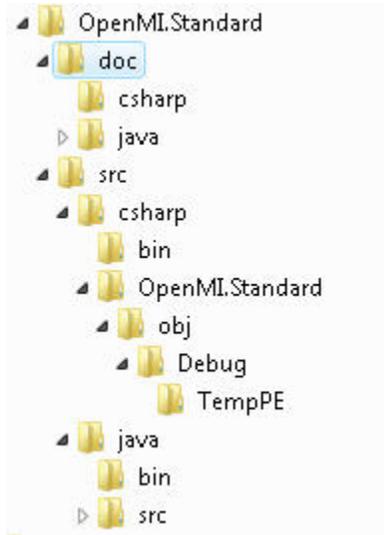


How is the OpenMI standard defined

In order to avoid any ambiguities the only official definition of the OpenMI standard is provided with the OpenMI standard release. ([Download the OpenMI Standard](#)).

The OpenMI standard release is a zipped directory structure, which contains the java and the C# code of the implemented interfaces, the compiled library files, and the standard definition document.



The entry point for the standard definition is the *OpenMI.Standard.Definition.htm* file, which is also shown below:

OpenMI.Standard.Definition.htm

OpenMI standard definition:

There are two variants of OpenMI compliance. Components can be either OpenMI 1.4 .Net compliant or OpenMI 1.4 Java compliant.

OpenMI .Net compliant components must follow the compliance definition given in the comments in the file *ILinkableComponent.cs*

OpenMI Java compliant components must follow the compliance definition given in the comments in the file *ILinkableComponent.java*

In the C# code the following statements serve as the key requirements for OpenMI compliance:

source code comments from the *ILinkableComponent.cs* file

- § 1) An OpenMI compliant component must implement the *ILinkableComponent* interface according to specifications provided as comments in the *OpenMI.Standard* interface source code.
- § 2) An OpenMI compliant component must, when compiled, reference the *OpenMI.Standard.dll*, which is released and compiled by the OpenMI Association
- § 3) An OpenMI compliant component must be associated with a XML file, which complies to (can be validated with) the *LinkableComponent.xsd* schema.
- § 4) An OpenMI compliant component must be associated with a XML file, which complies to (can be validated with) the *OpenMIComplianceInfo.xsd* schema. This file must be submitted to the OpenMI Association.
- § 5) The OpenMI Association provides two additional interfaces which OpenMI compliant components may or may not implement; the *IManageState* interface and the *IDiscreteTimes* interface. However, if these interfaces are implemented, each method and property must implemented according to the comments given in the *OpenMI.Standard* interface source code.
- § 6) The OpenMI Association's downloadable standard zip file provides the only recognized version of source files, xml schemas and assembly file.

Similar rules apply to the Java implementation.

The OpenMI Standard consists of 4 enumerations and 24 interfaces which all together defines 96 methods or properties. Each of these methods /properties is unambiguously defined in the source code of the release.

As an example the documentation for one of these 96 methods is given below

Source code for the `OpenMI.Standard.ILinkableComponent.GetInputExchangeItem(...)` method

```
/// <summary>
/// <para>Returns the InputExchangeItem corresponding to the inputExchangeItemIndex
/// provided in the method arguments.</para>
///
/// <para>This method must be accessible after the Initialize( ) method has been
/// invoked and until the Prepare( ) method has been invoked. If this property
/// is accessed before the Initialize( ) method has been invoked or after the
/// Prepare( ) method has been invoked and the LinkableComponent cannot handle
/// this an exception must be thrown.</para>
///
/// <para>This method basically returns a reference to the InputExchangeItem object.
/// There is no guarantee that this object is not altered by other components
/// after it has been returned. It is the responsibility of the LinkableComponent
/// to make sure that such possible alterations does not subsequently corrupt
/// the LinkableComponent.</para>
/// </summary>
///
/// <param name="inputExchangeItemIndex">
/// Identifies the index-number of the requested InputExchangeItem (indexing starts from zero)
/// This method must accept values of inputExchangeItemIndex in the interval
/// [0, InputExchangeItemCount - 1]. If the inputExchangeItemIndex is outside this
/// interval an exception must be thrown.</param>
///
/// <returns>
/// The InputExchangeItem as identified by inputExchangeItemIndex.
/// </returns>

InputExchangeItem GetInputExchangeItem(int inputExchangeItemIndex);
```

Clearly the standard definition is neither amusing or easy reading. The main purpose of the standard definition is to provide unambiguous definitions in case of doubts for how to implement specific methods. In order to understand the basic concept of OpenMI or to learn how to use OpenMI in different contexts reading the documents under the heading "OpenMI for developers" or "OpenMI for modelers" is recommended.

An overview of the OpenMI standard interfaces is shown on the figure below. Click on the figure for better viewing

