

Directed Graph Markup Language (DGML) reference

11/04/2016 • 7 minutes to read •  +5

In this article

[DGML syntax](#)

[See also](#)

Directed Graph Markup Language (DGML) describes information used for visualization and to perform complexity analysis, and is the format used to persist code maps in Visual Studio. It uses simple XML to describe both cyclical and acyclic directed graphs. A directed graph is a set of nodes that are connected by links, or edges. Nodes and links can be used represent network structures, such as elements in a software project.

Note that some versions of Visual Studio support only a subset of DGML capabilities, see [Version support for architecture and modeling tools](#).

Note

When you edit a .dgml file, IntelliSense helps you identify attributes that are available for each element and their values. To specify color in an attribute, use names for common colors, such as "Blue", or ARGB hexadecimal values, such as "#ffa0b1c3". DGML uses a small subset of Windows Presentation Foundation (WPF) color definition formats. For more information, see [Colors Class](#).

DGML syntax

The following table describes kinds of elements that are used in DGML:

- `<DirectedGraph></DirectedGraph>`

This element is the root element of code map (.dgml) document. All other DGML elements appear within the scope of this element.

The following list describes optional attributes that you can include:

Background - The color of the map background

BackgroundImage - The location of an image file to use as the map background.

GraphDirection - When the map is set to tree layout (Sugiyama), arrange the nodes so that most of the links flow in the specified direction: `TopToBottom`, `BottomToTop`, `LeftToRight`, or `RightToLeft`. See [Change the map layout](#).

Layout - Set the map to the following layouts: `None`, `Sugiyama` (tree layout), `ForceDirected` (quick clusters), or `DependencyMatrix`. See [Change the map layout](#).

NeighborhoodDistance - When the map is set to tree layout or quick clusters layout, show only those nodes that are a specified number (1-7) of links away from selected nodes. See [Change the map layout](#).

Example:

XML	 Copy
<pre><?xml version="1.0" encoding="utf-8"?> <DirectedGraph Title="DrivingTest" Background="Blue" xmlns="http://schemas.microsoft.com/vs/2009/dgml"> <Nodes> ... </Nodes> <Links> ... </Links> <Categories> ... </Categories> <Properties> ... </Properties> </DirectedGraph></pre>	


- `<Nodes></Nodes>`

This optional element contains a list of `<Node/>` elements, which define nodes on the map. For more information, see the `<Node/>` element.

Note

When you reference an undefined node in a `<Link/>` element, the map creates a `<Node/>` element automatically.

Example:

XML	 Copy
<pre><?xml version="1.0" encoding="utf-8"?> <DirectedGraph Title="DrivingTest" xmlns="http://schemas.microsoft.com/vs/2009/dgml"> <Nodes> <Node ... /> </Nodes> <Links> <Link ... /> </Links> </DirectedGraph></pre>	

- `<Node/>`

This element defines a single node. It appears within the `<Nodes><Nodes/>` element list.

This element must include the following attributes:

Id - The unique name of the node and the default value of the **Label** attribute, if no separate **Label** attribute is specified. This name must match the **Source** or **Target** attribute of the link that references it.

The following list describes some of the optional attributes that you can include:

Label - The display name of the node.

Style attributes. See [Customize code maps by editing the DGML files](#).

Category - The name of a category that identifies elements that share this attribute. For more information, see the `<Category/>` element.

Property - The name of a property that identifies elements that have the same property value. For more information, see the `<Property/>` element.


Group - If the node contains other nodes, set this attribute to **Expanded** or **Collapsed** to show or hide its contents. There must be a `<Link/>` element that includes the

Category="Contains" attribute and specifies the parent node as the source node and the child node as the target node. See [Group code elements](#).

Visibility - Set this attribute to Visible, Hidden, or Collapsed. Uses System.Windows.Visibility. See [Hide or show nodes and links](#).

Reference - Set this attribute to link to a document or URL. See [Link documents or URLs to code elements and links](#).

Example:

XML	 Copy
<pre><?xml version="1.0" encoding="utf-8"?> <DirectedGraph Title="DrivingTest" xmlns="http://schemas.microsoft.com/vs/2009/dgml"> <Nodes> <Node Id="Driver" Label="Student" Category="Person" /> <Node Id="Passenger" Label="Instructor" Category="Person" /> <Node Id="Car" Label="Car" Category="Automobile" /> <Node Id="Truck" Label="Truck" Category="Automobile" /> </Nodes> <Links> <Link ... /> </Links> <Categories> <Category Id="Person" Background="Orange" /> <Category Id="Automobile" Background="Yellow"/> </Categories> </DirectedGraph></pre>	

- <Links></Links>

This element contains the list of <Link> elements, which define links between nodes. For more information, see the <Link/> element.

Example:

XML	 Copy
<pre><?xml version="1.0" encoding="utf-8"?> <DirectedGraph Title="DrivingTest" xmlns="http://schemas.microsoft.com/vs/2009/dgml"> <Links> <Link ... /> </Links></pre>	

```
</Links>  
</DirectedGraph>
```

- `<Link/>`

This element defines a single link that connects a source node to a target node. It appears within the `<Links></Links>` element list.

ⓘ Note

If this element references an undefined node, the map document automatically creates a node that has the specified attributes, if any.

This element must include the following attributes:

Source - The source node of the link

Target - The target node of the link

The following list describes some of the optional attributes that you can include:

Label - The display name of the link

Style attributes. See [Customize code maps by editing the DGML files](#).

Category - The name of a category that identifies elements that share this attribute. For more information, see the `<Category/>` element.

Property - The name of a property that identifies elements that have the same property value. For more information, see the `<Property/>` element.

Example:

XML

 Copy

```
<?xml version="1.0" encoding="utf-8"?>  
<DirectedGraph Title="DrivingTest"  
  xmlns="http://schemas.microsoft.com/vs/2009/dgml">  
  <Nodes>  
    <Node Id="Driver" Label="Student" Category="Person" />  
    <Node Id="Passenger" Label="Instructor" Category="Person" />  
    <Node Id="Car" Label="Car" Category="Automobile" />  
    <Node Id="Truck" Label="Truck" Category="Automobile" />  
  </Nodes>
```

```


<Links>
  <Category Id="Person" Background="Orange" />
  <Category Id="Automobile" Background="Yellow"/>
  <Link Source="Driver" Target="Car" Label="Passed" Stroke="Black"
Background="Green" Category="PassedTest" />
  <Link Source="Driver" Target="Truck" Label="Failed" Stroke="Black"
Background="Red" Category="PassedTest" />
</Links>
</DirectedGraph>

```

- `<Categories></Categories>`

This element contains the list of `<Category/>` elements. For more information, see the `<Category/>` element.

Example:

XML	 Copy
<pre> <?xml version="1.0" encoding="utf-8"?> <DirectedGraph Title="DrivingTest" xmlns="http://schemas.microsoft.com/vs/2009/dgml"> <Categories> <Category ... /> </Categories> </DirectedGraph> </pre>	

- `<Category/>`

This element defines a `Category` attribute, which is used to identify elements that share this attribute. A `Category` attribute can be used to organize map elements, provide for shared attributes through inheritance, or define additional metadata.

This element must include the following attributes:

Id - The unique name of the category and the default value of the `Label` attribute, if no separate `Label` attribute is specified.

The following list describes some of the optional attributes that you can include:


Label - A reader-friendly name for the category.

BasedOn - The parent category from which the `<Category/>` of the current element inherits.

In the example for this element, the `FailedTest` category inherits its `Stroke` attribute from the `PassedTest` category. See "To create hierarchical categories" in [Customize code maps by editing the DGML files](#).

Categories also provide some basic template behavior that controls the appearance of nodes and links when they are displayed on a map. See [Customize code maps by editing the DGML files](#).


Example:

XML	 Copy
<pre> <?xml version="1.0" encoding="utf-8"?> <DirectedGraph Title="DrivingTest" xmlns="http://schemas.microsoft.com/vs/2009/dgml"> <Nodes> <Node Id="Driver" Label="Driver" Category="Person" /> <Node Id="Car" Label="Car" Category="Automobile" /> <Node Id="Truck" Label="Truck" Category="Automobile" /> <Node Id="Passenger" Category="Person" /> </Nodes> <Links> <Link Source="Driver" Target="Car" Label="Passed" Category="PassedTest" /> <Link Source="Driver" Target="Truck" Label="Failed" Category="FailedTest" /> </Links> <Categories> <Category Id="Person" Background="Orange" /> <Category Id="Automobile" Background="Yellow"/> <Category Id="PassedTest" Label="Passed" Stroke="Black" Background="Green" /> <Category Id="FailedTest" Label="Failed" BasedOn="PassedTest" Background="Red" /> </Categories> </DirectedGraph> </pre>	

- `<Properties></Properties>`

This element contains the list of `<Property/>` elements. For more information, see the `<Property/>` element.

Example:

XML	 Copy

```
<?xml version="1.0" encoding="utf-8"?>
<DirectedGraph Title="DrivingTest"
xmlns="http://schemas.microsoft.com/vs/2009/dgml">
  <Properties>
    <Property ... />
  </Properties>
</DirectedGraph>
```

- <Property/>

This element defines a **Property** attribute that you can use to assign a value to any DGML element or attribute, including categories and other properties.

This element must include the following attributes:

- **Id** - The unique name of the property and the default value of the **Label** attribute, if no separate **Label** attribute is specified.
- **DataType** - The type of data stored by the property

If you want the property to appear in the **Properties** window, use the **Label** property to specify the property's display name.

See [Assign categories to code elements and links](#).

Example:

XML

 Copy

```
<?xml version="1.0" encoding="utf-8"?>
<DirectedGraph Title="DrivingTest"
xmlns="http://schemas.microsoft.com/vs/2009/dgml">
  <Nodes>
    <Node Id="Driver" Label="Driver" Category="Person" DrivingAge="18"/>
    <Node Id="Car" Label="Car" Category="Automobile" />
    <Node Id="Truck" Label="Truck" Category="Automobile" />
    <Node Id="Passenger" Category="Person" />
  </Nodes>
  <Links>
    <Link Source="Driver" Target="Car" Label="Passed"
Category="PassedTest" />
    <Link Source="Driver" Target="Truck" Label="Failed"
Category="FailedTest" />
  </Links>
  <Categories>
    <Category Id="Person" Background="Orange" />
  </Categories>
```



```
<Category Id="Automobile" Background="Yellow"/>
<Category Id="PassedTest" Label="Passed" Stroke="Black"
Background="Green" />
<Category Id="FailedTest" Label="Failed" BasedOn="PassedTest"
Background="Red" />
</Categories>
<Properties>
  <Property Id="DrivingAge" Label="Driving Age"
DataType="System.Int32" />
</Properties>
</DirectedGraph>
```

Aliases for commonly-used paths

Replacing commonly-used paths with aliases helps reduce the size of the .dgml file and the time required to load or save the file. To create an alias, add a `<Paths></Paths>` section at the end of the .dgml file. In this section, add a `<Path/>` element to define an alias for the path:

XML

 Copy

```
<Paths>
  <Path Id="MyPathAlias" Value="C:\...\..." />
</Paths>
```

To reference the alias from an element in the .dgml file, enclose the `Id` of the `<Path/>` element with a dollar sign (\$) and parentheses (()):

XML

 Copy

```
<Nodes>
  <Node Id="MyNode" Reference="$(MyPathAlias)MyDocument.txt" />
</Nodes>
<Properties>
  <Property Id="Reference" Label="My Document" DataType="System.String"
IsReference="True" />
</Properties>
```

See also

- [Map dependencies across your solutions](#)
- [Use code maps to debug your applications](#)

- [Find potential problems using code map analyzers](#)

Is this page helpful?

 Yes  No
