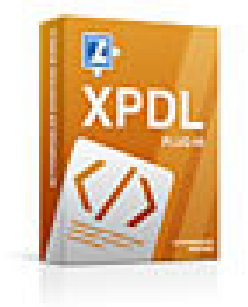
ActiveModeler Avantage



XPDL Plugin Users Guide

KAISHA TEC

Sangyo Plaza Annex 3-32-3 Shimo Renjaku,

Mitaka-shi Tokyo, 181-0013 JAPAN

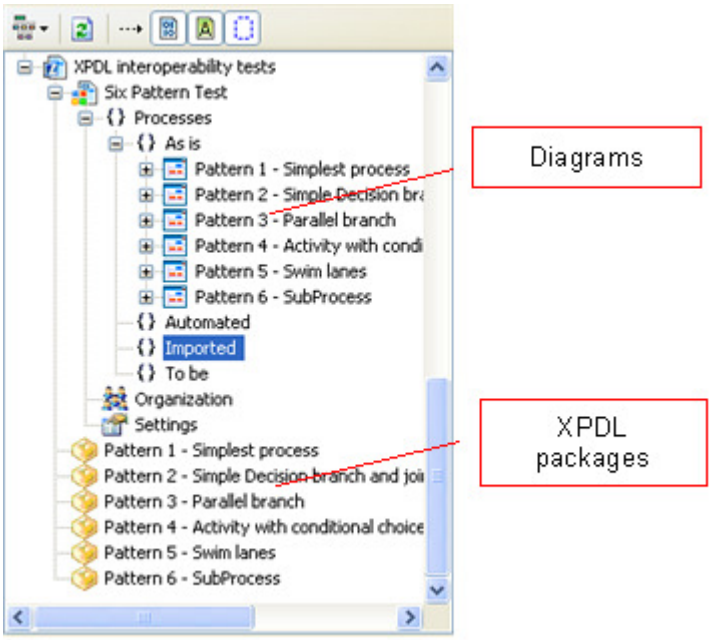
Document Revision:	1.101
Date:	July 1, 2007
Printed	12:56:06 AM

XPDL plugin

The XPDL plugin provides input and output of XPDL 2.0 and limited support for XPDL 1.0.

About the XPDL plugin

The XPDL plugin reads and writes XPDL 2.0 files. The XPDL plugin adds an **`.xpdl`** Package file type to the workspace.

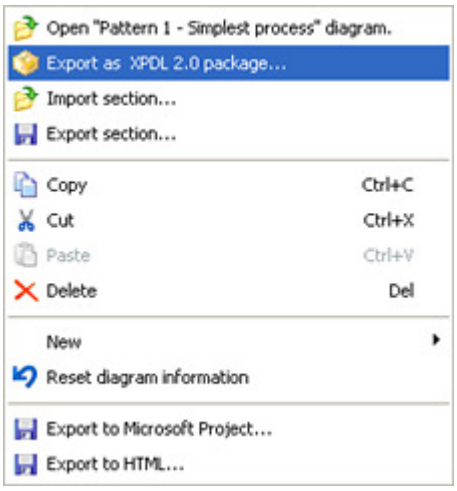


You can view and edit **`.xpdl`** files by double-clicking on them in which case they are edited in XML syntax.

To export an XPDL package

Right-click on the diagram you wish to export and then choose the **Export as XPDL 2.0 package...** menu.

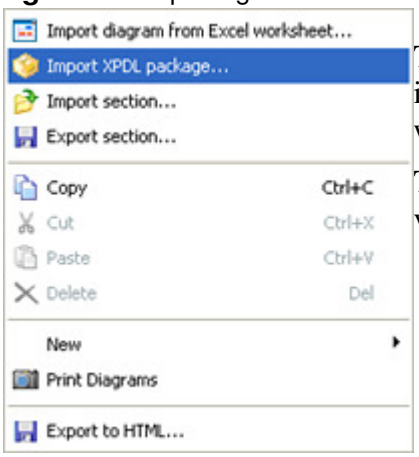
Figure 1. Exporting to XPDL



To import an XPDL package

Right-click on a MeasurePoint in a process model and then choose the **Import XPDL package...** menu item.

Figure 2. Importing from XPDL



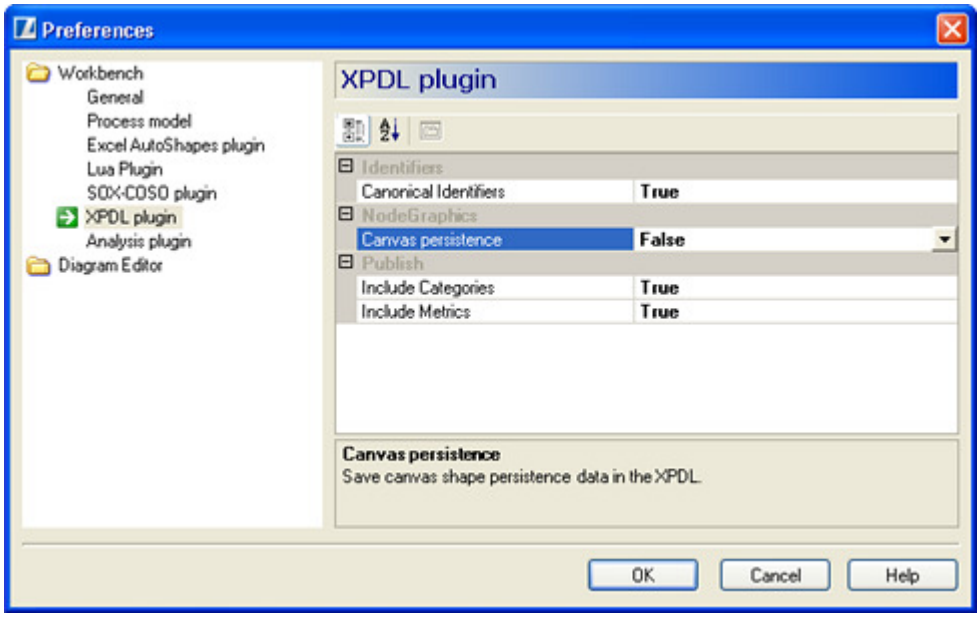
The measurepoint becomes the anchor location in the process model where the BPMN diagram will be created.

The plugin will automatically detect which version of XPDL is being used.

XPDL Plugin Preferences

The XPDL preferences let you customize the behaviour of the XPDL plugin.

Figure 3. The XPDL plugin preferences page



Canonical Identifiers (for round tripping)

Within a single process model file, object identifiers must be unique. Normally identifiers based on GUIDs (Globally Unique Indentifiers) are used, for example `dif6db6247fb0b47509d34389e8d187864` is a typical Diagram object identifier.

When Avantage reads from an XPDL package, it defers to the identifiers contained within the package instead of using automatically generated identifiers.

What would happen if you exported a BPMN diagram to XPDL and then re-imported it back to the same process model? You would see an error in the Log View window with a message directly or indirectly related to a duplicate ID. This is because as far as Avantage is concerned there is already a diagram present with that identifier.

BPMN dot notation to the rescue

BPMN objects can be identified by their **position** in the BPTree, for example **`My diagram.Main.Lane A.Task A`** identifies the BPMN Task object named "Task A" which occurs in "Lane A" which occurs within "Main" pool which occurs within "My Diagram".

If you set **Canonical Identifiers** to **TRUE** then the XPDL plugin does not write real BPObject indentifiers into the xpdl package and instead writes their dot notation location equivalents.

Figure 4. Dot notation identifiers

```
<xpdl:Transition Id="Pattern 4 - Activity with conditional
choice.Main.Start.Action 1" From="Pattern 4 - Activity with conditional
choice.Main.Start" To="Pattern 4 - Activity with conditional
choice.Main.Make choice" Name="Action 1" Quantity="1"/>
```

When the XPDL plugin reads a package with Canonical indentifiers, the XPDL plugin creates BPObjects with automatically assigned identifiers instead.

The default setting for Canonical Identifiers is **TRUE** which is suitable for round tripping between Avantage.

➔ **Note**: You will most likely get errors in the log if you attempt to import 3rd party XPDL files with **Canonical identifiers** TRUE.

Canvas Persistence (for round tripping)

When this setting is **TRUE**, the diagram editor's shape persistence information is emitted in the xpdl in the form of an extended attribute for the entity. This information is richer than **NodeGraphicsInfos** (which are also supported).

This setting is intended for Avantage round-tripping and is ignored by other XPDL software so there is no harm in enabling it. This setting preserves details like shape color gradients, color fill transparency, fill patterns, line widths, font names and sizes and more.

Include Categories

If the Process Analytics bundle is installed you can optionally enable the output of Category meta data if it is present in your process.

Figure 5. Categories meta data

```
<xpdl:ExtendedAttribute Name="KT.AM.Core.Categories.CategorySet"
Value="AM2.5 Categories">

      <Category Name="Paper handling, sorting" Id="0"
backgroundColor="Color [PaleGreen]" labelPrefix="V"/>

      <Category Name="Generate original document" Id="1"
backgroundColor="Color [PaleGreen]" labelPrefix="PV"/>
```

```
      <Category Name="Edit, process paper" Id="2"
backgroundColor="Color [LightGreen]" labelPrefix="PV"/>

      <Category Name="Approve on paper" Id="3" backgroundColor="Color
[LightGreen]" labelPrefix="PV"/>

      <Category Name="Carry paper" Id="4" backgroundColor="Color
[PaleGreen]" labelPrefix="P"/>

      <Category Name="Hardware operation" Id="5"
backgroundColor="Color [LemonChiffon]" labelPrefix="MV"/>

      <Category Name="Data entry" Id="6" backgroundColor="Color
[LemonChiffon]" labelPrefix="MV"/>

      <Category Name="Edit, software operation" Id="7"
backgroundColor="Color [LemonChiffon]" labelPrefix="MV"/>

      <Category Name="Approve on system" Id="8" backgroundColor="Color
[LemonChiffon]" labelPrefix="MV"/>

      <Category Name="Machine Operation" Id="9" backgroundColor="Color
[CornflowerBlue]" labelPrefix="MV"/>

      <Category Name="Engineering" Id="10" backgroundColor="Color
[PaleVioletRed]" labelPrefix="MV"/>

      <Category Name="Telephony" Id="11" backgroundColor="Color
[SandyBrown]" labelPrefix="HV"/>

      <Category Name="Oral communications" Id="12"
backgroundColor="Color [SandyBrown]" labelPrefix="HV"/>

      <Category Name="Transportation" Id="13" backgroundColor="Color
[SandyBrown]" labelPrefix="-"/>

      <Category Name="Mixed" Id="14" backgroundColor="Color
[SandyBrown]" labelPrefix="-"/>

</xpdl:ExtendedAttribute>
```

Include Metrics

If the Process Analytics bundle is installed you can optionally enable the output of Metrical analysis meta data if it is present in your process.

Figure 6. Metrics extended attributes

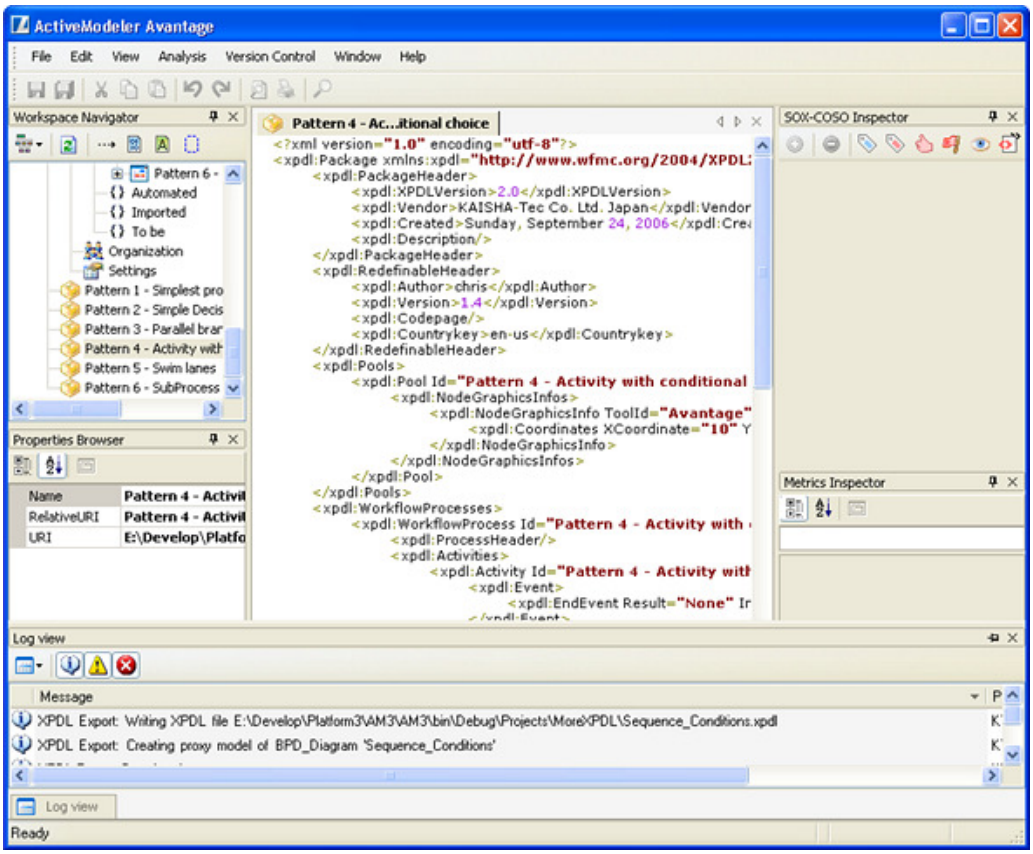
```
<xpdl:ExtendedAttributes>
    <xpdl:ExtendedAttribute Name="KT.AM.Metrics" Value="Metrics">
        <Metric Name="Time Based Unit Cost" Id="KT.AM.Metrics.Properties.Cost.UnitCost.TimeBased" DefaultValue="0"/>
        <Metric Name="Wait Time Unit Cost" Id="KT.AM.Metrics.Properties.Cost.UnitCost.WaitTime" DefaultValue="0"/>
        <Metric Name="Wait Time Total Cost" Id="KT.AM.Metrics.Properties.Cost.TotalCost.WaitTime" DefaultValue="0"/>
        <Metric Name="Time Based Total Cost" Id="KT.AM.Metrics.Properties.Cost.TotalCost.TimeBased" DefaultValue="0"/>
        <Metric Name="Unit Variable Cost(non-time)" Id="KT.AM.Metrics.Properties.UnitCost.Variable" DefaultValue="0"/>
        <Metric Name="Total Variable Cost(non-time)" Id="KT.AM.Metrics.Properties.TotalCost.Variable" DefaultValue="0"/>
        <Metric Name="Fixed Cost" Id="KT.AM.Metrics.Properties.Cost.FixedCost" DefaultValue="0"/>
        <Metric Name="Total Cost" Id="KT.AM.Metrics.Properties.Cost.TotalCost" DefaultValue="0"/>
        <Metric Name="External Volume" Id="KT.AM.Metrics.Properties.Volume.ExternVolume" DefaultValue="1000"/>
        <Metric Name="Internal Volume" Id="KT.AM.Metrics.Properties.Volume.InternVolume" DefaultValue="0"/>
        <Metric Name="Time Period" Id="KT.AM.Metrics.Properties.Volume.TimePeriod" DefaultValue="0"/>
        <Metric Name="Unit Time" Id="KT.AM.Metrics.Properties.Time.UnitTime" DefaultValue="5m"/>
        <Metric Name="Total Time" Id="KT.AM.Metrics.Properties.Time.TotalTime" DefaultValue="83h 20m"/>
        <Metric Name="Wait Time" Id="KT.AM.Metrics.Properties.Time.WaitTime" DefaultValue="0m"/>
        <Metric Name="Wait Per Unit" Id="KT.AM.Metrics.Properties.Time.WaitPerUnit" DefaultValue="False"/>
    </xpdl:ExtendedAttribute>
```

For help with using process metrics please refer to the Metrics plugin guide.

Use of the Log View

All operations of the XPDL plugin are written into the log. It is useful to **pin** the Log View to the bottom of your workspace so that it it pops up only when you want to look at the results of an operation. You can double-click on an entry to see more details about the entry.

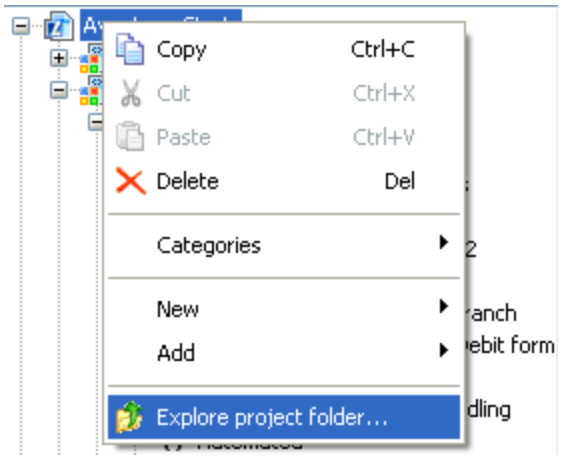
Figure 7. Log View pinned to bottom of workspace



To discover where the file has been written

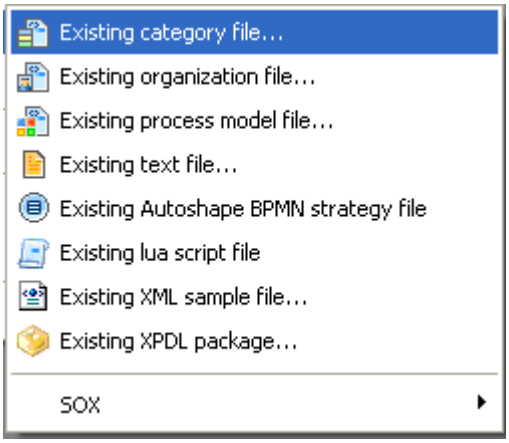
Choose the "**Explore project folder**" menu action. This will open an Explorer Window displaying the contents of your project folder.

Figure 8. Exploring the project folder



Adding XPDL package files to the Workspace

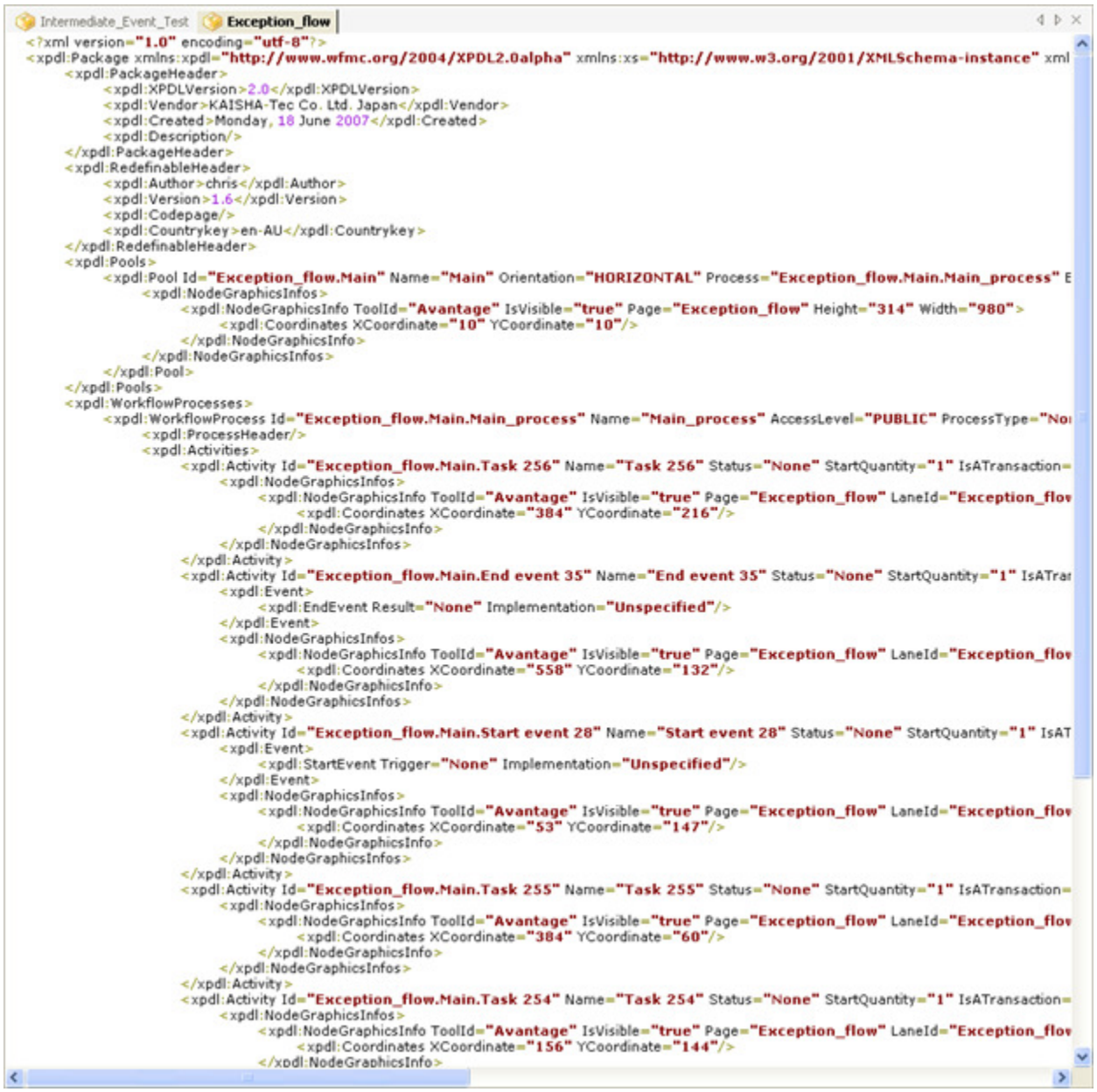
It is convenient to work with the xpdl file in the Avantage workspace. To add an xpdl package to the workspace, right click on the project and choose the "**Add existing XPDL package...**" action.



To view an XPDL package

You can view/edit an XPDL package by clicking on its icon in the navigator tree.

Figure 9. XPDL Package in editor window



Limitations

➔ XPDL 2.0 is not the penultimate process model interchange standard. It is basically a committee based hack of the XPDL 1.0 standard in order to try to make it fit better with BPMN.

➔ The plugin **does not** support XPDL 1.0 export.

➔ The plugin pays little regard to reading XPDL 1.0 packages comprehensively, because the focus is on XPDL 2.0. The plugin will construct a BPMN diagram consisting of the Pools, Lanes, Tasks and connections from the XPDL 1.0 document.

➔ The XPDL plugin writes NodeGraphicsInfo (graphical shape coordinate/location, information (see example below).

```
<xpdl:NodeGraphicsInfos>
      <xpdl:NodeGraphicsInfo ToolId="Avantage" IsVisible="true"
      Page="Intermediate_Event_Test"
LaneId="Intermediate_Event_Test.Main" Height="43" Width="43">
      <xpdl:Coordinates XCoordinate="253" YCoordinate="108"/>
      </xpdl:NodeGraphicsInfo>
</xpdl:NodeGraphicsInfos>
```

This allows other vendor's XPDL 2.0 software to read the shape position and size information from Avantage BPMN process diagrams.

➔ The XPDL plugin discards NodeGraficsInfo elements when importing. Instead, the plugin flags the newly created diagram as 'needing layout' and the diagram is automatically layed out before it is first displayed.

➔ Vendor XPDL 2.0 implementations which are not fully W3C compliant in the XML department can cause trouble with data bindings. For example, one vendor generates this kind of error when validated:

```
Element
'{http://www.wfmc.org/2004/XPDL2.0alpha}DataMapping'
cannot be empty according to the DTD/Schema.
```

Another vendor's XPDL 2.0 when validated has this kind of error:

```
Incorrect definition for the root element in schema.
```