

HOW TO work with OMI files

OpenMI systems can be considered software systems that combine a set of OpenMI- compliant components, possibly in addition to non-OpenMI- compliant components. Such a system can deploy and run OpenMI components by accessing them through their standard interface. The system however, also needs to be able to locate and instantiate the openMI component properly. For that purpose, the OMI file has been defined.

The OpenMI file is an XML file which contains sufficient information to identify a component, instantiate the binary unit on your machine (i.e. find the assembly and the class to instantiate) and populate it with input data. An XML schema definition has been created to enable default tools to parse the information.

In principle, OMI files can reside anywhere on your system. Users are therefore free to organize their own repository, as long as they can find the relevant OMI files themselves when configuring their model combination.

An example OMI file is given in Figure 1. The underlying schema definition (XSD) is provided in Figure 2 and 3.

```
<?xml version="1.0"?>
<LinkableComponent Type="Oatc.OpenMI.Examples.ModelComponents.SpatialModels.GroundWaterModel.GWModelLC"
Assembly="..\GroundWaterModel\bin\Debug\Oatc.OpenMI.Examples.ModelComponents.SpatialModels.GroundWaterModel.dll"
>
  <Arguments>
    <Argument Key="modelID" ReadOnly="true" Value="GW local" />
    <Argument Key="simulationStart" ReadOnly="true" Value="1990,1,2,0,0" />
    <Argument Key="simulationEnd" ReadOnly="true" Value="1993,1,1,0,0" />
    <Argument Key="nx" ReadOnly="true" Value="20" />
    <Argument Key="ny" ReadOnly="true" Value="30" />
    <Argument Key="ox" ReadOnly="true" Value="3000" />
    <Argument Key="oy" ReadOnly="true" Value="3000" />
    <Argument Key="cellSize" ReadOnly="true" Value="125" />
    <Argument Key="groundWaterLevel" ReadOnly="true" Value="0.5" />
    <Argument Key="gridAngle" ReadOnly="true" Value="20.0" />
  </Arguments>
</LinkableComponent>
```

Figure 1 OMI file example

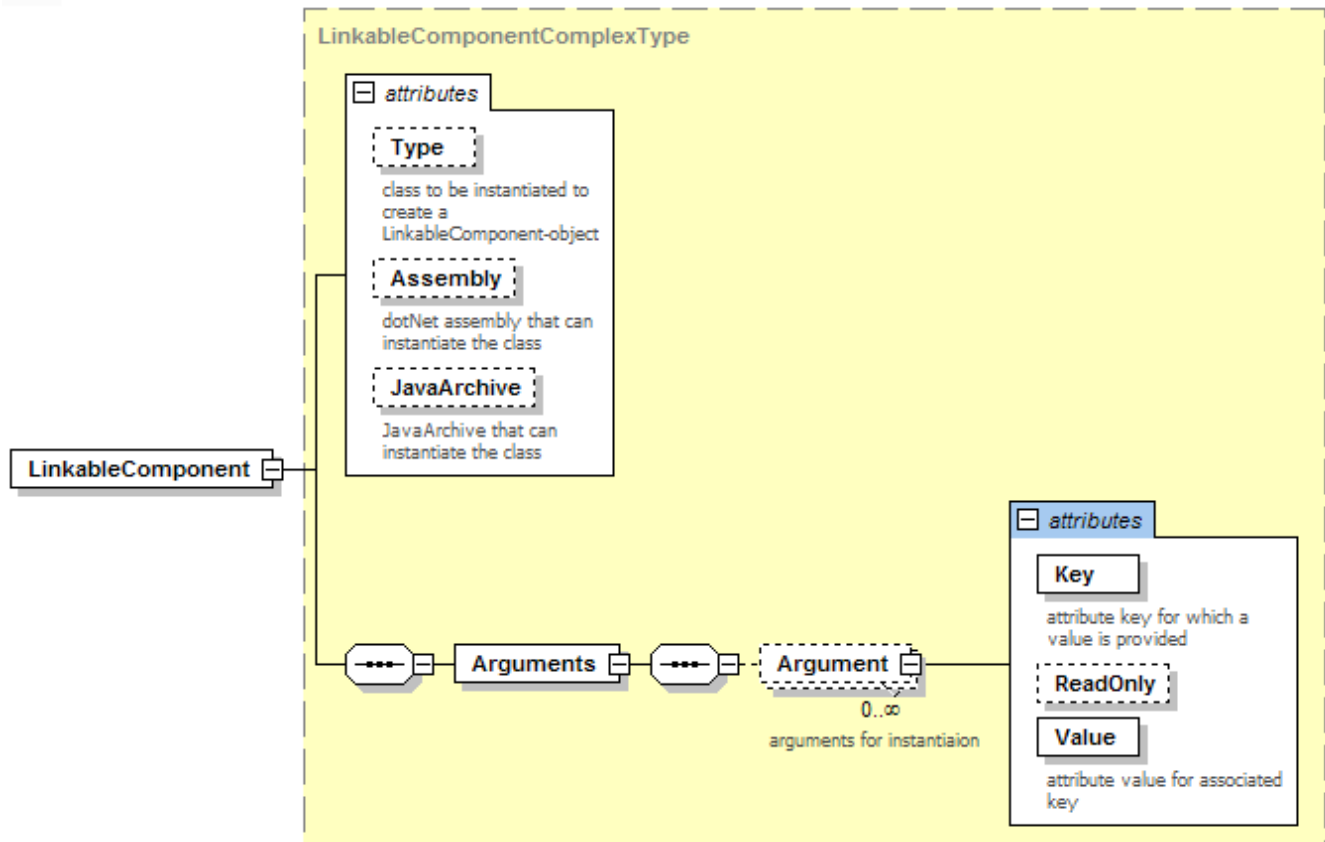


Figure 2 XML schema definition of the OMI file (graphical)

```

<?xml version="1.0"?>
<!-- OpenMI Linkable component entry point to instantiate the object-->
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns="http://www.openmi.org" targetNamespace="
http://www.openmi.org" elementFormDefault="qualified" attributeFormDefault="unqualified" version="1.4.0.0">
  <xsd:element name="LinkableComponent" type="LinkableComponentComplexType"/>
  <xsd:complexType name="LinkableComponentComplexType">
    <xsd:sequence>
      <xsd:element name="Arguments">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="Argument" minOccurs="0" maxOccurs="unbounded"
>
              <xsd:annotation>
                <xsd:documentation>arguments for instantiaion<
/xsd:documentation>

              </xsd:annotation>
              <xsd:complexType>
                <xsd:attribute name="Key" type="xsd:string"
use="required" form="unqualified">
                  <xsd:annotation>
                    <xsd:documentation>attribute
key for which a value is provided</xsd:documentation>

                  </xsd:annotation>
                </xsd:attribute>
                <xsd:attribute name="ReadOnly" type="xsd:
boolean" use="optional" form="unqualified"/>
                <xsd:attribute name="Value" type="xsd:string"
use="required" form="unqualified">
                  <xsd:annotation>
                    <xsd:documentation>attribute
value for associated key</xsd:documentation>

                  </xsd:annotation>
                </xsd:attribute>
              </xsd:complexType>
            </xsd:element>
          </xsd:sequence>
        </xsd:complexType>
      </xsd:element>
    </xsd:sequence>
    <xsd:attribute name="Type" type="xsd:string" form="unqualified">
      <xsd:annotation>
        <xsd:documentation>class to be instantiated to create a LinkableComponent-
object</xsd:documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="Assembly" type="xsd:string" use="optional" form="unqualified">
      <xsd:annotation>
        <xsd:documentation>dotNet assembly that can instantiate the class</xsd:
documentation>
      </xsd:annotation>
    </xsd:attribute>
    <xsd:attribute name="JavaArchive" type="xsd:string" use="optional" form="unqualified">
      <xsd:annotation>
        <xsd:documentation>JavaArchive that can instantiate the class</xsd:
documentation>
      </xsd:annotation>
    </xsd:attribute>
  </xsd:complexType>
</xsd:schema>

```

Figure 3 XML schema definition of the OMI file

💡 Many model applications are designed so to user will enter data needed for the model run through the application user interface and subsequently save all this into files or a database. Implementing some code that will create and save the corresponding OMI file when the user hits the save button can be a very useful and simple way to ensure that such models automatically becomes available as OpenMI compliant models and directly can be added to OpenMI configurations e.g. using the OpenMI configuration editor.