

2

XML

The persistent world

Gaudi Framework Tutorial, 2001



This part of the tutorial deals with XML and its use in the description of the LHCb detector geometry.

It will be followed by a small exercise, aiming at defining the geometry of a new detector and visualizing it.

You can refer to the chapter “Detector description” of the Gaudi reference guide for further information.

Schedule:	Timing	Topic
	30 minutes	Lecture
	60 minutes	Practice
	90 minutes	Total

Objectives

After completing this lesson, you should be able to do the following:

- Read and write XML files
- Write LHCb detector geometry in XML
- Visualize the geometry

2-2

Gaudi Advanced Tutorial, 2001



Lesson Aims

- Read and write XML files : this includes having ideas on the syntax of XML and understanding how DTDs (Document type definitions) are used. The XmlEditor tool is also presented.
- Write LHCb detector geometry in XML : we'll present the DTD used for defining the LHCb detector geometry (including materials).
- Visualize the geometry : the GaudiLab software is presented. It allows to display a given XML geometry.

XML syntax overview

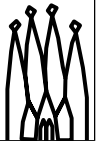
- A markup language
- Extendible
- Main components :
 - XML declaration
 - Comments
 - Elements
 - Attributes
 - Values

```
<Email>
  <TimeStamp [redacted] />
  <Sender> sender@cern.ch </Sender>
  <Recipient>
    recipient@cern.ch
  </Recipient>
  [redacted] Lunch... </Subject>
  <Body>
    [redacted]
    <Signature>
      Sender's signature
    </Signature>
  </Body>
</Email>
```



Usage of DTDs

- Definition of a grammar
- Specifies :
 - Allowed elements
 - Allowed attributes for a given element
 - Allowed children for a given element
- Written in markup language too



XML syntax overview

XML is a markup language, as HTML. The specificity of this class of languages is the usage of “tags” to structure the code. The very basic and mandatory rule of markup languages is that all tags must nest properly. The specificity of XML among other markup languages, as HTML, is that it is extendible. This means that a user can define his own tags.

The main components of XML are :

- The XML declaration : it must be at the beginning of every XML document. It says that this file is an XML file conforming to the XML standard version 1.0 and is encoded in UTF-8.
- Comments : They start with <!-- and end with -->. Comments in XML cannot be nested. This means you cannot open comments inside comments, as you can open a tag inside another tag.
- Elements : these are the tags. They generally go by two : the opening one of the form <name> and the closing one, either </name> or just </>. The only exception is for tags with no child. You can then write <name/>. In the example Email, TimeStamp or Signature are elements. The very basic and mandatory rule of XML is that all XML element tags must nest properly and that there must be only one root XML element at the top level of each XML document. An XML document that follows these rules is called well-formed.
- Attributes : These are pairs of the form name=value that are embedded inside the opening part of the tag. This is the case in the TimeStamp tag. Attributes usually describe the properties of the given element. The value of an attribute is a string and it has to be enclosed inside quotes. You can choose to use either simple or double quotes. If you use the former, you can then safely use the latter inside and vice-versa.
- Values : raw text can appear among nested XML elements. This is called value of the enclosing element. This is the case of the string “Could we meet at 14:00?”

Usage of DTDs

DTD means Document type definition. It defines a grammar for the XML file concerned by specifying mainly two things :

- The existing XML elements : their name and their possible children, possibly with an order and wildcards
- The attributes of each XML element : their name as well as their usage (required or not, default value, ...)

DTD are written in a markup language not far from XML. It is however not XML.

Note as an example that HTML is just XML plus the definition of a specific DTD. Some of the elements defined in this DTD are <HTML>, <BODY>, <A> or
. One attribute of the element <A> is href.

Example of DTD usage

```
<?xml version='1.0'
encoding='UTF-8'?>
<!-- This is an example of XML -->
<Sender>
  sender@cern.ch
</Sender>
<Recipient>
  recipient@cern.ch
</Recipient>
<Subject>
  Lunch...
</Subject>
<Body>
  Could we meet at 14:00?
  <Signature>
    Sender's signature
  </Signature>
</Body>
</Email>
```

```
<?xml version="1.0"
encoding="UTF-8"?>
<
  (
    ,
  )>
<!ELEMENT TimeStamp EMPTY>
<!ELEMENT Sender #PCDATA>
<!ELEMENT Recipient #PCDATA>
<!ELEMENT Subject #PCDATA>
<!ELEMENT Body
  (#PCDATA|Signature)*>
<!ELEMENT Signature #PCDATA>
```

2-5

Gaudi Advanced Tutorial, 2001



Example of DTD usage

Here is the DTD corresponding to the XML. Note that it is used as “external” dtd, which means that it is defined in a separated file, called file.dtd and referenced into the XML file in the second line. It could also be included inside the XML file but this is strongly discouraged since you want to be able to reuse a given DTD in several XML file as a consistency proof.

If an XML file is fulfilling the requirements of a given DTD, it is called valid. Otherwise, it is not valid, even if well-formed.

The LHCb Detector DTD

- Divided into 3 main parts
 - structure
 - geometry
 - material
- External DTDs, to be referenced in every LHCb XML files

2-6

Gaudi Advanced Tutorial, 2001



The LHCb DTDs

There are actually three main DTDs in LHCb, each of them allowing to describe a particular type of information :

- structure.dtd : allows to describe the structure of the detector. This is mainly a way of describing a tree of detector elements.
- geometry.dtd : this is where the actual geometry is described. Each detector element of the structure part references one of the geometries described here.
- material.dtd : this is the definition of the materials used in LHCb. They can then be referenced from the geometry.

Some specificities

- Expressions evaluator – units & functions known

```
12.2*mm + .17*m / tan (34*degree)
```

- parameter : a kind of macro

```
<parameter name="InCell" value="40.6667*mm"/>  
<parameter name="MidCell" value="1.5*InCell"/>
```

- References : element + “ref”

```
<detelemref href="LHCb/structure.xml#LHCb"/>
```

protocol://hostname/path/file.xml#ObjectID

2-7

Gaudi Advanced Tutorial, 2001



The LHCb DTD specificities

There are some specificities in the LHCb DTD, mainly three of them.

Every numerical value required by an attribute or an element in any of the DTDs of LHCb is an expression. This means that the value will be evaluated by the numerical expression parser. Thus, most of the current units and constants are already known. You can safely use degree, rad or pi for instance. On top of that, many mathematical functions are also known, such as sin, or exp but also arctan and many others.

A special element called parameter is defined in all DTDs from LHCb. This element allows the user to define his own parameters that can be then reused in any expression or value in the rest of the XML code. It has a name, a type, a value and a comment. The parameter element is actually a kind of macro since it will be replaced by its value everywhere it appears at parsing time. As for macros, the scope of a parameter is really uneasy to define. It is defined everywhere “after” its definition. The problem is that this “after” highly depends on the way you read the XML. The basic rule is that you should always define a parameter in a place that is above every node that will use it.

Some nodes have names finishing by “ref”. Each time there is a corresponding node without the ref. The “ref” nodes are actually kinds of hyperlinks on the “without ref” ones. The hyperlink is in general specified using the format : protocol://hostname/path/to/the/file.xml#ObjectID. The protocol and hostname parts can be omitted if the file resides on the local host. It is possible to write a hyperlink without the full path name in case one needs to refer to an XML object residing inside another file. In this case the relative path will be appended to the location of the currently parsed XML file. For example having the current file location /full/path/to/current.xml and inside this file a hyperlink as href="next/file.xml#NextOID" the hyperlink will be resolved as /full/path/to/next/file.xml#NextOID. If the hyperlink has the form #ObjectID this means that the referred object is located in the same file. Note that relative paths are strongly encouraged for every file except the top most one, since the whole sets of file may be copied in several different locations one day.

Structure Elements

- DDDDB : the root
- catalog : a list
- detelem : a detector element
- geometryInfo : connection to the geometry
- userParameter(Vector) : hook for adding parameters
- specific : hook for extending the DTD

```
<DDDB>  
<catalog name="...">  
<detelem name="...">  
<geometryinfo  
  lvname="..."  
  npath="..."  
  support="..." />  
<userParameter  
  comment="..."  
  name="..."  
  type="string">  
  ...  
</userParameter>  
<specific>  
  ...  
</specific>  
</detelem>  
</catalog>  
</DDDB>
```

2-8

Gaudi Advanced Tutorial, 2001



The LHCb DTD for structure

Here is a list of the main elements defined in the LHCb DTD for describing the structure of the detector :

- **DDDB** : this is to fulfill the XML basic rule that each XML document must have only one root XML element This is the root element.
- **catalog** : this is simply a list of elements, with a given name. This is a way to classify detector elements.
- **detelem** : detector elements are the essential part of the structure of the detector description. They fully describe a given part of the detector by holding data on the geometry of this part as well as on the subparts constituting it.
- **userParameter** and **userParameterVector** : this allows the user to add a parameter or a vector of parameters for a given detector element. This is intended to be used for specific parameters appearing in the subdetectors' descriptions. Their usage is described deeper in the next talk.
- **geometryInfo** : this element describes the geometry of a given detector element (logical volume, support and path from the support to the geometry).
- **specific** : this is the place where a user can extend the default detector description language and introduce new elements for his own needs. It's foreseen that the new XML elements be defined in a local DTD section of the XML data file or in a specific DTD file. Its usage is described deeper in the next talk.

Geometry Elements (1)

- **DDDB** : the root
- **catalog** : a list
- **logvol** : logical volume
- **physvol** : physical volume
- **paramphysvol(2D)(3D)** : replication of physical volumes
- **tabproperty** : tabulated properties

```
<DDDB>
<catalog name="...">
  <logvol material="..."
    name="...">
    <physvol logvol="..."
      name="..." />
  </logvol>
  <logvol name="...">
  <paramphysvol number="5">
  <physvol logvol="..."
    name="..." />
  <posXYZ z="20*cm" />
  </paramphysvol>
  </logvol>
</catalog>
</DDDB>
```

2-9

Gaudi Advanced Tutorial, 2001



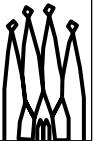
Geometry Elements(2)

- **posXYZ, posRPhiZ, posRThPhi** : translations
- **rotXYZ, rotAxis** : rotations
- **transformation** : composition of transformations
- **box, trd, trap, cons, tub, sphere**
- **union, intersection, subtraction** : boolean solids
- **surface**

```
<subtraction name="sub2">
  <box name="box3"
    sizeX="1*m"
    sizeY="1*m"
    sizeZ="15*cm" />
  <tubs name="tub2"
    outerRadius="15*cm"
    sizeZ="25*cm" />
</subtraction>
<posXYZ z="-40*cm" />
<rotXYZ rotX="90*degree" />
```

2-10

Gaudi Advanced Tutorial, 2001



The LHCb DTD for geometry

Here is a first list of the main elements defined in the LHCb DTD for describing the geometry of the detector :

- **DDDB** : this is to fulfill the XML basic rule that each XML document must have only one root XML element This is the root element.
- **catalog** : this is simply a list of elements, with a given name. This is a way to classify the logical volumes.
- **logvol** : this defines a logical volume. See lesson 1.
- **physvol** : this defines a logical volume. See lesson 1.
- **paramphysvol, paramphysvol2D, paramphysvol3D** : these are ways to define many physical volumes in one shot, by replicating a given volume and applying a given transformation between each replica.
- **tabproperty** : this defines a tabulated property. This is used to describe optical properties of materials and surfaces. See "*Optical properties & Surfaces*" available at <http://lhcb-comp.web.cern.ch/lhcb-comp/Frameworks/DetDesc/Documents/Optical.pdf>.

The LHCb DTD for geometry

Here is a second list of the main elements defined in the LHCb DTD for describing the geometry of the detector :

- **posXYZ, posRPhiZ, posRThPhi** : these are 3 ways of defining a translation : cartesian, cylindrical and spherical coordinate systems.
- **rotXYZ, rotAxis** : these are two ways of defining a rotation. Either along X, Y or Z axis, or along a user-defined axis.
- **transformation** : this defines a new transformation, by composition of several others
- **box, trd, trap, cons, tub, sphere** : these are all kinds of solids, namely boxes, simple trapezoids, general trapezoids, conic sections, tube sections and sphere. Please report to "*The LHCb Detector Description DTD*" available at <http://lhcb-comp.web.cern.ch/lhcb-comp/Frameworks/DetDesc/Documents/lhcbddt.pdf> for further details.
- **union, subtraction, intersection** : these are boolean operations on solids.
- **surface** : this defines a surface.

Material Elements

- **materials** : the root
- **catalog** : a list
- **tabproperty** : tabulated properties
- **atom**
- **isotope**
- **element** : a mixture of isotopes
- **material** : mixtures of elements or materials

```
<isotope A="11*g/mole"
  name="Bor_11" .../>
<element name="Boron"
  symbol="B" ...>
  <isotoperef href="#Bor_10"
    fractionmass="0.20"/>
  <isotoperef href="#Bor_11"
    fractionmass="0.80"/>
</element>
<element name="Oxygen"
  symbol="O" ...>
  <atom A="16*g/mole"
    Zeff="8.0000"/>
</element>
<material name="CO2" ...>
  <component name="Carbon"
    natoms="1"/>
  <component name="Oxygen"
    natoms="2"/>
</material>
```

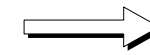
2-11

Gaudi Advanced Tutorial, 2001



XmlEditor

- Explorer-like XML viewer
- No need to know XML syntax
- Checks the DTD when opening a file
- Allows copy, paste and drag and drop of nodes
- Allows view of several files at the same time
- Hide references across files



Easy XML edition

`$LHCBSOFT/Det/XmlEditor/v*/scripts/xmlEditor(.bat)`

<http://lhcb-comp.web.cern.ch/lhcb-comp/Frameworks/DetDesc/Documents/XmlEditor.pdf>

2-12

Gaudi Advanced Tutorial, 2001



The LHCb DTD for material

Here is a list of the main elements defined in the LHCb DTD for describing the materials of the detector :

- **materials** : this is to fulfill the XML basic rule that each XML document must have only the one root XML element This is the root element.
- **catalog** : this is simply a list of elements, with a given name. This is a way to classify materials.
- **tabproperty** : this defines a tabulated property. This is used to describe optical properties of materials and surfaces. See "*Optical properties & Surfaces*" available at <http://lhcb-comp.web.cern.ch/lhcb-comp/Frameworks/DetDesc/Documents/Optical.pdf>.
- **atom** : this is used to define an element when it is not a mixture of isotopes.
- **isotope** : this is the definition of a given isotope of a given atom.
- **element** : this is a real life element, that is in general a mixture of several isotopes with given proportions.
- **component** : this is used to define mixtures. It associates a material and a proportion.
- **material** : this defines a mixture of several elements or even several other mixtures with given proportions.

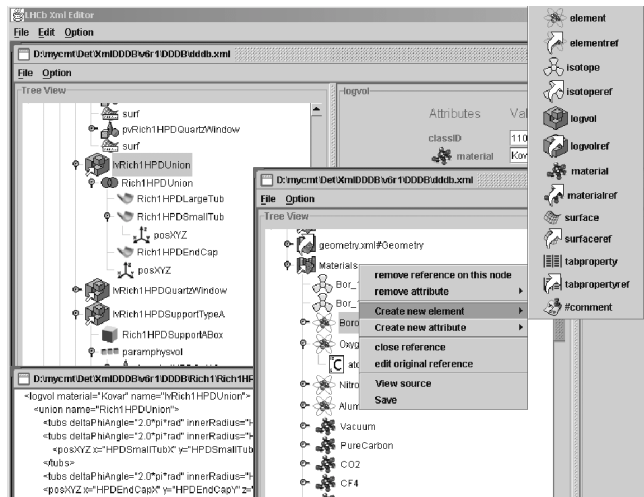
XmlEditor

The XML editor is a tool provided to edit the XML files of the detector description database without having to learn the XML syntax.

To start XmlEditor, just type `$LHCBSOFT/Det/XmlEditor/v*/scripts/xmlEditor` in a UNIX shell or `%LHCBSOFT%\Det\XmlEditor\v*\scripts\xmlEditor.bat` at the dos prompt. v* is the number of the version you want to use (v4r1 is the latest one at this time).

A documentation dedicated to the editor is available at <http://lhcb-comp.web.cern.ch/lhcb-comp/Frameworks/DetDesc/Documents/XmlEditor.pdf>.

XmlEditor screenshot



2-13

Gaudi Advanced Tutorial, 2001



XmlEditor screenshot

You can see on this picture the different elements of the XmlEditor GUI. A typical window is divided into two parts. On the left hand side, the tree of elements is displayed. Each element has an icon, the ones with a big arrow are references, the one with a small arrow are open references. You can right click on an element to get a contextual popup menu and perform some actions (create/remove elements/attributes, edit/close reference, view XML source...). On the right hand side are displayed the attributes of the selected node.

Several window can be open at the same time, even if they reuse the same file. Drag and drop or cut and paste across windows is allowed.

Panoramix

- Events and Geometry viewer
- Takes LHCb specificities into account
 - references
 - logical volumes hierarchy
 - subDetectors
- Interactive move inside the geometry

\$LHCBSOFT/Vis/Panoramix/v*/scripts/panoramix(.bat)
<http://www.lal.in2p3.fr/SI/Panoramix/tutorial/tutorial.html>

2-14

Gaudi Advanced Tutorial, 2001



Panoramix

Panoramix is the new package for visualization of both the detector geometry and the events. In our case, the event display is not available to avoid waiting too much at launch time (due to Sicb).

To start panoramix, just type :

\$LHCBSOFT/Vis/Panoramix/v*/scripts/panoramix where V* is the version number under Linux

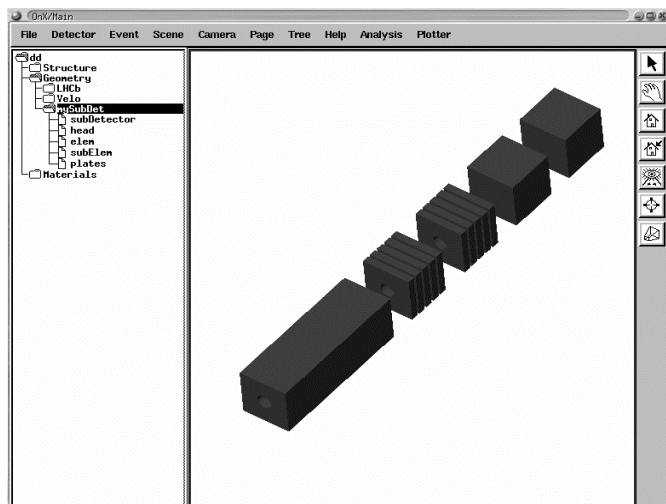
%LHCBSOFT%\Vis\Panoramix\v*\scripts\panoramix.bat where V* is the version number under Windows

or double click on the icon if you run a file manager.

Note that these scripts make take -noevent as argument. In this case, panoramix starts faster but you are not able to display events.

A documentation dedicated to panoramix is available at
<http://www.lal.in2p3.fr/SI/Panoramix/tutorial/tutorial.html>.

Panoramix screenshot



2-15

Gaudi Advanced Tutorial, 2001



Panoramix screenshot

You can see on this picture the different elements of the XmlEditor GUI. A tool bar on the right is missing and we won't use here the tree structure on the left. The center window displays the geometry, with different colors for different sub detectors.

References

- Gaudi user guide
- XmlEditor documentation
- “*The LHCb detector description DTD*”
- “*Extending detector elements and implications*”
- Panoramix tutorial

- Everything can be found at :
<http://lhcb-comp.web.cern.ch/lhcb-comp/>

2-16

Gaudi Advanced Tutorial, 2001



References

All references are available on the LHCb computing page. Follow the links to the pages dedicated to Gaudi, Detector Description and Event Displaying.

Exercise 1

2-17

Gaudi Advanced Tutorial, 2001

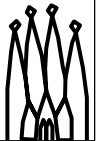


Exercise Goal

- Define the geometry of a new subDetector
- Use XmlEditor to write the XML
- Use Panoramix to visualize it

2-18

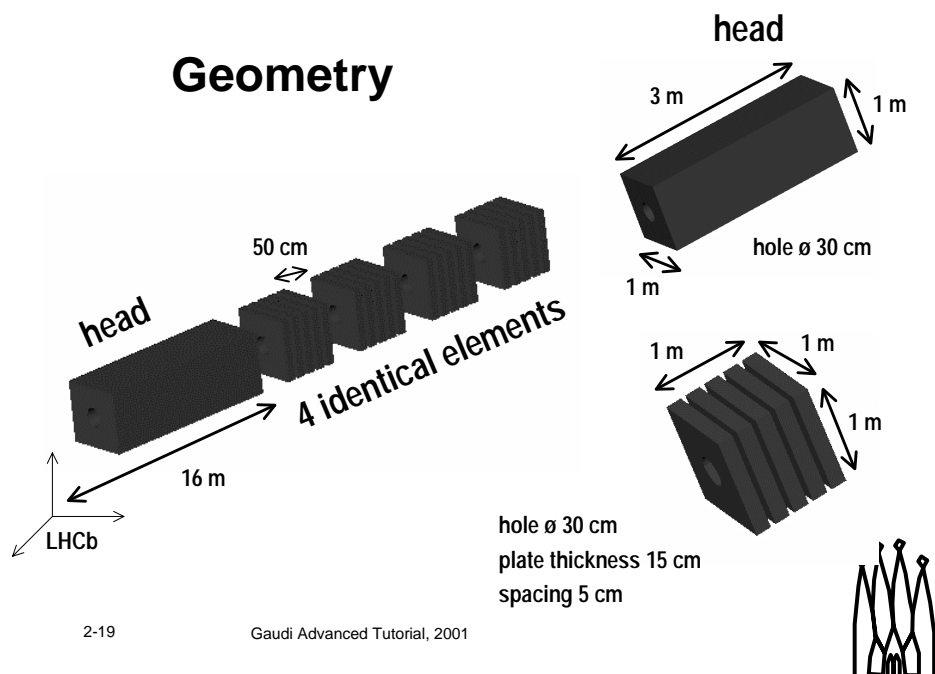
Gaudi Advanced Tutorial, 2001



Exercise goal

The goal of the first exercise is to define the geometry of a new sub detector, called mySubDet. We will use XmlEditor to write the xml code and panoramix to visualize the geometry.

Geometry



Geometry

This is the geometry we want to describe. It looks like it is complicated but it is actually very simple.

What is shown here is the goal of the exercise. It does not mean you have to describe that in the first attempt. What is suggested is to first describe the head and the elements as simple boxes. In a second step, the details inside one element could be added. At last, the tubular holes can also be added.

How to start

- `source $LHCBHOME/scripts/tutorial.csh`
- `cd tutorial`
- `getpack Tutorial/Detector v3`
- `cd Tutorial/Detector/v3`
- `cp -r data/DDDBStart data/DDDB`
- `cd cmt`
- `cmt config`
- `source setup.csh`

To launch the applications :

- `setenv DISPLAY ETSUN**:0.0`
- `xhost + lxplus (on the local host)`
- `xmlEditor`
- `panoramix`

How to start

These is the command lines to enter in order to start the tutorial. Here are some details

- You must first get the package Tutorial/Detector in his local area
- Then XML data are contained inside the data subdirectory. This one contains 4 versions of the data. DDDBStart is the one you should start with, the other are solutions of the exercises. The version you want to use for edition and visualization should reside in data/DDDB. So DDDBStart has to be copied there.
- At last, the setup.csh file must be sourced to setup the environment.

Once this is done, you can safely run xmlEditor and panoramix by typing their name on the command line.

What has to be done

- the directory `data/DDDB` contains part of the LHCb detector description (actually only Velo) and an empty `mySubDet`.
- To be done :
 - define the structure of the detector in `data/DDDB/mySubDet/structure.xml`
 - define the geometry of the detector in `data/DDDB/mySubDet/geometry.xml`
 - visualize the result and play with it

2-21

Gaudi Advanced Tutorial, 2001



Hints on the structure

- This is defined in `data/DDDB/mySubDet/structure.xml`
- You should define 6 detector elements :
 - One main
 - One for the head, and one per element
- Detector elements only contain references, no detector element. Thus, edit `structure.xml` directly to add new detector elements at the same level as the main detector element
- `GeometryInfo` nodes need a `lvname`, a `npath` and a support. Don't forget them
- Several detector elements may have the same logical volume, if they are physically identical

2-22

Gaudi Advanced Tutorial, 2001



What has to be done

The `data/DDDB` directory (provided you created it from `data/DDDB1`) contains part of the LHCb detector description, actually the Velo, plus an empty description of `mySubDet`. You should of course complete this description, ie the structure of the sub detector and its geometry. You should then be able to visualize it.

Note that nothing has to be done for materials. You can safely use the set of materials defined in the given geometry.

Don't hesitate to look at the Velo description to understand how everything works.

Hints on the structure

The structure of the new sub detector is defined in the file `data/DDDB/mySubDet/structure.xml`.

This one already contains the (empty) main detector element. You should complete it and define 5 other detector elements :

- one for the head element of the sub detector
- one for each of the four other elements

Note that the main detector element may not directly contain the sub elements but only references on them. The elements themselves must be added at the top level of the file `structure.xml`. Once the elements are created, you can create the references by a simple `Ctrl + Shift + drag and drop`.

Each detector element needs a `geometryInfo` element. This one describe the geometry of the element by giving the associated logical volume as well as the parent detector element and the path between the geometry of the parent and the one of the current element. Don't forget to give all these elements and take care of the case.

One last remark : a given logical volume may be used to define the geometry of several identical detector elements. In our case, the geometry of the 4 parts of the sub detector will be define only once.

Hints on geometry

- Do not forget to add your new sub detector to the LHCb geometry
- You should create 5 logical volumes only :
 - The main one
 - One for the head
 - One for the other elements
 - One for the inner of the element (you will need a coordinate system change)
 - One for the plates
- Remember the existence of parametrised physical volumes for defining several identical volumes
- Take care of the case !

2-23

Gaudi Advanced Tutorial, 2001



Hints on geometry

The geometry of the new sub detector is defined in the file `data/DDDB/mySubDet/geometry.xml`.

You have to define the geometry of every detector element defined in the structure. Actually, there are only three logical volumes here : the main one and the ones for head and elements of the detector. Then you will have to define two new ones to describe the inner geometry of one element : one for defining a plate and another to contain a parameterized physical volume of 5 plates, that will be placed inside the element.

This makes 5 logical volumes and many physical volumes in the `mySubDet/geometry.xml` file. Do not forget one more physical volume in file `LHCb/geometry.xml` to reference the main logical volume inside the LHCb global geometry.

Once more take care of the case.

Hints on XmlEditor

- File→open to open a file
- your files are in `mycmt/Tutorial/Detector/v3/data/DDDB`
- Double click to open nodes
- Right click to add elements/attributes
- Click on a value or node name to edit it
- Copy/paste available (Ctrl V→paste, Ctrl C→copy, Ctrl X→cut, Ctrl B→paste into parent)
- Drag & drop available (move, Ctrl→copy, Ctrl Shift→ref)
- Prefer File→save all to save you work

2-24

Gaudi Advanced Tutorial, 2001



Hints on XmlEditor

These are a subset of XmlEditor functionalities that should be sufficient for our needs :

- You open or close a file using the File menu. The most recent files are kept there
- Once you opened a file, every node but the top one is closed. Double click to let the children appear.
- Most of the modifications (add/remove element/attributes) can be reached by right clicking on the node concerned
- To edit an existing node or attribute, just click on it and wait a second
- Copy/Paste is available. The shortcuts are the usual Ctrl-C, Ctrl-v and Ctrl-X to copy, paste and cut plus Ctrl-B to paste into parent.
- Drag and drop is also available. The default action is to move the node. You can copy nodes by hitting Ctrl and create references on nodes by clicking on Ctrl and Shift
- To save you work, you have three possibilities. File->Save all saves every open file. File->Save save every file displayed in the current window. Right click->Save saves only the file in which the node resides. Take care that this does not save the other files displayed inside the window.

Hints on Panoramix

- Use Detector→mySubDet to display your geometry
- Then, two modes are available (you change by clicking on the icons on the right) :
 - “small hand”, you can :
 - rotate with left button + move
 - translate with middle + move or left + Ctrl + move
 - zoom with middle + Ctrl + move or left + middle + move
 - “pointer”, you can :
 - open a volume with Ctrl + left click
 - remove/add color with left click
- The “eye” icon automatically resizes the view to your screen
- Relaunch panoramix when changing geometry

2-25

Gaudi Advanced Tutorial, 2001



References

2-26

Gaudi Advanced Tutorial, 2001



Hints on Panoramix

After launching panoramix, you should have 2 windows displayed : the main window and the message window. The main one has a menu and a tool bar on the right. This is where the geometry will be displayed. The message one will give you useful error messages if your geometry is not correct.

In order to display some geometry, go to the Detector menu and choose the sub detector you want to display (mySubDet is a good choice !). The sub detector should appear (if not look at the error message on the other window and correct your geometry) and you will be able to play with it. There are mainly two modes : the viewing mode (or “small hand” mode) and the pointer mode.

In the viewing mode, you can change the way you see the geometry :

- rotate with left button + move
- translate with middle + move or left + Ctrl + move
- zoom with middle + Ctrl + move or left + middle + move

In the pointer mode, you can open logical volumes (closed by default) to see more details of your geometry :

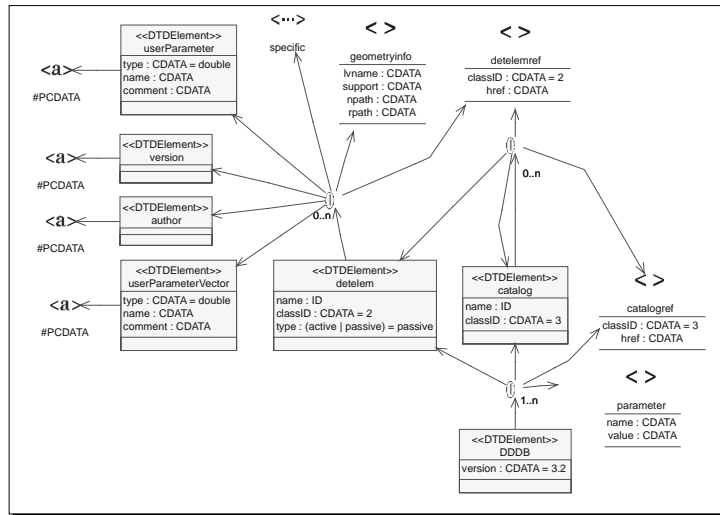
- open a volume with Ctrl + left click
- remove/add color with left click

Here is a small description of the other buttons of the tool bar :

- the small house with an arrow allows to mark the current view for latter retrieval
- the small house without an arrow retrieves the marked view
- the eye resizes and centers the objects displayed so that it fits well in the screen (without rotating them)

One last advice : panoramix is not able to reload dynamically the geometry. Thus, you should relaunch it every time you modify your geometry.

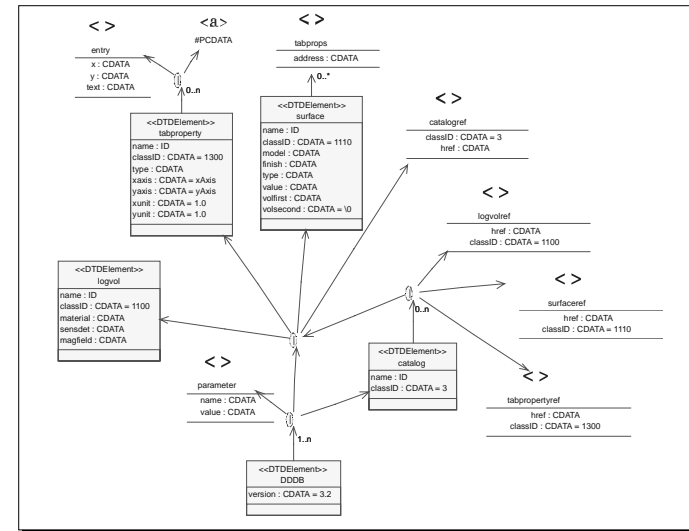
The LHCb Structure DTD



2-27 Gaudi Advanced Tutorial, 2001



The LHCb Geometry DTD (1)



2-28 Gaudi Advanced Tutorial, 2001



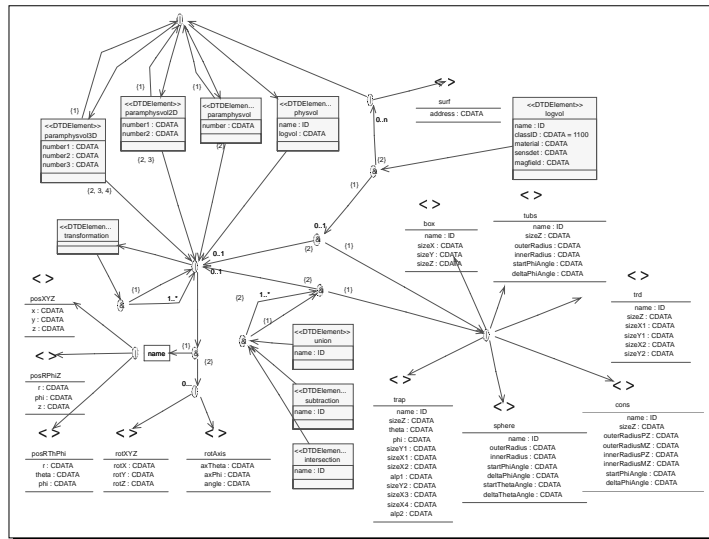
The LHCb Structure DTD

This is a global picture of the DTD for structure, using the UML notations. This is given as a reference.

The LHCb Geometry DTD – first part

This is a global picture of part of the DTD for geometry, using the UML notations. This is given as a reference.

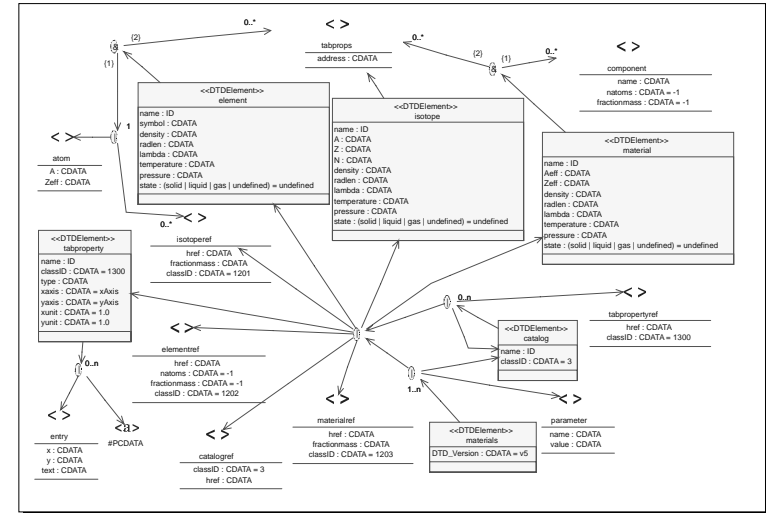
The LHCb Geometry DTD (2)



2-29 Gaudi Advanced Tutorial, 2001



The LHCb Materials DTD



2-30 Gaudi Advanced Tutorial, 2001



The LHCb Geometry DTD – second part

This is a global picture of part of the DTD for geometry, using the UML notations. This is given as a reference.

The LHCb Materials DTD

This is a global picture of the DTD for materials, using the UML notations. This is given as a reference.