

# Geant4 Documentation

1. Current status and types of “documentation”
2. Migration from docbook → ReST/Sphinx
3. Switch from svn → git
4. Style files, web-pages and more...

Alex Howard  
Documentation Co-ordinator

# Geant4 Documentation

1. Current status and types of “documentation”
2. Migration from docbook → ReST/Sphinx
3. Switch from svn → git
4. Style files, web-pages and more...

Alex Howard  
Documentation Co-ordinator

**Special thanks to Daren Sawkey and Ben Morgan for their assistance in the migration**

# Current Status

- “Documentation” covers a number of areas and formats:
  - Guides: application, developers, reference, getting started
  - Formats - output: pdf, html, (both)
  - Formats – input: docbook (guides), direct html (working groups), command-line (html create), oxygen, lxr browser
  - Hypernews is also a “go-to” place for many answers to specific problems, often replacing more complete documentation?
- Restrictions:
  - docbook is not the easiest to format (different html to pdf outputs sometimes),
  - versioning – everything is in the head so difficult to have future developments documented whilst updating current docs
- Some things are only web-based (commands, oxygen), whilst others are only pdf (physics reference manual)
- How to keep things up-to-date with developments?

# A reminder...

- As Marc emphasised last year – I am the documentation co-ordinator, not the master  
→ documentation is the responsibility of **ALL** of us
- Docbook and the release mechanism was cumbersome → we discussed at last year's CM and in the meantime a number of explicit replacement strategies
- We have a plethora of document filetypes: docbook (xml), latex (PRM), raw-html, plain text (e-mails and READMEs), word-docs
- Building the full documentation required a two page recipe and complex GNUmakefiles. Style files and other customisations could be painful. Converting between docs was not easy (e.g. PRM (latex) → ToolkitDevelopersGuide (docbook/xml))
- People would often send me bits of the documentation to be included and wouldn't know what their additions “looked” like and in a variety of formats!
- Perhaps also there was a resistance to change/update things
- Can this be improved? (back to my first two points...)

# Your Agreement

- Documentation WG decided on ReST/Sphinx as a long-term way forward
- We have prototypes for the migration of the InstallationGuide, ApplicationDevelopersGuide, and the PhysicsReferenceManual. Please see attached docs.
- Do we all agree on this?
- In addition we took the opportunity to put the migrated files in git → we would like to block svn access when the guides are migrated (or during the final stages)
- Please e-mail me if this causes problems or if you're planning on adding documentation in the coming weeks
- I'll make an announcement once migration has taken place, thereafter all commits will be exclusively through git and in ReST

# ReST/Sphinx

- It's plaintext input – format is readable but also a mark-up language
- Restructured Text/ReST/RST (not good names for googling!) is a mark-up language which allows formatted text to be written in plain text
- **Big** advantage: the input is human readable like a plain text file, but the formatting gives it the full power of docbook and latex
- The style is logical (like a wiki) whilst powerful (like latex) and multi-formatted (like docbook)
- Editors: on linux there is retext, many others (can also be any text editor)
- Sphinx: Tool that takes ReST input and generates output documentation in a variety of formats, e.g. HTML, PDF, ePub.

# Advantages of Sphinx (Ben)

- 1) Written in Python, simple “pip install Sphinx” on all platforms.
- 2) Standard documentation tool for Python, widespread adoption by many projects, so should have longevity and developments
- 3) Simple sphinx-managed Make based build system.
- 4) Many add on components, easy to create new ones using Python.
- 5) Easy integration with version control and CI systems to build, test and deploy generated documentation automatically.
- 6) Some limited integration with Doxygen/C++ via the Breathe bridge.

# Migration Strategy

- Concept: migrate from docbook to sphinx
- Pandoc used as an intermediate tool (internally – Daren is the expert)
- Sphinx offers ReST (restructured text) as input and can output in any useful format (currently html, pdf, latex are relevant for us).
- Allows the possibility also to review and correct the documents (“fresh eyes”) and hopefully not introduce new features.
- Opportunity for all of us to (proof)-read, expand and update the documents
- Agreement: ReST/Sphinx as the new format? svn deprecated and move to git
- For each guide we will block svn commits at the time of the migration and thereafter commits will be through git and format will be ReST - so not latex or xml (docbook)
- We're already making significant progress → see attached examples
- Maybe the (long) flights home are an opportunity to look through a few (of the ~1200) pages in a documentation guide?



# git

- All the new documentation is in a git repository:  
<https://gitlab.cern.ch/geant4/geant4-documentation>
- It's separate from the geant4 source code
- The README is the place to look for problems (e.g. sphinx version, pandoc python scripts etc...)
- Currently restricted to documentation working group members, will become full collaboration after migration
- As always remember that git is a 3-stage commit process: add, commit, push
- As in previous years, if it's helpful if you can e-mail if documentation is forthcoming, arrived, delayed etc....
- Hopefully things are cleaner/simpler/more straightforward now
- The master branch will be the most up-to-date
- We will create release-synchronised branches

# git workflow

- (1) create a branch on which to make the changes
- (2) edit, commit, rinse, repeat
- (3) Submit the branch as a Merge Request to the ***geant4-documentation***
- (4) Changes can be automatically tested/reviewed
- (5) when all good, merged to master

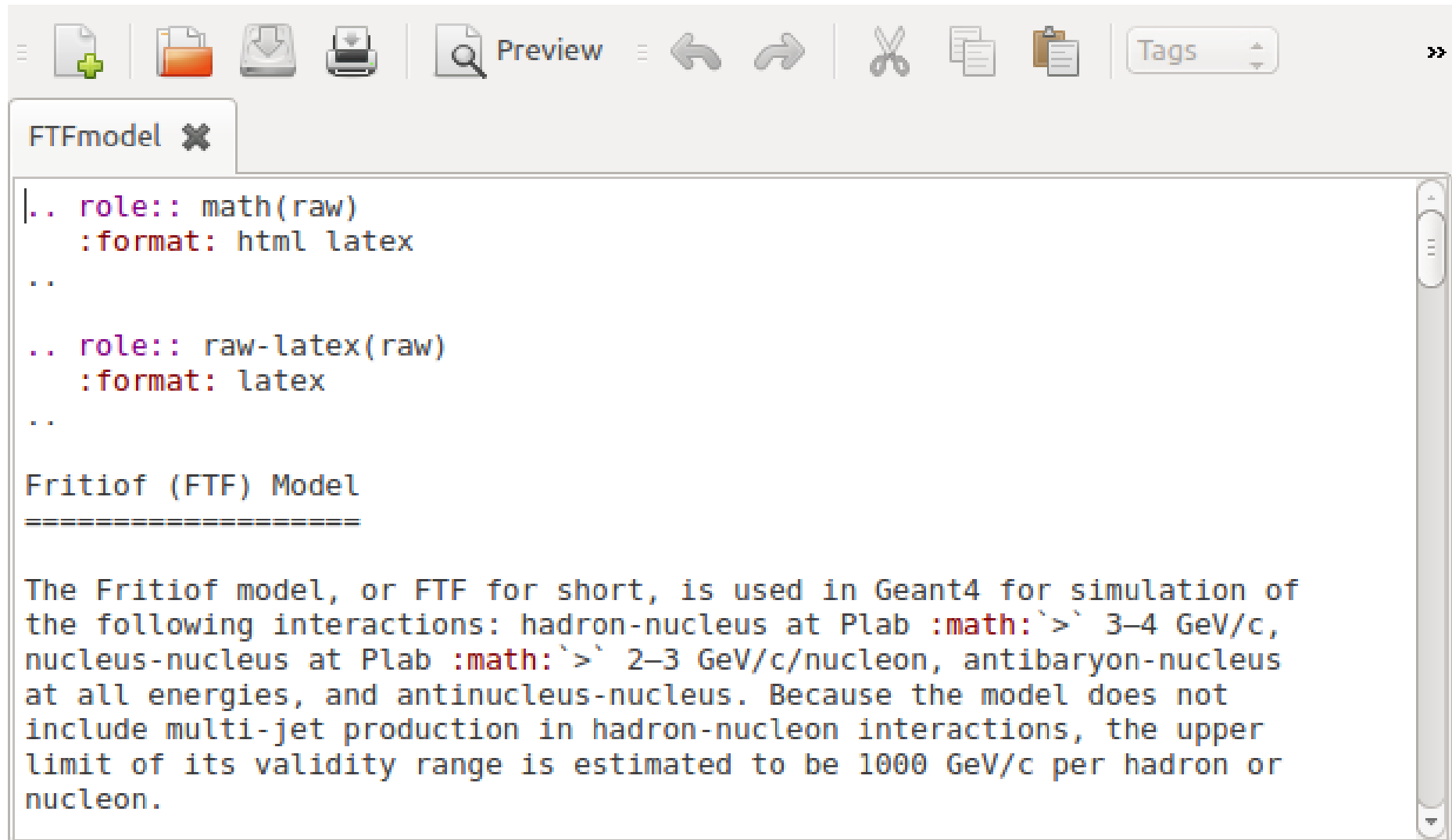
Same process as with the code:–

**see Gabriele/Ben's talk in plenary 2!!**

# Retext (linux)

- The editor is a normal text based one
- Has a “preview” button
  - Converts ReStructuredText into its graphical representation
- Very simple to see what changes you've implemented

# ReST/retext examples



```
FTFmodel ✕  
|.. role:: math(raw)  
  :format: html latex  
..  
.. role:: raw-latex(raw)  
  :format: latex  
..  
  
Fritiof (FTF) Model  
=====
```

The Fritiof model, or FTF for short, is used in Geant4 for simulation of the following interactions: hadron-nucleus at  $P_{lab} > 3-4$  GeV/c, nucleus-nucleus at  $P_{lab} > 2-3$  GeV/c/nucleon, antibaryon-nucleus at all energies, and antinucleus-nucleus. Because the model does not include multi-jet production in hadron-nucleon interactions, the upper limit of its validity range is estimated to be 1000 GeV/c per hadron or nucleon.

# ReST (figures & equations)

lines. If a point is below the string mass threshold, it is shifted to the nearest diffraction line.

```
.. figure:: hadronic/theory_driven/FTFmodel/Diag.eps
   :alt: *Diagram of the final states of hadron-hadron interactions.*\
   :width: 13.00000cm
   :height: 9.00000cm

   *Diagram of the final states of hadron-hadron interactions.*\
```

Unlike the original Fritiof model, the final state diagram of the current model is complicated, which leads to a mass sampling algorithm that is not simple. This will be considered below. The original model had no points corresponding to elastic scattering or to the binary final states. As it was known at the time, the mass of an object produced by

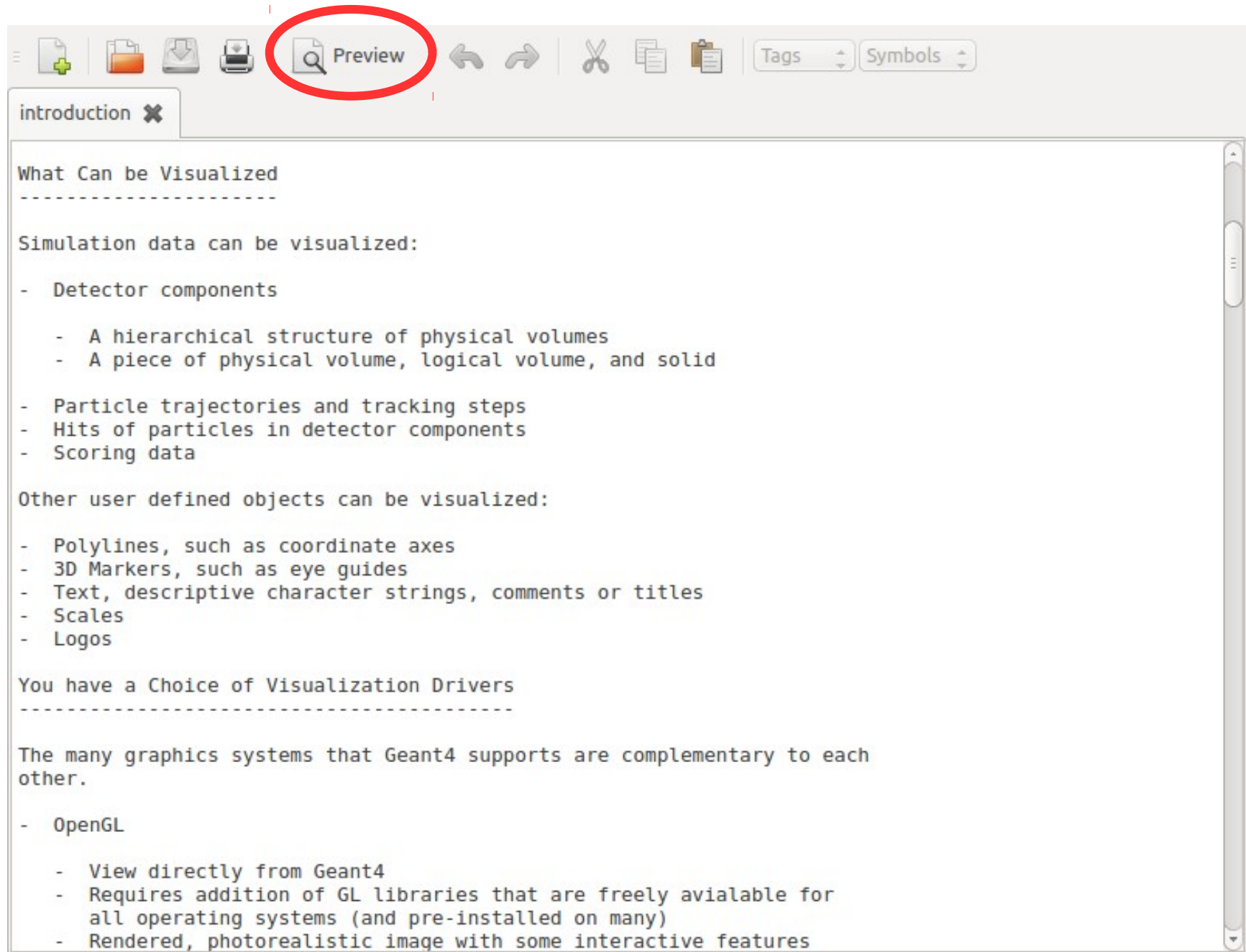
diffraction dissociation,  $M_x$ , for example from the reaction  $p+p \rightarrow p+X$ , is distributed as  $dM_x/M_x \propto dM_x^2/M_x^2$ , so it was natural to assume that the object mass distributions in all inelastic interactions obeyed the same law. This can be re-written using the light-cone momentum variables,  $P^+$  or  $P^-$ ,

```
.. math::

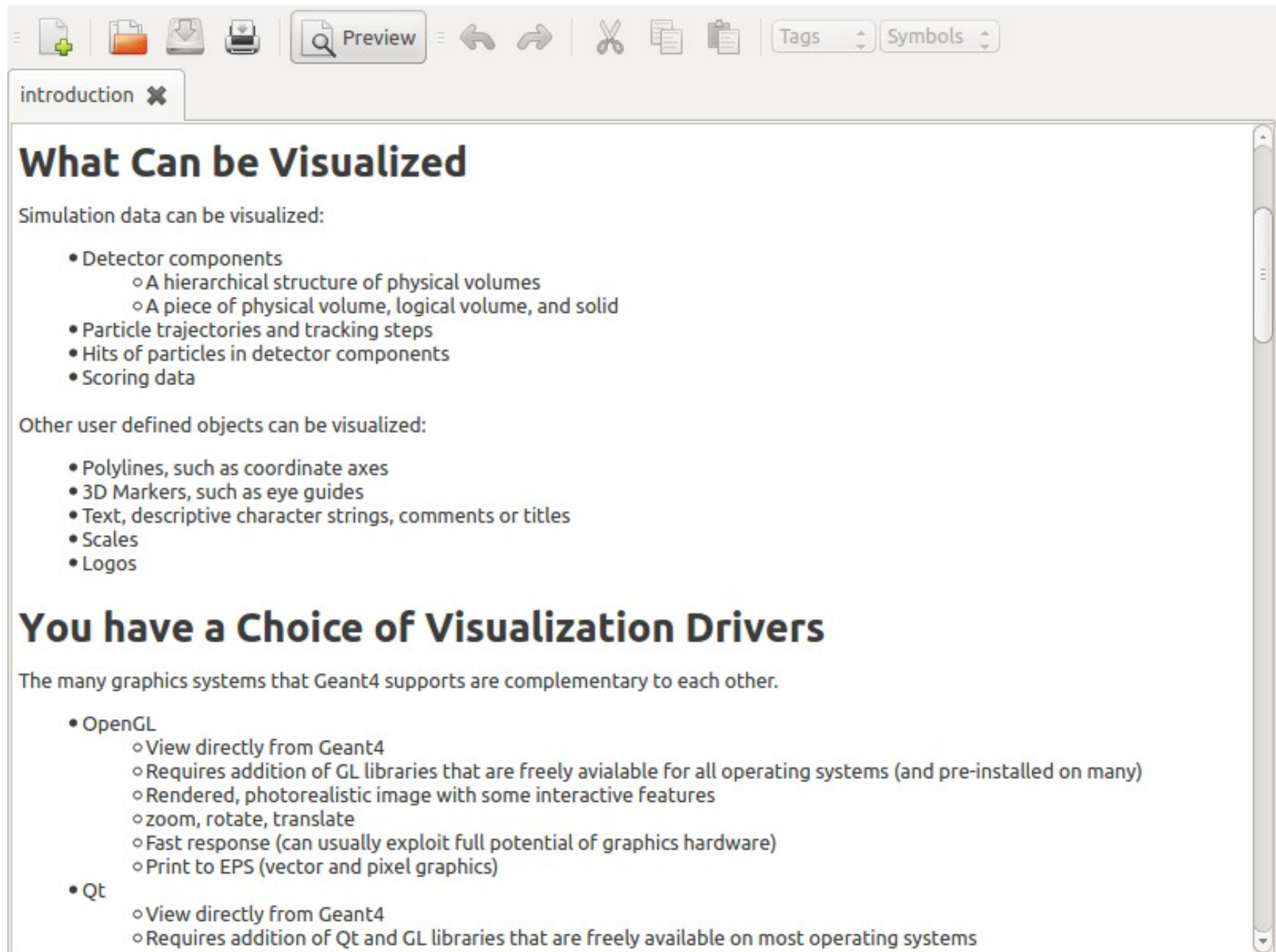
   P^+=E+p_z , \ \ \ P^- = E-p_z
   \label{eq:definition_light_cone}
```

where  $E$  is an energy of a particle, and  $p_z$  is its longitudinal momentum along the collision axis. At large energy and positive  $p_z$ ,  $P^+ \approx (M^2 + p_T^2) / (2p_z)$ . At negative

# Retext - preview



# Retext - preview



The screenshot shows a software window titled 'Retext - preview'. The window has a menu bar with icons for file operations (new, open, save, print) and a 'Preview' button. Below the menu bar is a tab labeled 'introduction'. The main content area displays the following text:

## What Can be Visualized

Simulation data can be visualized:

- Detector components
  - A hierarchical structure of physical volumes
  - A piece of physical volume, logical volume, and solid
- Particle trajectories and tracking steps
- Hits of particles in detector components
- Scoring data

Other user defined objects can be visualized:

- Polylines, such as coordinate axes
- 3D Markers, such as eye guides
- Text, descriptive character strings, comments or titles
- Scales
- Logos

## You have a Choice of Visualization Drivers

The many graphics systems that Geant4 supports are complementary to each other.

- OpenGL
  - View directly from Geant4
  - Requires addition of GL libraries that are freely available for all operating systems (and pre-installed on many)
  - Rendered, photorealistic image with some interactive features
  - zoom, rotate, translate
  - Fast response (can usually exploit full potential of graphics hardware)
  - Print to EPS (vector and pixel graphics)
- Qt
  - View directly from Geant4
  - Requires addition of Qt and GL libraries that are freely available on most operating systems

# Common Style?

- As I mentioned last year I would like to have a common style (“brand”) for the documentation linked in to the web-pages
- Currently we have quite a mixture, particularly with the WGs and also between documentation
- Sphinx offers the opportunity to simply change styles easily
- ReST could also be used for non-doc web-pages but outside of the core/drupal area
- Do we have a common style? Who should decide?



# Summary

- We've made significant progress on migrating away from docbook/latex and towards ReST/Sphinx
- I'll produce a how-to page on the new process – it's **much** simpler than before 😊
- Hopefully we'll all feel more inclined/inspired/motivated to produce and update our documentation
- Decision: git/ReST/Sphinx exclusively for the core documentation – **any objections?**
- Can we also have a common style for web-pages? Links into our “brand” - G4Logo, Videos, public perception etc...
- **Discussion?**

