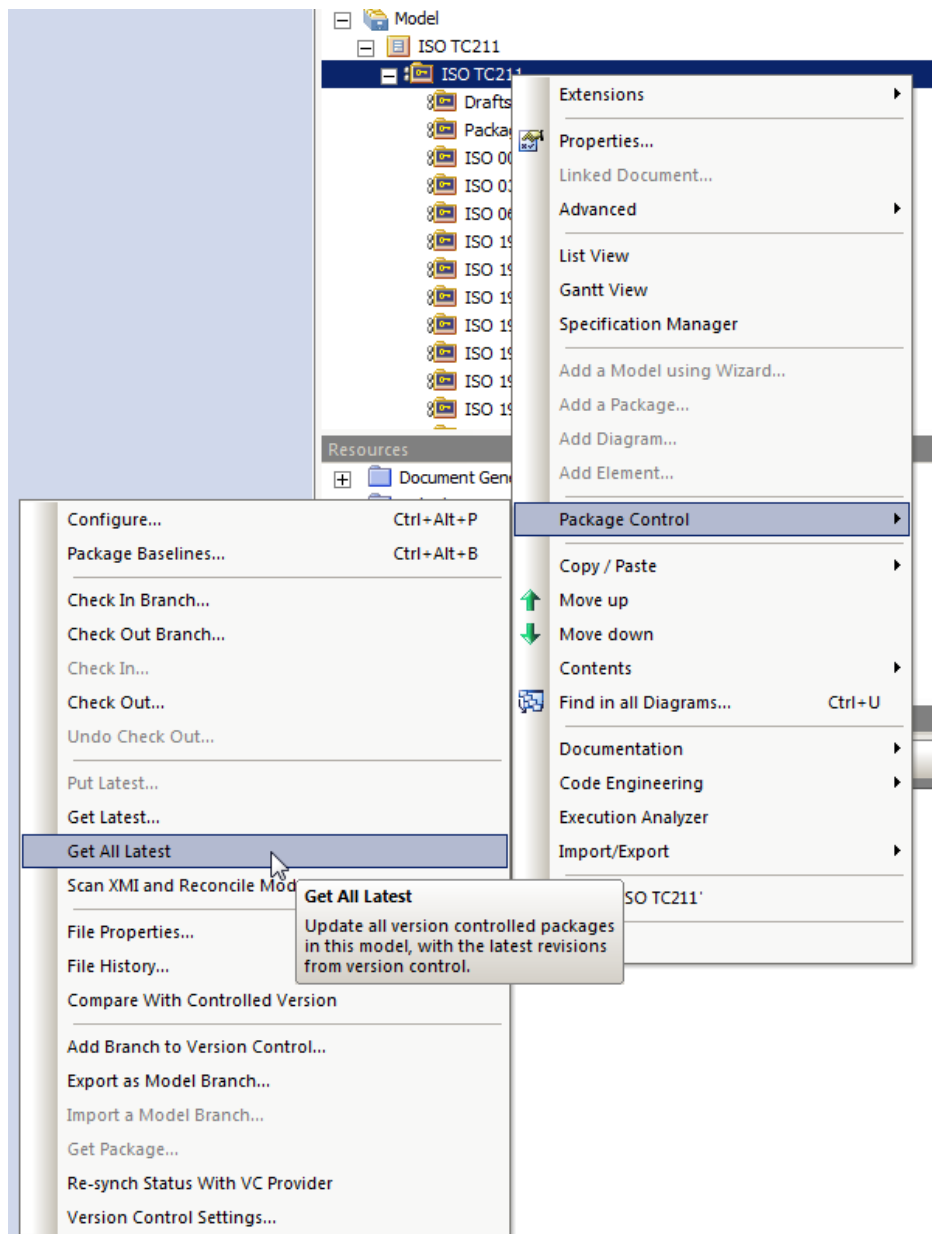
Now you can populate these packages from the repositories. It is important to use the '**Get Package**' command for this operation (see below).



6.5.Fetch the ISO packages.

- Right click on the ISO TC211 folder and select Package Control/Get Package.
- Select 'isotc211' version control configuration and the list should populate with XMI files.
- Select the XMI file called: **isotc211\ISO TC211.xml** and click OK.

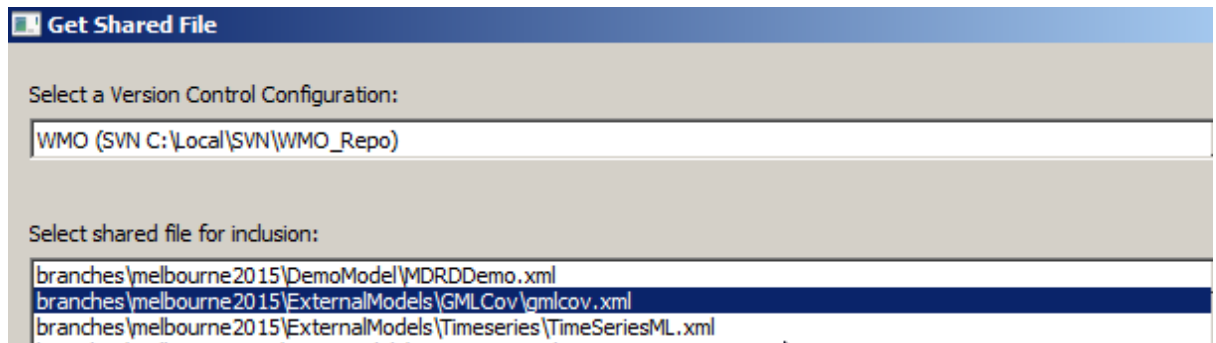
This installs a set of stubs. To fully include the ISO models right click on the Package you have just imported (ISO TC211) and select Package Control/Get All Latest (then select the first import option given). This will download all the ISO models from your repository. It may take a few minutes.



6.6. Get the OGC models

Note: Update 2017 -This step may be omitted

It may be useful to have a copy of TimeseriesML and GMLCov (GML Coverages schema). These are in the 'branches\melbourne2015\ExternalModels' directory. Do a 'Get Package' for each of these. You won't need to do the 'Get All Latest' step for these models.



6.7. Get the WMO models

Finally you can do a 'Get Package' for each of the WMO models.

For WMDR make sure you get the packages from the branches\development branch in the 'WMO' configuration.

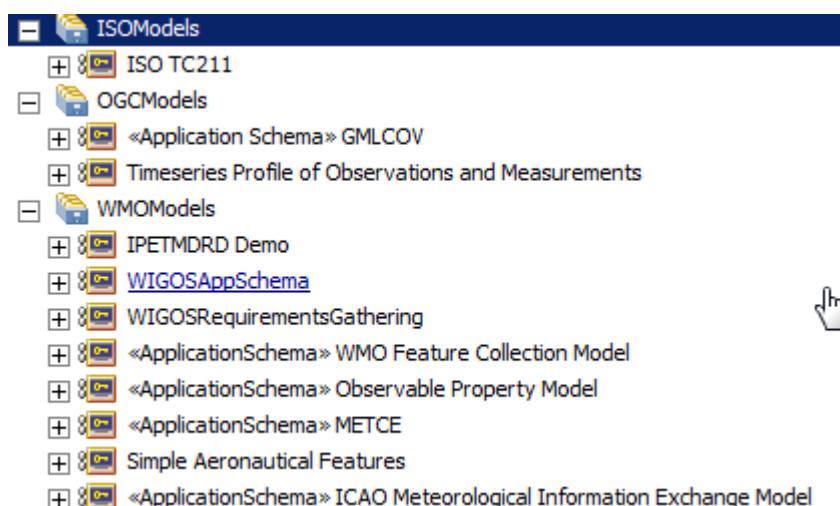
The WMDR model is in branches/development/xmi/wmdr.xml

For METCE make sure you get the branch from the trunk in the 'WMO-METCE' configuration.

The METCE model is in trunk/wmo-metce.xml

You should now see all the models in your EAP project, along with 'locked' symbols to demonstrate that they are connected to subversion and they aren't editable.

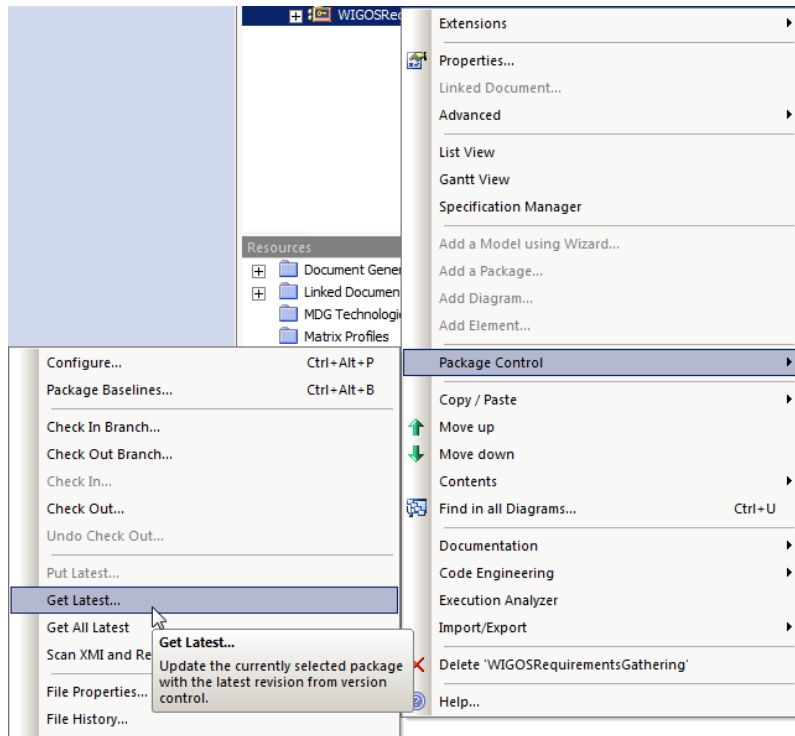
Note: Update 2017 – it will look similar to below but may not be identical.



7. Final setup step: Update the models

To get the *latest* version of the WMO models:

Right click on each WMO package and select Package Control/ Get Latest as shown below.



This will download the latest copies of models from the assembly subversion repository.

It is strongly recommended that you do this every time you open the project (especially if you know there are changes being actively made).

Note: Please don't 'check out' the WMO models – treat them as read only unless the group agrees otherwise. That will enable the editor(s) to work on them.

8. Schema maintenance procedure

There are several steps to updating the WMDR schema, involving:

- Making model edits
- Regenerating the schema
- Updating model documentation
- Updating schema documentation

To edit the model the workflow is:

- Get latest (ensure you are up to date)
- Check out
- Edit the model appropriately
- Check in

To generate the schema the workflow is:

- Generate GML Application Schema from EA
- Run python script to fix up a couple of bugs – this step is very important.

To update the HTML version of the model the workflow is:

- Generate HTML documentation from EA
- Zip up and replace the old version on SVN

To regenerate the schema documentation the workflow is:

- Run the 'docflex' tool on the new version of the schema

These steps are described in more detail below, but it is also recommended to read the WMO Guide to Data Modelling.

8.1.Editing the model – Step 1: Get Latest

Before any editing it is important to ensure you have the latest version from the SVN. Do a 'get latest' on the WMDR package as described in Section 7 above.

8.2.Editing the model – Step 2: Check Out the Model

Right click on the model package containing the WMDR model. This is probably called WMDR but you may have set things up differently. *It is the package above the <<ApplicationSchema>> WMDSchema package.*

Select Package Control / Check Out....

Make sure you use 'Check Out..' not any other check out command.

8.3.Editing the model – Step 3: Make Edits to the Model

The model can be edited to make changes. It is strongly recommended to read the WMO Guide to Data Modelling.

Some key points to note:

Every attribute in the model, and every association role target end has a 'tagged value' called 'sequenceNumber'. This tagged value controls the order of elements in the schema and it is critical that it is present and that there are no duplicates in a class or the schema order will change.

So – to add a new attribute to a class:

- Add the attribute
- Select its type from the appropriate model – e.g. CharacterString is in ISO 19103. Don't just type CharacterString – you have to select it from the model or the XML schema generation may not work.
- Add a tagged value to the attribute called 'sequenceNumber'. Give it the next sequence number in the list. E.g. if the class has 5 attributes, call this '6'. If you wish to place it earlier in sequence you will need to modify the other sequence numbers accordingly for that class.
- Note that associations from classes also have sequence numbers – these have typically been set higher e.g. 10, 20, 30. However care must be taken not to clash with these numbers also – modify if needed.

Cardinality changes can be made simply by changing the cardinality.

Name changes can be made simply by changing the name.

For more complex changes like adding new classes please read the WMO Guide to Data Modelling.

8.4.Editing the model – Step 4: Check in changes to the Model

Now the changes are made the model can be checked in to SVN. Right click on WMDR and select Package Control / Check In...

Alternatively the schema can be re-generated and tested at this point and the model checked in when you are happy with the schema output.

8.5.Generating the Schema – Step 1: Generate initial schema from EA

Enterprise architect can automatically generate the WMDR GML Application Schema.

To do so, right click on the <<Application Schema>> WMDSchema package and select Extensions/GML/Generate GML Application Schema.

Ensure you select GML 3.3 as this is needed to encode the gml:Reference type for codelists.

Generate the schema to overwrite the branches/development/xsd/wmdr.xsd in your SVN working copy.

However this schema is not yet 100% correct so there is a second step...

8.6.Generating the Schema – Step 2: Run the Python script

The output from EA contains a bug (a typo - xer instead of exr), and also does not correctly encode the responsible party relationship.

There is a python script in the SVN repository that must be run to do this.

The script will need modifying to work locally as it has a path name hardcoded in it.

The script is run by 'python postProcessModel.py' in the python directory.

Note that this script was written to work on Linux. If running on Windows the `os.system` call on line 23 probably won't work. This could be re-written or it can be commented out and done manually. All this line does is find and replace the EA typo <http://www.opengis.net/gml/3.3/xer> with <http://www.opengis.net/gml/3.3/exr> (This bug has been reported to EA).

The rest of the script should work on Windows with python installed.

8.7. Generating the HTML Documentation – Step 1: Generate from EA

To generate the HTML model, right click on the application schema package and select Documentation/HTML Report

8.8. Generating the HTML Documentation – Step 2: Create Zip

Using a zip program create a zip file of all the files in the HTML report just created. Call it `wmdr.zip` and overwrite the file on SVN at `branches/development/html/wmdr.zip`

8.9. Generating the Schema Documentation – Step 1: Generate using Docflex

The docflex tool is used to generate schema documentation. <http://www.filigris.com/docflex-xml/generator.php>

Download the tool then run `generator.bat`. This provides a simple GUI where you select the schema to document. Generation takes a few minutes.

8.10. Generating the Schema Documentation – Step 2: Check in to SVN

The schemadoc generated consists of many files. Since SVN tracks individual files it is important to SVN remove any that are no longer used and SVN add any that are new.

To do this, inspect the files created in the schema doc (particularly in the `wmdr` subdirectory) and look at the TortoiseSVN symbols to check for new files, or files that have not been updated (and are therefore no longer used). Check in all changes to subversion.

Alternatively you can SVN Delete the old schematron and commit before adding the new afresh.

8.11. Finally

Ensure that model (`xmi`), schema, html and schemadoc are all checked in. Update the example file if necessary to validate against the new schema. If appropriate, create a new tag in SVN for the new version.

