Authors:

Dominic Lowe, Australian Bureau of Meteorology

Jörg Klausen, MeteoSwiss

Tom Kralidis, Meteorological Service of Canada

|  |  |  |
| --- | --- | --- |
| 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 |
| 2020-05-29 | Jörg Klausen | Use of EA15.1, cloud repositories and github instead of SVN, major changes to this document |

Table of Contents

[1. Introduction 3](#_Toc41660129)

[2. Install Enterprise Architect 3](#_Toc41660130)

[3. Install a git client 4](#_Toc41660131)

[4. Setup local working copies of the required model components 4](#_Toc41660132)

[4.1. METCE 4](#_Toc41660133)

[4.2. WMDR 5](#_Toc41660134)

[5. Initial Set up the WMDR model in Enterprise Architect 6](#_Toc41660135)

[5.1. Import the ISO TC211 Harmonized Model 6](#_Toc41660136)

[5.2. Import the METCE model 7](#_Toc41660137)

[5.3. Import the WMDR model 8](#_Toc41660138)

[6. Generate WMDR WMDR GML Application Schema 9](#_Toc41660139)

[7. Final setup steps 9](#_Toc41660140)

[8. Model development and schema maintenance 10](#_Toc41660141)

[8.1. Update the local working copy of the Github repository 10](#_Toc41660142)

[8.2. Update the EA model 10](#_Toc41660143)

[8.3. Regenerate the GML Application Schema 11](#_Toc41660144)

[8.4. Export the WMDR package as XMI 11](#_Toc41660145)

[8.5. Generate the HTML documentation 11](#_Toc41660146)

[8.6. Test the examples 11](#_Toc41660147)

[8.7. Update any other documentation as needed 11](#_Toc41660148)

[8.8. Commit changes and create a pull request 11](#_Toc41660149)

[9. Release Procedure 12](#_Toc41660150)

[10. Recreating earlier versions of the EA model 12](#_Toc41660151)

[10.1. Release 1.0RC9 12](#_Toc41660152)

[11. ---- Obsolete material beyond this point ---- 13](#_Toc41660153)

[12. Schema maintenance procedure 13](#_Toc41660154)

[12.1. Editing the model – Step 1: Get Latest 13](#_Toc41660155)

[12.2. Editing the model – Step 2: Check Out the Model 13](#_Toc41660156)

[12.3. Editing the model – Step 3: Make Edits to the Model 14](#_Toc41660157)

[12.4. Editing the model – Step 4: Check in changes to the Model 14](#_Toc41660158)

[12.5. Committing Updates to Subversion 14](#_Toc41660159)

[13. Install a Subversion Client 16](#_Toc41660160)

[13.1. Check the SVN installation 16](#_Toc41660161)

[13.2. Configure TortoiseSVN to work with your proxy 17](#_Toc41660162)

[14. Setup version control 18](#_Toc41660163)

[15. Final setup step: Update the models 24](#_Toc41660164)

[16. Register to access the model repository. 25](#_Toc41660165)

[17. Release Procedure 26](#_Toc41660166)

[17.1. Updating the version in the EA project 26](#_Toc41660167)

[17.2. Updating remaining repository artifacts 26](#_Toc41660168)

[17.3. Creating release tag in Subversion 27](#_Toc41660169)

[17.4. Generating the HTML Documentation from EA 29](#_Toc41660170)

[17.5. Generating the Schema Documentation using Docflex 29](#_Toc41660171)

[17.6. Generating HTML schema documentation with XMLSpy 30](#_Toc41660172)

[17.7. Generating model and schema specification PDF 30](#_Toc41660173)

[17.8. Creating the release bundle 30](#_Toc41660174)

[a. Create local SVN working copy of WMDR model 31](#_Toc41660175)

# Introduction

The WIGOS Metadata Representation (WMDR) is the reference implementation of the WIGOS Metadata Standard. The WMDR is developed using **SPARX Enterprise Architect (EA) version 15.1** and is distributed as a GML Application Schema, a specialized XML Schema Definition (XSD). EA is used in conjunction with a GitHub[[1]](#footnote-1). Unfortunately, EA is not integrated with GitHub, so there is no direct version control of the model from within EA. This requires a slightly more complex work-flow and coordination amongst the collaborators on the model.

It is assumed there is a primary model editor who maintains the GitHub repository. Having multiple editors working on the model can create synchronisation and ‘locking’ problems that can be difficult to resolve. If you would like to contribute to the model development, please carefully follow the procedures outlined in Section 8. Also, contact the maintainer(s) of the GitHub repository before doing any work.

This document explains how to:

* Install Enterprise Architect
* Install a local GitHub repository
* Fetch the model(s) from the repository
* Modify the model (e.g. add new attributes)
* Regenerate the XML Schema
* Update the repository.

# Install Enterprise Architect

Obtain at least the ‘Professional’ version of Enterprise Architect from

<http://www.sparxsystems.com.au/products/ea/purchase.html>.[[2]](#footnote-2)

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

# Install a git client

The models are stored in a GitHub repository at <https://github.com/wmo-im/wmdr>. Because EA does not directly support versioning with git, some care needs to be taken when developing the model.

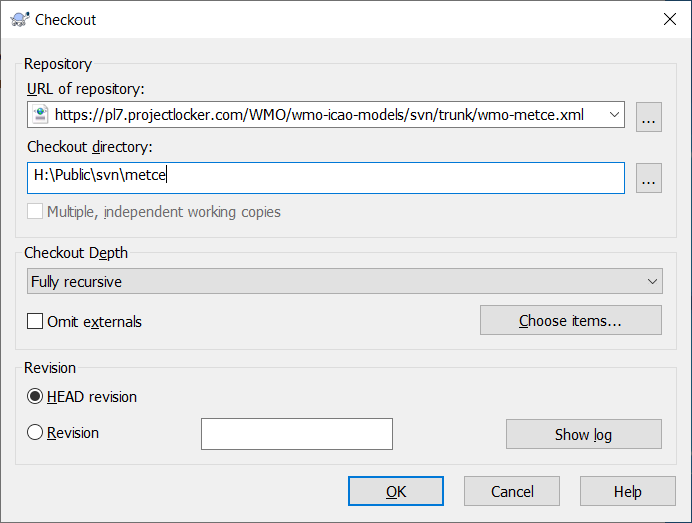
Install a Git client like TortoiseGit from <https://tortoisegit.org/download/>. Accept all defaults during the installation.

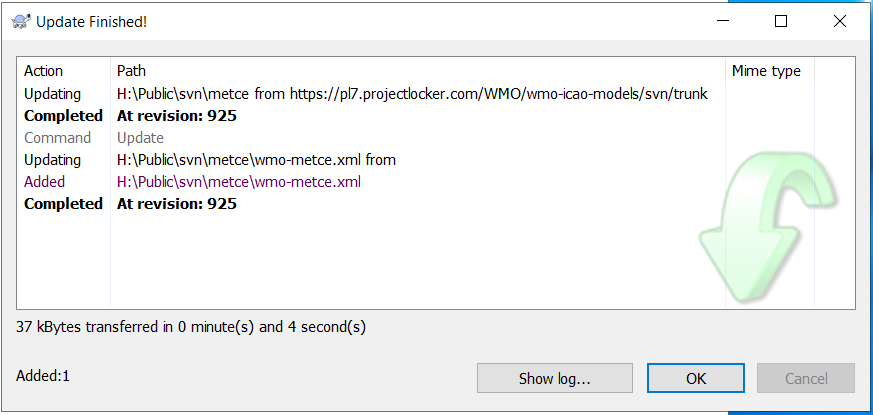
# Setup local working copies of the required model components

In Windows Explorer, create a directory ‘svn’ and two directories ‘metce’ and ‘wmdr’ in it.

## METCE

Right-click on the ‘metce’ folder and navigate to the ‘SVN checkout’ context menu item. Fill in the URL of the repository as shown and click OK.

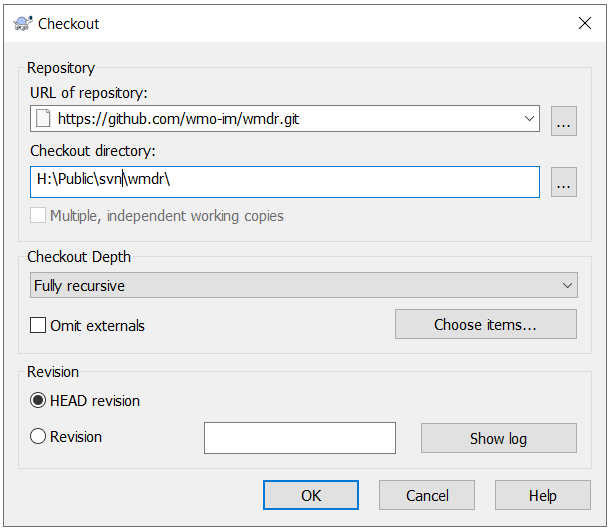




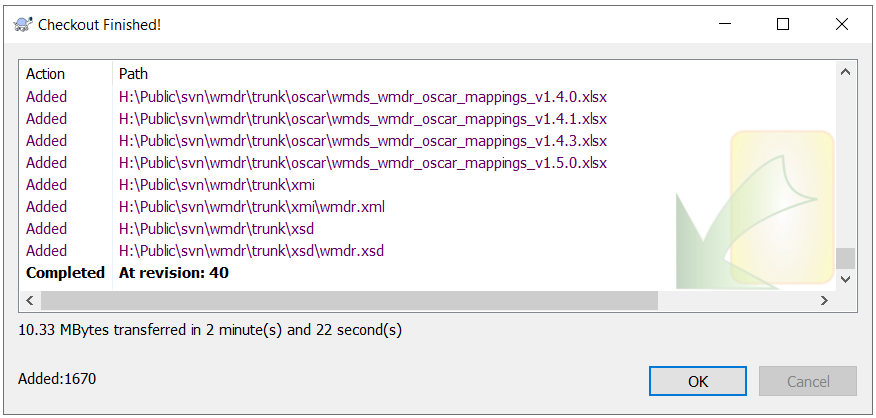
## WMDR

Right-click on the ‘wmdr’ folder and navigate to the ‘SVN checkout’ context menu item. Fill in the URL of the repository as shown and click OK.

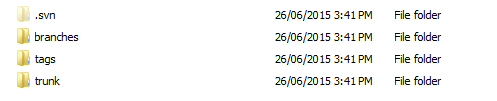
In the dialogue that follows, enter the URL of the GitHub repository as follows.



Click OK to create a local SVN copy of the remote GitHub repository. At the end of this process, you should get a confirmation similar to the following.



As a result, you should also see a set of new folders (you may or may not see the hidden *.svn* folder depending on your Windows folder settings).



# Initial Set up the WMDR model in Enterprise Architect

**NB:** These steps only need to be done once by the designated model editor. Other users/editors of the model should fetch the model from GitHub and open it in EA.

Open EA and select ‘Create a new project’, save it on a local drive as ‘wmdr.eapx’ .

Highlight the ‘Model’ in the EA Browser space and select ‘Design’ > ‘Add’ > ‘Package’.

Call it ‘ISO’.

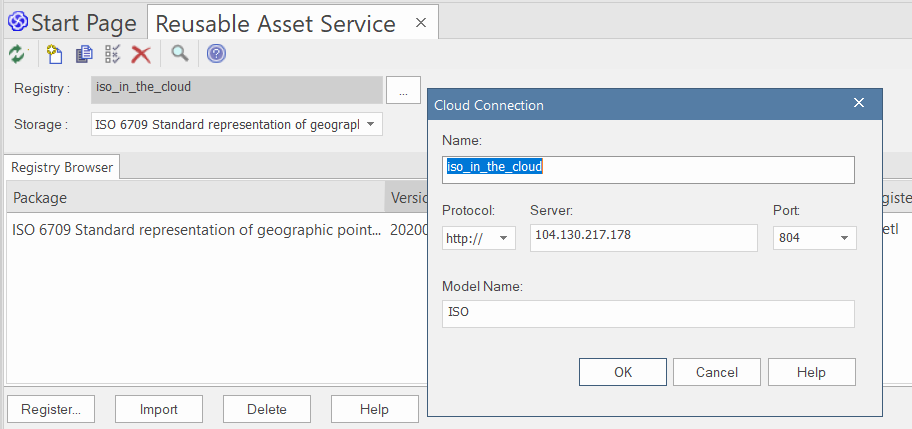
Repeat the process and create 2 more packages called ‘METCE’ and ‘WMDR’, respectively.

The WMDR model uses the ISO 19156 Observation & Measurements Model as a base. It also used to use the WMO METCE model. These components as well as the actual WMDR model need to be imported.

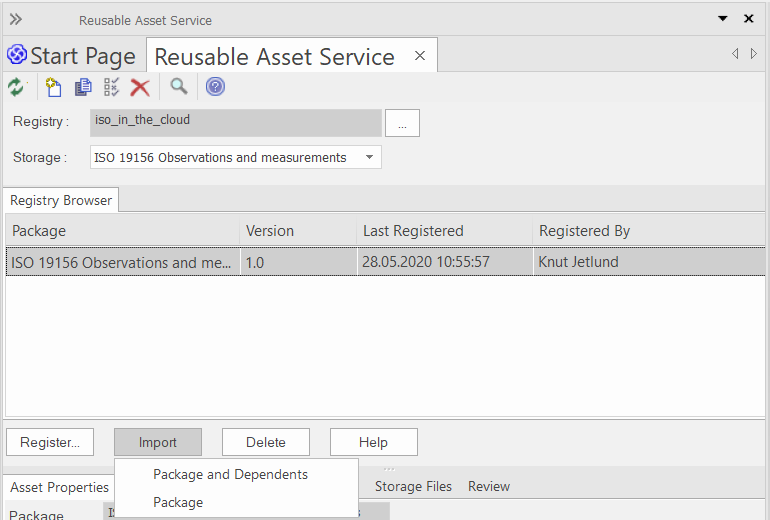
## Import the ISO TC211 Harmonized Model

In EA, highlight the ‘ISO’ package and select Publish > Reusable Assets > Import. To connect to the asset service: See the user guide for Reusable Assets[[3]](#footnote-3). Apply the following parameters for the Registry:

* Name: iso\_in\_the\_cloud
* Protocol: http
* Server: 104.130.217.178
* Port: 804
* Model Name: iso

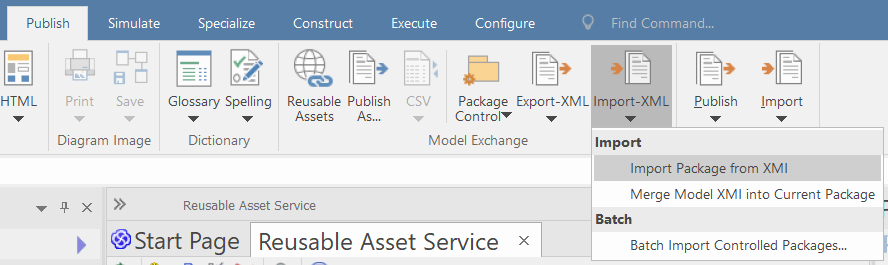


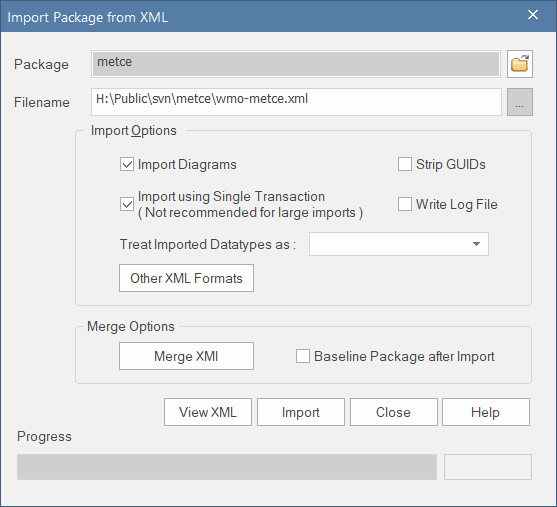
Select the ISO 19156 under ‘Storage’ and Click ‘Import’ > ‘Package’. One storage is created for each main standard number, containing all parts and editions.



## Import the METCE model

In EA, highlight ‘METCE’ and navigate to ‘Publish’ > ‘Import XML’.

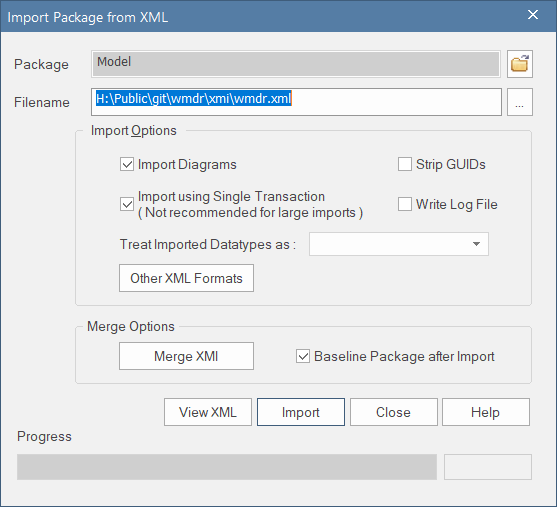




Click on ‘Import’.

## Import the WMDR model

In EA, highlight the root model ‘Model’ and navigate to ‘Publish’ > ‘Import XML’.

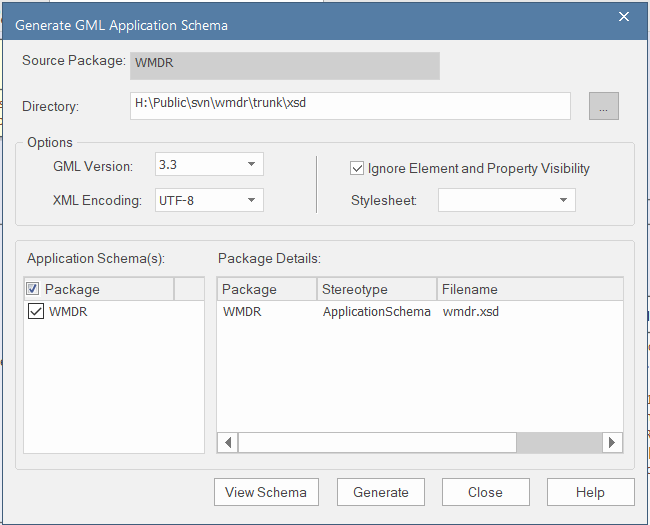


Click ‘Import’, then select ‘No’ when prompted to place the model at the root level.

# Generate WMDR WMDR GML Application Schema

Enterprise Architect can automatically generate the WMDR GML[[4]](#footnote-4) Application Schema[[5]](#footnote-5). http://www.opengeospatial.org/standards/gml

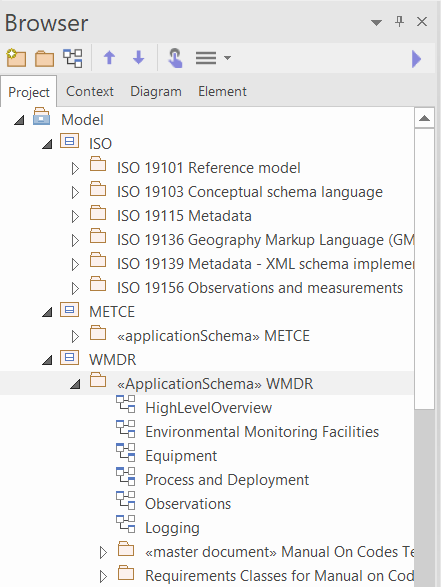
To do so, highlight the ‘<<Application Schema>> WMDR’ package, select ‘Specialize’ > > ‘Generate GML Application Schema’ and fill in the form as shown. If you wish, double-click on the Filename to choose another name for the .xsd file.



# Final setup steps

The EA project files can get relatively large. To check integrity and reduce file size, select ‘Configure’ > ‘Integrity’ > ‘Manage .EAPX/.EAP File’ > ‘Compact .EAPX/.EAP File’.

The final model should look something like the following in the Browser.



# Model development and schema maintenance

Several steps are involved in updating the model and the GML schema:

* Fetch an up-to-date version of the model from the Github repository
* In EA, edit the UML model
* In EA. Regenerate the GML schema
* Commit updates to the GitHub repository and create a Pull request.

## Update the local working copy of the Github repository

The exact procedures depend on your GitHub client, but the normal process involves

* $ git pull [--rebase]

This will fetch the latest changes from the remote repository [and optionally rebase the local working copy].

## Update the EA model

In general, modelling work should be done in a branch. There are a few commands to work with branches as follows.

* $ git branch –av (to list all branches, local and remote)
* $ git branch <my-branch> (create a new branch [if needed])
* $ git checkout <my-branch> (switch to a branch, and update working directory)

From your local GitHub repository, open the **/eamodel/wmdr.eapx** file from within EA.

The model can be edited to make changes. It is strongly recommended to read the WMO Guide to Data Modelling[[6]](#footnote-6).

Importantly, 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 element order in the will change.

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.

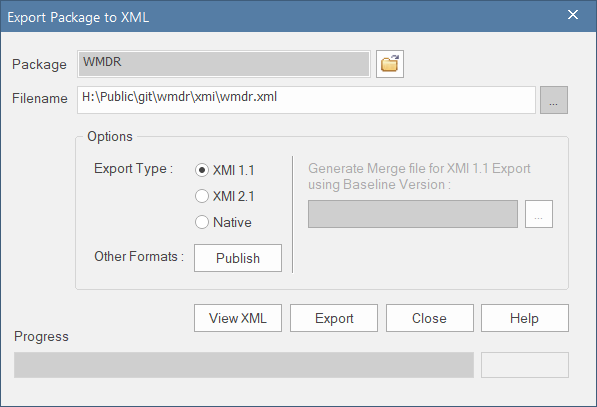
For more complex changes like adding new classes please consult the WMO Guide to Data Modelling.

## Regenerate the GML Application Schema

See Section 6.

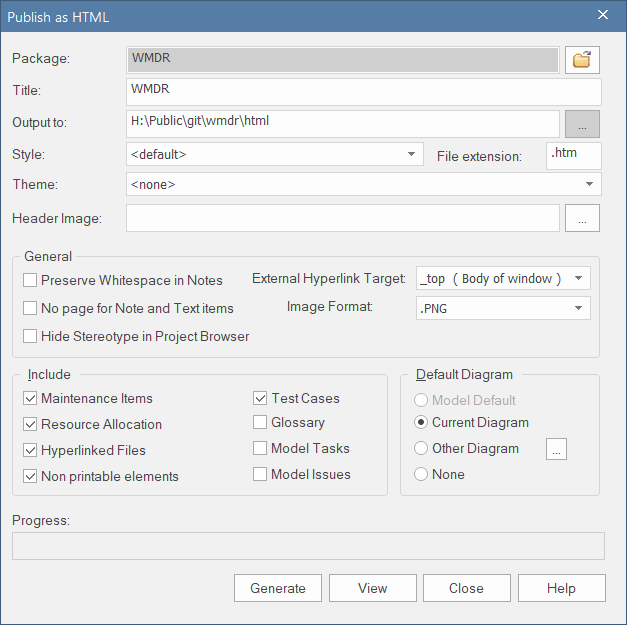
## Export the WMDR package as XMI

In EA, select ‘Publish’ > ‘Export XML’.



## Generate the HTML documentation

In EA, select ‘Publish’ > ‘HTML’.



## Test the examples

tbd

## Update any other documentation as needed

tbd

## Commit changes and create a pull request

Once all editing work is completed,

* In EA, compress the model (cf. Section 7), then quit EA.

Navigate to your local GitHub repository, stage all changed files and commit your changes.

* $ git add .
* $ git commit –m “commit message”

Push your local changes to the remote GitHub repository (=pull request).

* $ git push

# Release Procedure

The maintainer(s) of the remote repository will decide if the branch can be merged into Master and delete <my-branch> eventually. The maintainer(s) will also create a ‘release’ by tagging the Master branch.

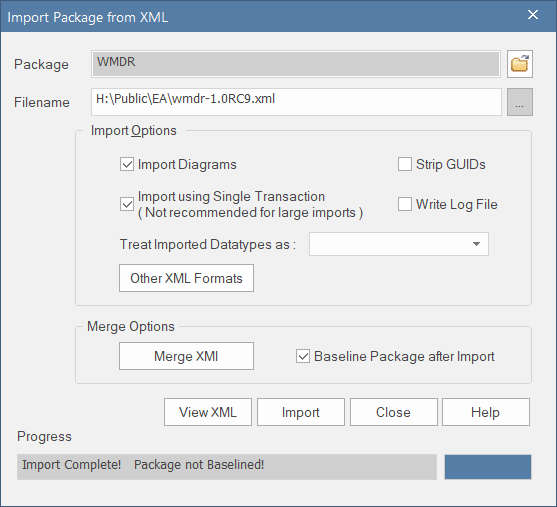
# Recreating earlier versions of the EA model

## Release 1.0RC9

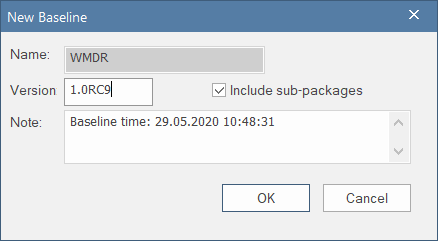
Extract the xmi/wmdr.xml file to a local place from the GitHub release 1.0RC9 and rename it to wmdr-1.0RC9.xml.

Also, extract the xsd/wmdr.xsd file to a local place from the GitHub release 1.0RC9 and rename it to wmdr-1.0RC9.xsd.

Create a root package, then Publish > Import XML with the following settings.



When prompted, create a baseline as follows.

.

Verify the model by generating the XSD file. In EA, select ‘Specialize’ > > ‘Generate GML Application Schema’. Save the XSD as wmdr-1.0RC9-recreated.xsd.

The wmdr-1.0RC9.xsd and the newly recreated wmdr-1.0RC-recreated.xsd files must be almost the same, the only differences being

* different versions of Enterprise Architect (line is a comment) – Line 2
* the location of the block describing wmdr:description (Lines 79-92 in the schema from release 1.0RC9; Lines 21-34 in the recreated schema)

1. <https://github.com/wmo-im/wmdr> [↑](#footnote-ref-1)
2. A fully functional 30 day trial can be downloaded from [www.sparxsystems.com.au/products/ea/trial.html](http://www.sparxsystems.com.au/products/ea/trial.html) [↑](#footnote-ref-2)
3. <https://sparxsystems.com/enterprise_architect_user_guide/15.0/model_repository/connect_to_asset_service.html> [↑](#footnote-ref-3)
4. http://www.opengeospatial.org/standards/gml [↑](#footnote-ref-4)
5. https://sparxsystems.com/enterprise\_architect\_user\_guide/15.1/model\_domains/generate\_gml\_application\_schem.html [↑](#footnote-ref-5)
6. (include reference) [↑](#footnote-ref-6)