

# How to use Visualization Attributes for GaussVis

Johannes Ferber  
johannes.ferber@cern.ch

September 2, 2003

## General

Visualization settings in GaussVis can be applied to two different kinds of entities, that are logical volumes and materials. Since –for example– physical volumes aren't entities in this sense, settings cannot be applied to physical volumes.

## Properties

As mentioned above, properties can be defined for materials and logical volumes. Therefore the volumes (or materials, respectively) have to be mapped to a unambiguous name first which is then mapped to the desired settings. The first mapping takes place in `~/newmycmt/Det/XmlDDDB/(include version here)/DDDB/Visualization/LogVols.xml` (`./Materials.xml`, respectively, see section *Example* below).

The actual visualization properties are then mapped in `~/newmycmt/Det/XmlDDDB/(include version here)/DDDB/Visualization/VisAtts.xml`. The available properties are defined in the DTD (`~/newmycmt/Det/XmlDDDB/(include version here)/DDDB/DTD/colors.dtd`) as follows:

- **Color:** The color in RGB-notation (red, green and blue in the range 0..1). Mainly used for materials.
- **Visibility:** Defines whether a volume or material is visible or not. However, the visibility depends also on the properties of the mother volume (see section *Relations between Properties* below). Should be applied to logical volumes only.
- **opened status:** Defines whether child elements are visible. Should be applied to logical volumes only.
- **Visualization mode:** Defines whether the logical volumes is displayed as wireframe or with a plain surface. Should be applied to logical volumes only.

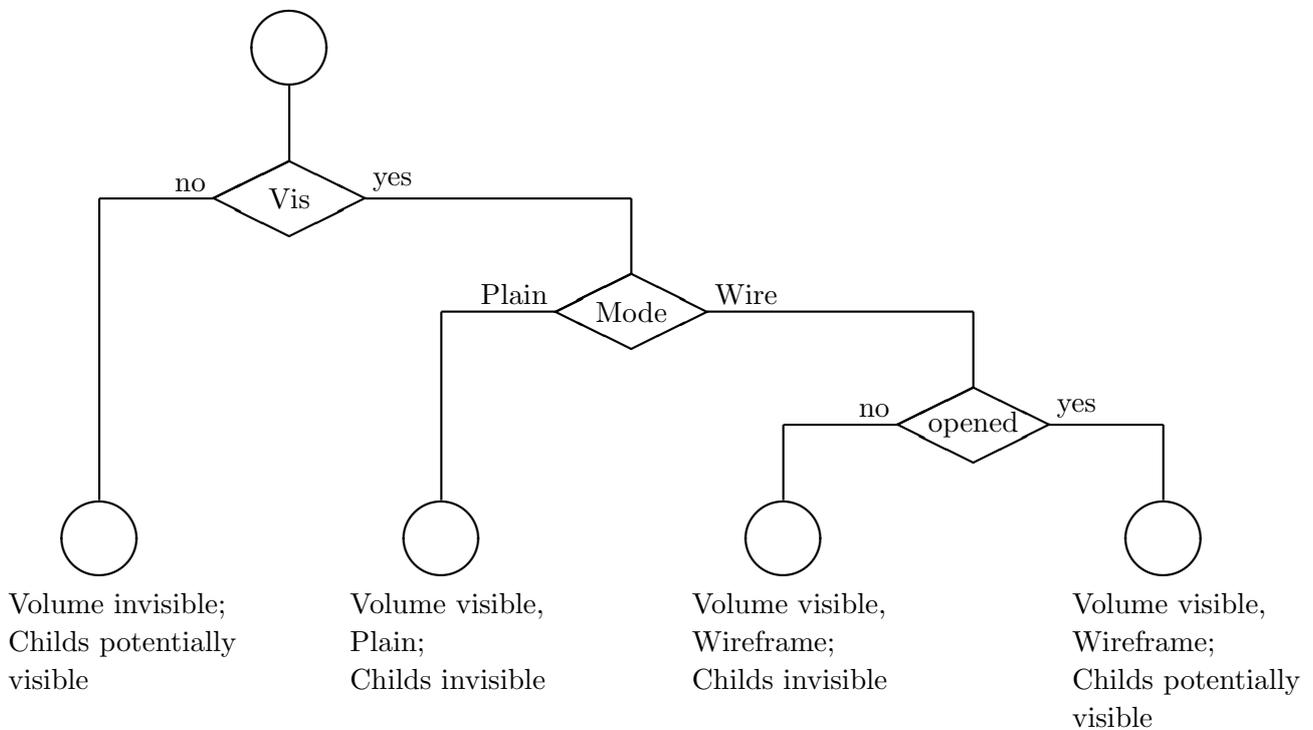
## Relations between Properties

The above enumerated visualization properties are not independent from each other. To achieve the desired representation certain precedence orders and dependencies have to be obeyed.

First, properties of mother volumes take priority over the child's properties. If –for example– the mother volume is visible, but closed (i.e. the **opened** status is **No**), the child volume won't be visible, no matter whether the child's visibility is set to **Yes** or **No**. If, on the other hand, the visibility of the mother volume is set to **No**, the child volume is potentially visible (depending on its own visibility attribute), no matter what the mother's **opened** status is (that's because the mother volume isn't visible and therefore unable to cover its child volumes. The **opened** status doesn't make sense for an invisible volume. Consider it to be opened). In addition, colors of logical volumes take precedence on the material colors.

Moreover also the visualization mode of a volume determines the visibility of child elements. A visible solid volume will always mask its child volumes while a wireframe allows to get visible access to child volumes as long as it is opened (a closed wireframe however covers its inner structure).

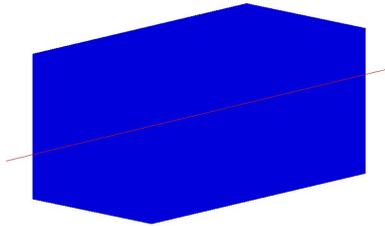
Last but not least one should keep in mind that there are a lot of logical volumes which just act as carriers for incorporated physical volumes and therefore consist of vacuum. In order to reveal the incorporated structure, these volumes should be set to invisible (or opened).



## Examples

The following two examples contain only the one corresponding line of the VisAtts.xml file. The third more extensive, recapitulating example contains all entries in the various files which are necessary for the visualization.

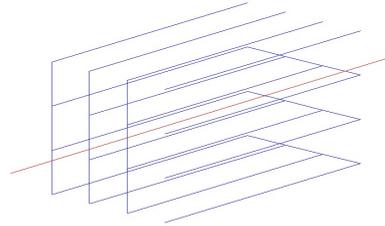
### Plain volume



```
<VisAtt  
name="EcalMiddleModule"  
mode="Plain" visible="Yes"  
opened="No"/>
```

**Note:** Since the volume is solid, an `opened="Yes"` would have had the same effect

### Wireframe volume



```
<VisAtt  
name="EcalMiddleModule"  
mode="WireFrame"  
visible="Yes" opened="No"/>
```

## Embedded Volumes

### Materials.xml:

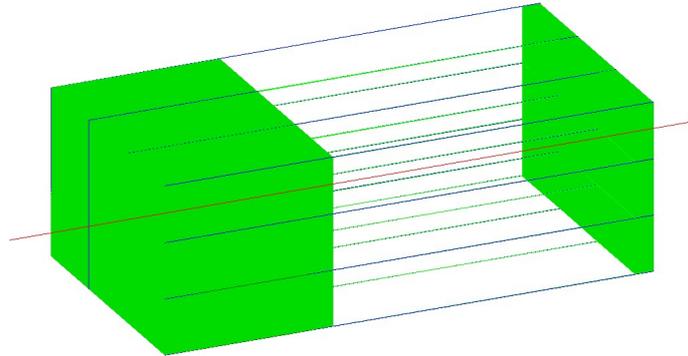
```
⋮  
<Item name="/dd/Materials/Ecal/EcalSteel" attr="EcalSteel"/>  
<Item name="/dd/Materials/Vacuum" attr="Vacuum"/>  
⋮
```

### LogVols.xml:

```
⋮
<Item name="/dd/Geometry/Ecal/Modules/Middle"
  attr="EcalMiddleModule"/>
  <Item name="/dd/Geometry/Ecal/Modules/MidFrontCover"
    attr="EcalMiddleFrontCover"/>
  <Item name="/dd/Geometry/Ecal/Modules/MidBackCover"
    attr="EcalMiddleBackCover"/>
  <Item name="/dd/Geometry/Ecal/Modules/MidReadOut"
    attr="EcalMiddleReadOut"/>
  <Item name="/dd/Geometry/Ecal/Modules/MidStack"
    attr="EcalMiddleStack"/>
    <Item name="/dd/Geometry/Ecal/Modules/MidCellSteel"
      attr="EcalMiddleCellSteel"/>
    <Item name="/dd/Geometry/Ecal/Modules/MidCellPlastic"
      attr="EcalMiddleCellPlastic"/>
    ⋮
  ⋮
⋮
```

### VisAtts.xml:

```
⋮
<VisAtt name="EcalSteel">
  <Color B="1" G="0" R="0"/>
</VisAtt>
<VisAtt name="Vacuum">
  <Color B="0" G="1" R="0"/>
</VisAtt>
⋮
<VisAtt name="EcalMiddleModule" mode="WireFrame" visible="Yes"
  opened="Yes"/>
  <VisAtt name="EcalMiddleFrontCover" mode="Plain"
    visible="Yes"/>
  <VisAtt name="EcalMiddleBackCover" mode="Plain"
    visible="Yes"/>
  <VisAtt name="EcalMiddleReadOut" mode="Plain"
    visible="Yes"/>
  <VisAtt name="EcalMiddleStack" mode="WireFrame"
    visible="Yes" opened="No"/>
    <VisAtt name="EcalMiddleCellSteel" mode="Plain"
      visible="Yes"/>
    <VisAtt name="EcalMiddleCellPlastic" mode="Plain"
      visible="Yes"/>
    ⋮
  ⋮
⋮
```



The example above first defines some unambiguous identifiers for the `EcalSteel` material, `Vacuum` and a couple of logical volumes. These identifiers are mapped in `VisAtts.xml` to their corresponding properties. While the effect of the `Color` attributes is obvious, the logical volumes are displayed as follows: the module itself is visible, which means that a wireframe with a color according to its material is displayed, in this case blue for steel. Since the module is opened and displayed as a wireframe, its child elements (`EcalMiddleFrontCover`, `EcalMiddleBackCover`, `EcalMiddleReadOut`, `EcalMiddleStack`) are potentially visible (depending on their own visibility setting). Therefore, the two covers and the `ReadOut` volume are visible and drawn as solids. Pay attention to the fact that vacuum, here green, doesn't differ from other materials in terms of visual representation.

Finally, the `MiddleStack` is displayed as a wireframe. Since it is just a carrier for some physical volumes like the `MiddleCellSteel`, it consists of vacuum as well and is therefore green. Its `opened` status is set to `No`, thus not revealing its inner structure. For this reason the incorporated volumes `EcalMiddleCellSteel` and `EcalMiddleCellPlastic` are –even though their own visibility flag is set– hidden.