

Upgrading to 1.4.0.0 from previous version

If you already have a model that is compliant to OpenMI 1.2 and this model is using the `org.OpenMI.Utilities.wrapper`, you can upgrade your model by following the steps below:

1. Uninstall OpenMI 1.2 or any other previous OpenMI releases on your computer.
2. Download the OATC SDK from <http://sourceforge.net/projects/openmi/>.
3. Unzip the files.
4. If you do not already have NUnit installed, it is recommended that you install it from <http://nunit.org/>.
5. Open the `Oatc.OpenMI.Sdk.sln` solution and rebuild all.
6. Add your existing wrapper project to the `Oatc.OpenMI.Sdk.sln` (also add unit test project, if you have that).
7. In your wrapper project (and unit test project if you have that) remove the reference to `org.OpenMI.Standard` and subsequently add a reference to the `OpenMI.Standard.dll` using the Browse tab in the Add Reference dialog in Visual Studio. The `OpenMI.Standard.dll` is located in `<yourUnzipfolder>\OpenMI.Standard\src\csharp\bin`.
8. In your wrapper project (and unit test project if you have that) references to `org.OpenMI.Backbone`, `org.OpenMI.DevelopmentSupport`, `org.OpenMI.Utilities.Wrapper`, `org.OpenMI.Utilities.Spatial` and `org.OpenMI.Utilities.Buffer` should be replaced by references to the projects `Oatc.OpenMI.Sdk.Backbone`, `Oatc.OpenMI.Sdk.DevelopmentSupport`, `Oatc.OpenMI.Sdk.Wrapper`, `Oatc.OpenMI.Sdk.Spatial` and `Oatc.OpenMI.Sdk.Buffer`, respectively. All 'using statements' in your source code should be changed accordingly.
9. If you are using fully-qualified class names for classes from the SDK or the `OpenMI.Standard`, these should also be changed (use search and replace). For example, if you have a class declared as `org.OpenMI.Utilities.Wrapper.LinkableEngine` this should be changed to `Oatc.OpenMI.Sdk.Wrapper.LinkableEngine`. Note that when changing fully-qualified references to interfaces in the `OpenMI.Standard` you may have to use the 'global' keyword; for example, `org.OpenMI.Standard.IDimension` is changed to `global::OpenMI.Standard.IDimension`.
10. Rebuild all.
11. If you are using unit test, run these now.
12. Create an `OpenMIComplianceInfo.xml` file for your component. This file must be verifiable with the schema: <http://www.openmi.org/schemas/OpenMIComplianceInfo.xsd>. You can find an example of such file for the simple river example, which is included in the SDK release (see `<yourUnzipFolder>\Oatc\src\csharp\Examples\ModelComponents\SimpleRiver.simpleRiver.xml`). Other examples of openmi compliance xml file can be found [here](#).
13. Mail the `OpenMIComplianceInfo.xml` file to `Association@openmi.org`, with a request to have your component officially accepted as OpenMI-compliant.

Note that the steps above are just one possible way to upgrade your code. You may want to organize things differently and you may have tools that will help you refactor your code.

With the OpenMI 1.4 release you will, as a model provider, have to provide all required assembly files (also the `OpenMI.Standard.dll` and `Oatc.OpenMI.Sdk DLLs`). For this reason it is recommended (but not required) that you change the `Oatc.OpenMI.Sdk` namespaces to, for example, `<MyCompany>.OpenMI.Sdk.*`. You may or may not choose to install your assemblies into the GAC. If you choose the latter you must sign all the assemblies with your own signature file.