WIGOS METADATA DATA RECORD MODEL AND SCHEMA MAINTENANCE

Authors:

Dominic Lowe, Australian Bureau of Meteorology

Jörg Klausen, MeteoSwiss

Tom Kralidis, Meteorological Service of Canada

Circulation: WIGOS team

|  |  |  |
| --- | --- | --- |
| Revisions: |  |  |
| 2015-06-29 | Dominic Lowe | Initial version |
| 2017-03-28 | Dominic Lowe | Revised to add schema generation section |
| 2017-09-28 | Jörg Klausen | References to svn repository updated |
| 2017-12-28 | Tom Kralidis | Update steps, screenshots, build/tag/release procedures |
| 2018-01-17 | Tom Kralidis | Update development workflow (section 6) |
| 2018-10-03 | Tom Kralidis | Update WMO contact, SVN repository locations |

Table of Contents

[1. Introduction 2](#_Toc506526177)

[2. Install Enterprise Architect 3](#_Toc506526178)

[3. Install a Subversion Client 3](#_Toc506526179)

[3.1. Check the SVN installation 4](#_Toc506526180)

[4. SVN Basic Concepts 5](#_Toc506526181)

[5. Register to access the model repository. 6](#_Toc506526182)

[6. Use SVN to set up a working copy of the shared model repository 7](#_Toc506526183)

[7. Set up Enterprise Architect to access the model repository 8](#_Toc506526184)

[7.1. Set up a reference to the WMO SVN working copy 8](#_Toc506526185)

[7.2. Setting up the METCE configuration 10](#_Toc506526186)

[7.3. Setting up the ISO TC211 configuration 10](#_Toc506526187)

[7.4. Prepare the EA project structure (optional step). 11](#_Toc506526188)

[7.5. Fetch the ISO packages. 13](#_Toc506526189)

[7.6. Get the WMO models 14](#_Toc506526190)

[8. Final setup step: Update the models 15](#_Toc506526191)

[9. Schema maintenance procedure 16](#_Toc506526192)

[9.1. Editing the model – Step 1: Get Latest 16](#_Toc506526193)

[9.2. Editing the model – Step 2: Check Out the Model 17](#_Toc506526194)

[9.3. Editing the model – Step 3: Make Edits to the Model 17](#_Toc506526195)

[9.4. Editing the model – Step 4: Check in changes to the Model 17](#_Toc506526196)

[9.5. Regenerating the Schema from EA 18](#_Toc506526197)

[9.6. Committing Updates to Subversion 18](#_Toc506526198)

[10. Release Procedure 19](#_Toc506526199)

[10.1. Updating the version in the EA project 19](#_Toc506526200)

[10.2. Updating remaining repository artifacts 20](#_Toc506526201)

[10.3. Creating release tag in Subversion 20](#_Toc506526202)

[10.4. Generating the HTML Documentation from EA 22](#_Toc506526203)

[10.5. Generating the Schema Documentation using Docflex 22](#_Toc506526204)

[10.6. Generating model and schema specification PDF 23](#_Toc506526205)

[10.7. Creating the release bundle 23](#_Toc506526206)

# Introduction

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

* The team has agreed to use **Enterprise Architect version 13.5** in conjunction with Subversion.
* Various editions of this software are available.
  + At a minimum the **Professional** edition is required 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.

# Install Enterprise Architect

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

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

At a minimum the ‘Professional’ version is required.

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

[www.sparxsystems.com.au/products/ea/trial.html](http://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. Installation may require administrative rights.

# 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.

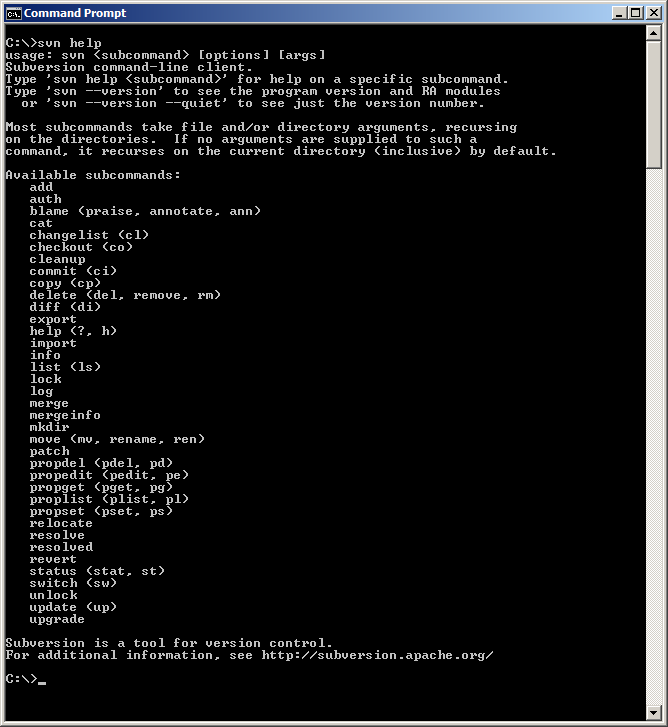
TortoiseSVN can be downloaded for free here (download the latest version)

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

## Check the SVN installation

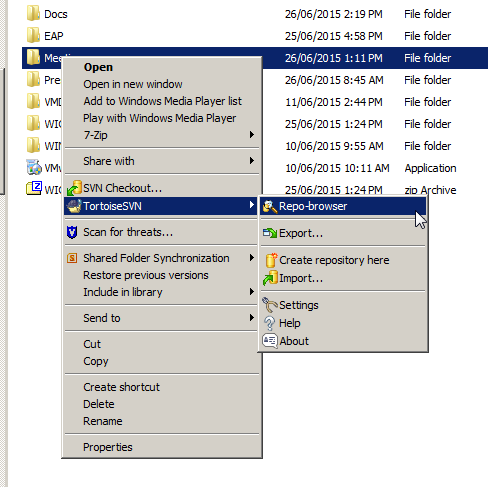
To check the command line tools are installed go to the **Start/Run** menu and type **‘cmd’**.

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



# 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 effectively be using 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.

# Register to access the model repository.

The development branch of the model repository can be found at:

<https://pl7.projectlocker.com/WMO/wmdr/svn/trunk>

To access this you will need to register with the WMO SVN (contact Enrico Fucile, WMO, efucile@wmo.int)

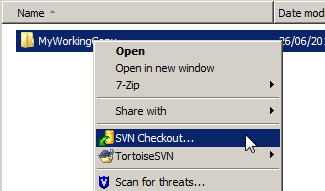
Once registered you can access the repository using your username and password.

# 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.

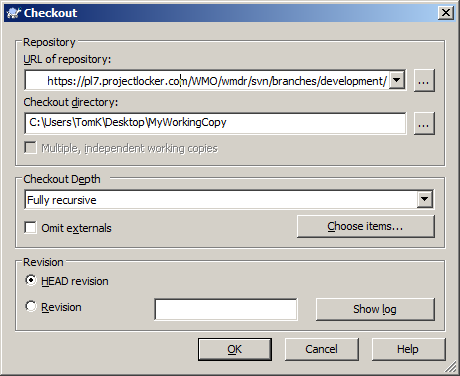
First create a new folder on your computer to hold the working copy (e.g. MyWorkingCopy).

Then right click on that folder and select ‘SVN Checkout’)

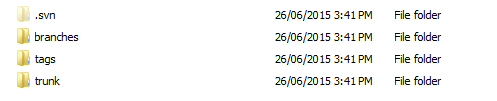


In the dialog box enter the repository URL as shown below (leaving the other values as they are) then select ‘OK’.

https://pl7.projectlocker.com/WMO/wmdr/svn/



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 Windows folder settings).



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:

* **trunk:** is the baseline for development. It contains a stable copy of the files
  + We actively work in trunk/ for feature development of the current model.
* **branches** are to be used for work on changes to stable branches, such as bug fixes.
  + When work done in stable branches is finally complete the work can be tagged and released, e.g., a bug fix version x.y.z
* **tags** will be used to **version** the model at a particular point in time
  + tags/ is simply ‘snapshots’ or copies of the repository (or parts of the repository) at a particular point in time
  + We use tags/ to version releases
  + Once something is tagged it does not change
  + We never work on tagged directories

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

# Set up Enterprise Architect to access the model repository

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

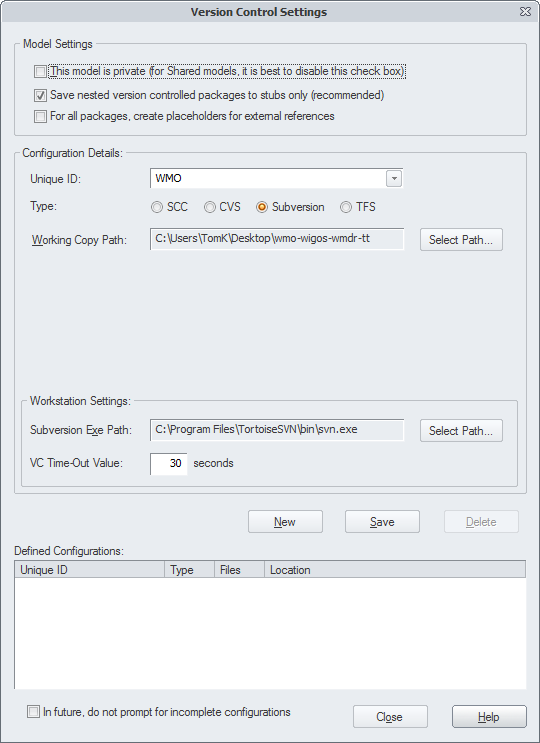
* Set up a reference to the working copy for each SVN repository
* ‘Get Package’ from the working copy for each model you wish to import

## Set up a reference to the WMO SVN working copy

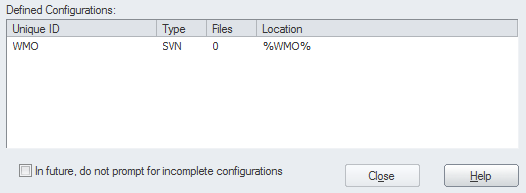
Open a new EA project and save it (dismiss any wizards).

Now select the menu item Configure/Settings

Complete the dialog box as shown in this screenshot:



* 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.

## 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://pl7.projectlocker.com/WMO/wmo-icao-models/svn/trunk>
* 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’

## Setting up the ISO TC211 configuration

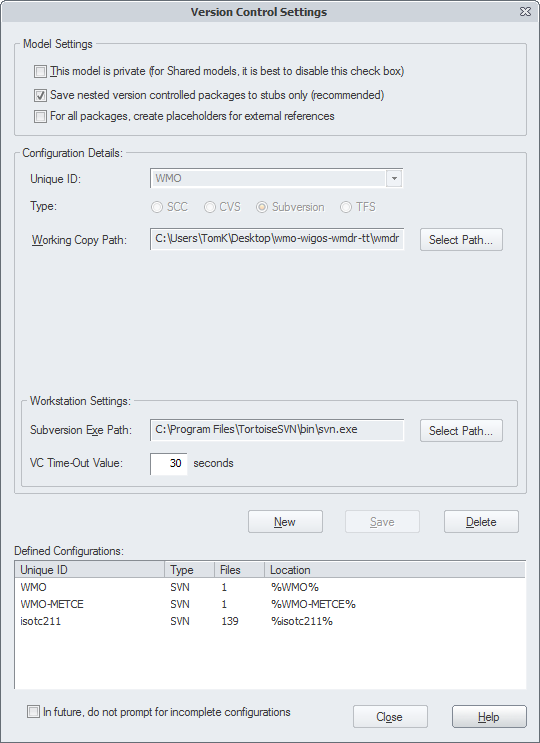
This is configured in the same way:

* SVN Checkout the ISO models to a directory on your computer using TortoiseSVN.
  + <https://www.seegrid.csiro.au/mirrors/iso-harmonized-model>

Alternatively, ISO TC211 models are accessible at

* + <https://inspire-twg.jrc.it/svn/iso>
* 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:

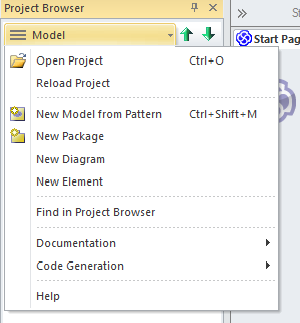


## Prepare the EA project structure (optional step).

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

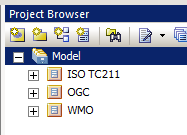
Although not essential, it is recommended that you create separate empty packages in EA for the ISO Models, OGC Models and the WMO models. This will make it easier to navigate later.

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

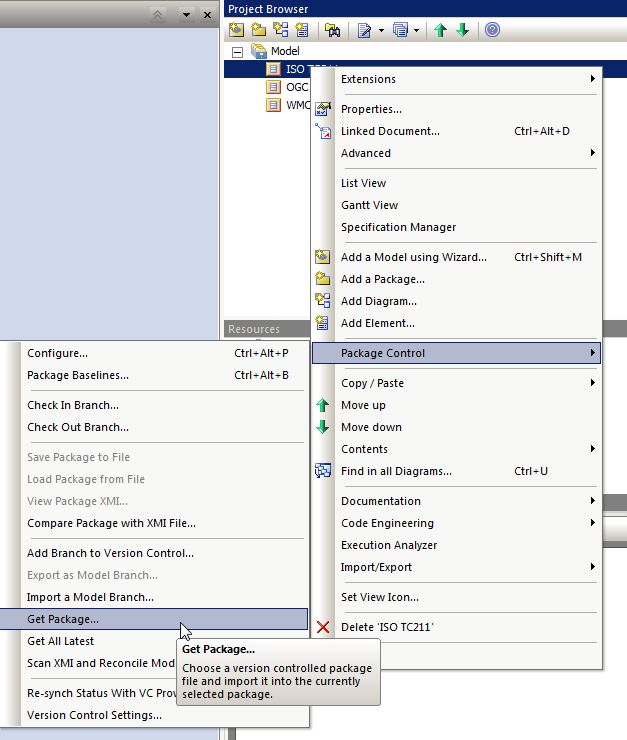


(Note: if you can’t see the Project Browser, select it from the View menu).

Your model should look like something this:



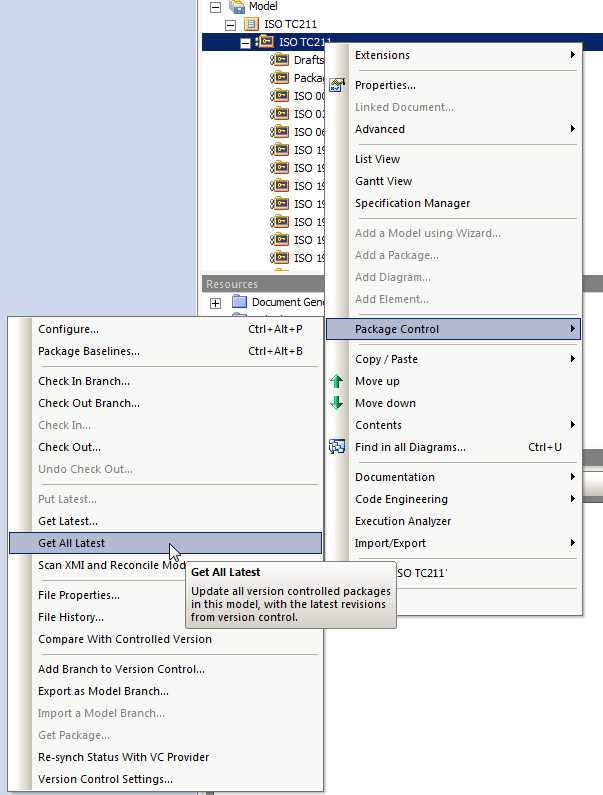
Now you can populate these packages from the repositories. It is important to use the **‘Get Package’** command for this operation (see below).

0

## 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.



## Get the WMO models

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

The active WMDR model is in trunk/xmi/wmdr.xml

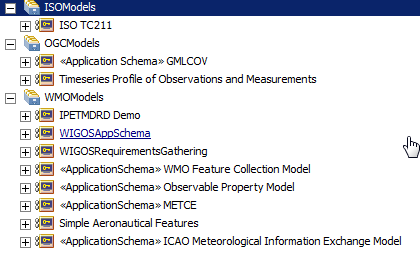
For METCE make sure you get the branch from the trunk in the ‘WMO-METCE’ configuration.

<https://pl7.projectlocker.com/WMO/wmo-icao-models/svn/trunk/releases/2.1/schema/schema/metce/1.2>

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.

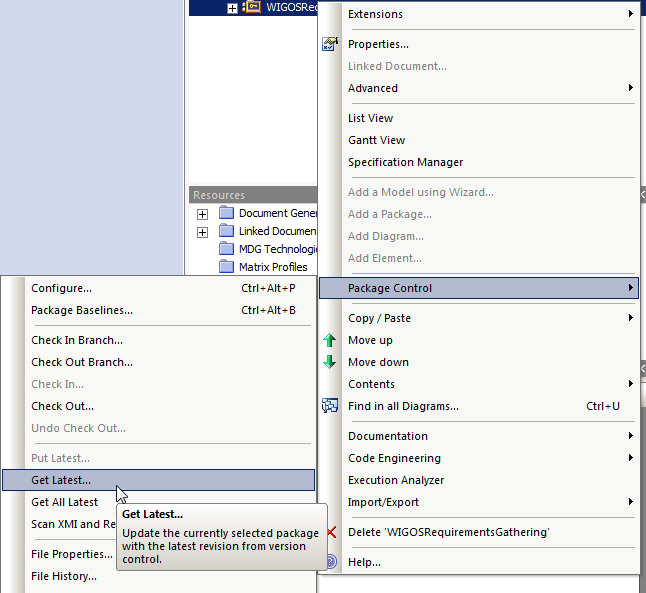
It will look similar to below but may not be identical.



# 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 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.

# Schema maintenance procedure

There are several steps to updating the WMDR schema, which include:

* Making model edits
* Regenerating the schema
* Committing updates to Subversion

These steps are described in more detail below, but it is also recommended to read the WMO Guide to Data Modelling (cf: http://wis.wmo.int/metce-uml).

## 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.

## 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.

## 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.

## 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.

## Regenerating the Schema from EA

Enterprise Architect can automatically generate the WMDR GML Application Schema.

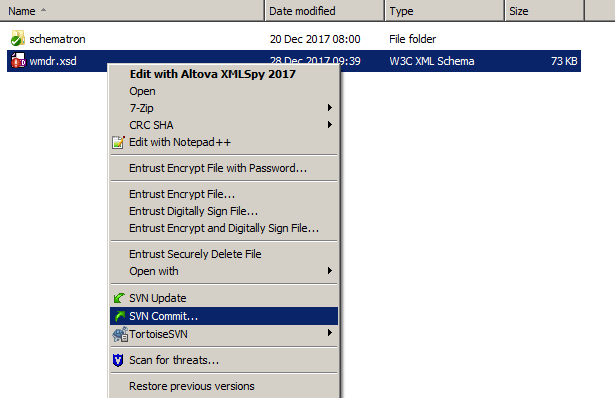
To do so, right click on the <<Application Schema>> WMDSchema package and select Extensions/GML/Generate GML Application Schema.

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

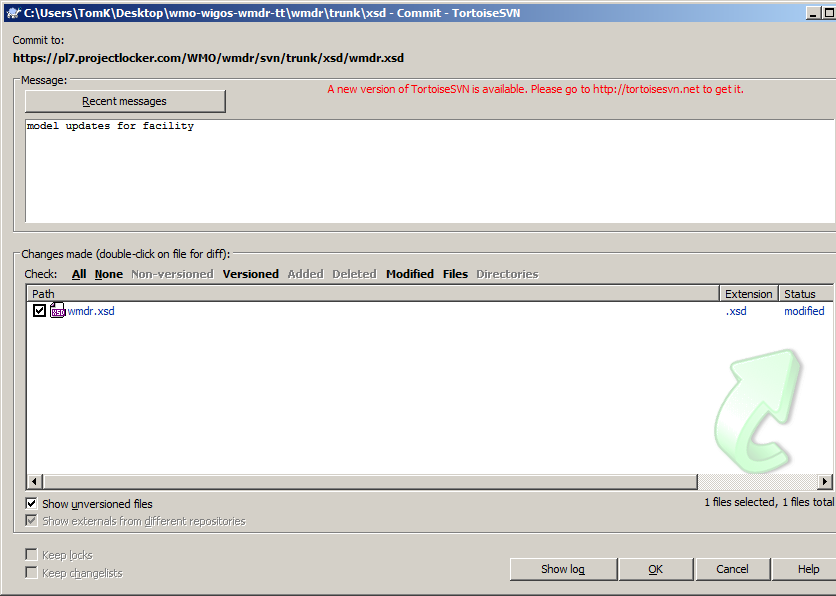
Generate the schema to overwrite the trunk/xsd/wmdr.xsd in your SVN working copy.

## Committing Updates to Subversion

Once the XML Schema has been updated, commit the updated schema to Subversion. In Windows Explorer, locate trunk/xsd/, right-click wmdr.xsd and select “SVN Commit”.



At this point the TortoiseSVN dialogue will appear. Enter a short description of the changes made to the model (resulting in the updated XML Schema) and ensure that wmdr.xsd is checked. Click OK. This will commit the updates made to the updated XML Schema to the SVN repository.



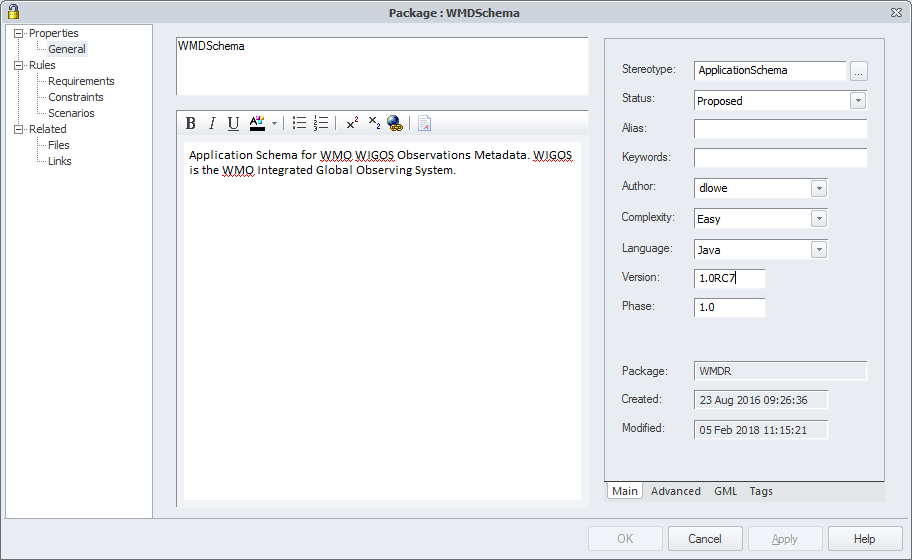
# Release Procedure

The following steps are required for creating a release of WMDR:

* Updating the version in the EA project
* Updating remaining repository artifacts
* Creating release tag in Subversion
* Generating model documentation
* Generating schema documentation
* Generating model and schema specification PDF
* Creating the release bundle

## Updating the version in the EA project

To ensure the correct version of the schema is provided in the resulting XML Schema, in the EA project browser right click the WMDR <<ApplicationSchema>> WMDSchema entry and select “Properties”. Update the Version property as shown below.



## Updating remaining repository artifacts

Ensure the following documents/files are up to date in trunk/ with the model and schema representation:

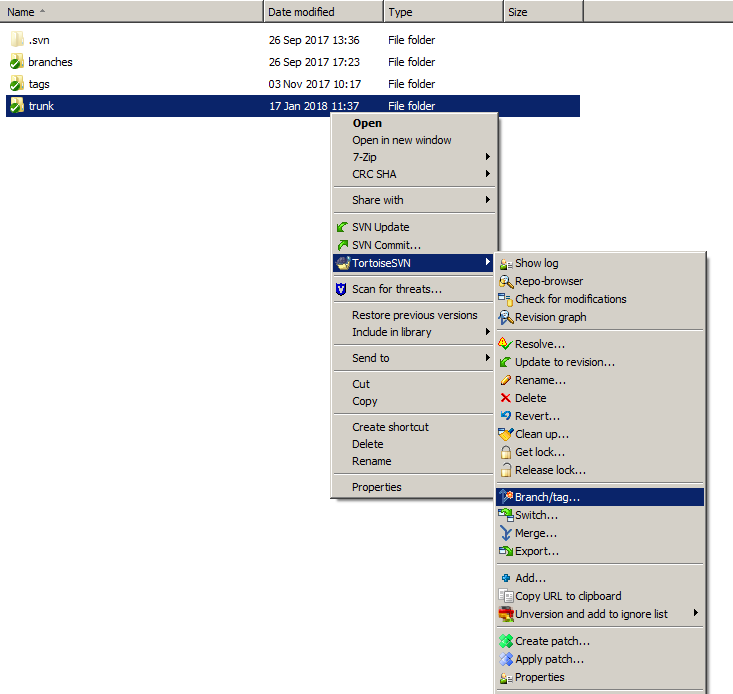
* documentation/WMDR\_ModelAndSchemaSpecification.docx: ensure model, versions and URLs are updated to reflect new version to be created
* examples/ (all files):
  + ensure all files validate against the XML Schema in xsd/wmdr.xsd
  + ensure the root element’s xsi:schemaLocation attribute points to the canonical XML Schema for the release (e.g. for 1.0RC7 use http://schemas.wmo.int/wmdr/1.0RC7/wmdr.xsd)

Following these updates, commit the necessary changes to the SVN repository.

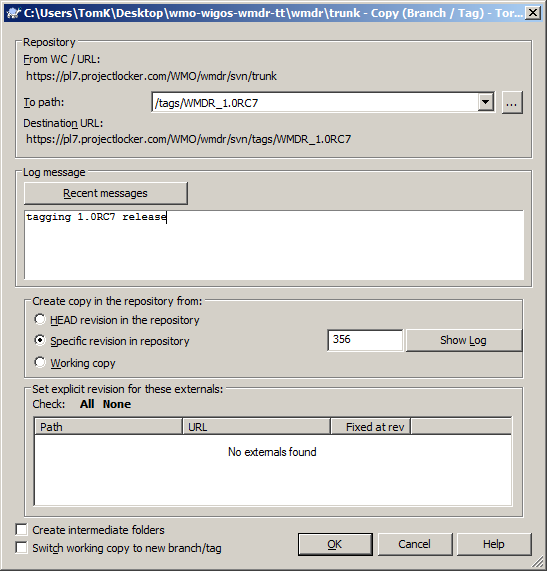
## Creating release tag in Subversion

At this point we are ready to create an official release tag of WMDR based on the current state of trunk/. Again, ensure all files as required are updated in SVN. Here we will use 1.0RC7 as our release version as the example.

In Windows Explorer, locate and right-click trunk/ and select “TortoiseSVN / Branch/Tag”.



The TortoiseSVN Branch/Tag dialogue box will appear. Ensure the “To path” text box reads “/tags/WMDR\_1.0RC7”/. Also add a Log message as per the below screenshot.



Click OK. The release tag is now created.

## Generating the HTML Documentation from EA

To generate the HTML model, right click on the application schema package and select Documentation/HTML Report. Ensure that the documentation is written to a directory that already exists. Click Ok. The documentation should then be available in the directory specified.

## Generating the Schema Documentation using Docflex

The docflex tool is used to generate schema documentation. More information on DocFlex can be found at <http://www.filigris.com/docflex-xml/generator.php>

Download the tool then run generator.bat. You may have to update the following line in generator.bat:

rem set JAVA\_HOME=C:\Program Files\Java\jre8

…to set your Java environment accordingly.

This provides a simple GUI where you select the schema to document (input field “XML files(s)”) as well as the output will be generated (input field “Output folder”). Generation takes a few minutes.

## Generating HTML schema documentation with XMLSpy

Select >Schem design > Documentation, accept defaults.

## Generating model and schema specification PDF

This step simply requires export of documentation/ WMDR\_ModelAndSchemaSpecification.docx to PDF using Microsoft Word’s File/Save As (Save As Type: PDF). The resulting filename should be [WMDR\_ModelAndSchemaSpecification.pdf](http://schemas.wmo.int/wmdr/1.0RC6/documentation/WMDR_ModelAndSchemaGuide.pdf)

## Creating the release bundle

Create a new Folder on your Desktop called WMDR-1.0RC7-release. Copy the following files / folders from your SVN repository tag at tags/WMDR-1.0RC7

* examples/ to WMDR-1.0RC7-release/examples/
* xsd/wmdr.xsd to WMDR-1.0RC7-release/wmdr.xsd
* WMDR\_ModelAndSchemaSpecification.pdf to WMDR-1.0RC7-release/documentation/
* the generated schema documentation to WMDR-1.0RC7-release/documentation/schemadoc
* the generated HTML documentation to WMDR-1.0RC7-release/html/

Create a zipfile of the folder WMDR-1.0RC7 and send to WMO for publication to schemas.wmo.int/wmdr. The zipfile needs to be extracted to <http://schemas.wmo.int/wmdr/1.0RC7/>.