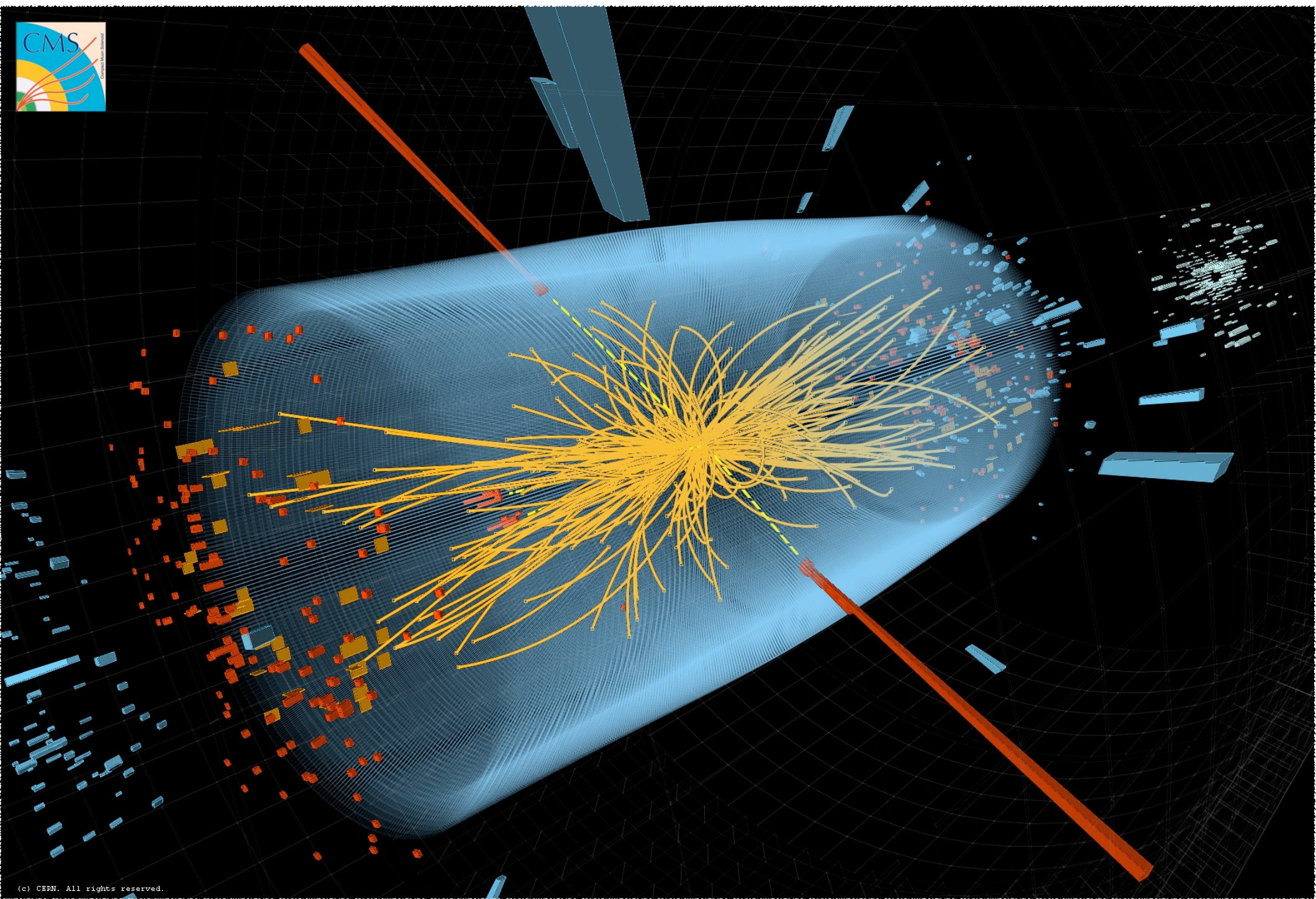


iSpy: a powerful and lightweight event display

G Alverson¹, G Eulisse², T McCauley², L Taylor²

¹ Northeastern University, ² Fermi National Accelerator Laboratory

ABSTRACT: iSpy is a general-purpose event data and detector visualization program that was developed as an event display for the CMS experiment at the LHC and has seen use by the general public and teachers and students in the context of education and outreach. Central to the iSpy design philosophy is ease of installation, use, and extensibility. The application itself uses the open-access packages Qt4 and Open Inventor and is distributed either as a fully-bound executable or a standard installer package: one can simply download and double-click to begin. Mac OSX, Linux, and Windows are supported. iSpy renders the standard 2D, 3D, and tabular views, and the architecture allows for a generic approach to production of new views and projections. iSpy reads and displays data in the ig format: event information is written in compressed JSON format files designed for distribution over a network. This format is easily extensible and makes the iSpy client indifferent to the original input data source. The ig format is the one used for release of approved CMS data to the public.



Data format

Extensible and human-readable

iSpy uses the *ig* format: JSON files in a zip archive. The event format is a valid python or JSON dictionary with three keys: *Types*, *Collection*, *Associations*.

```
"Types": {"Event_V2": [{"run", "int"}, {"event", "int"}, {"ls", "int"}, {"orbit", "int"}, {"bx", "int"}, {"time", "string"}, {"localtime", "string"}], "Tracks_V2": [{"pos", "v3d"}, {"dir", "v3d"}, {"pt", "double"}, {"phi", "double"}, {"eta", "double"}, {"charge", "int"}, {"chi2", "double"}, {"ndof", "double"}]}
```

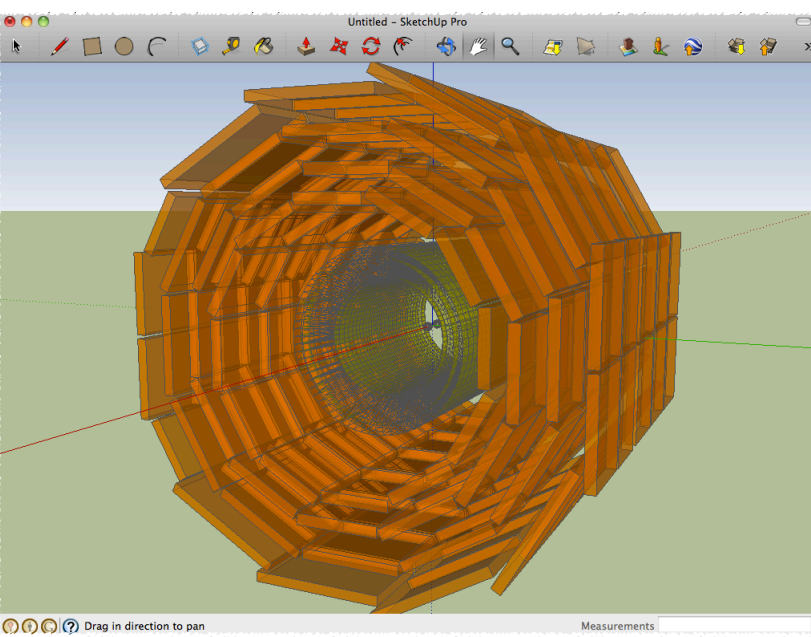
Collections contains the information specified in types. *Associations* associates two specific instances of different collections. An association element is a pair of two numbers: one specifying the collection index and the other the index of the particular object in the collection.

- Format is self-documenting
- ig files are easily parsed and written using C++, python, JavaScript, and ruby (for use in SketchUp for example). APIs are provided in github.
- ig files are usually created in CMSSW jobs, converting CMSSW objects into ig format: users require no special knowledge of CMS software and are insulated from differences in software versions and event formats
- Flexibility and extensibility make the format in-principle experiment independent

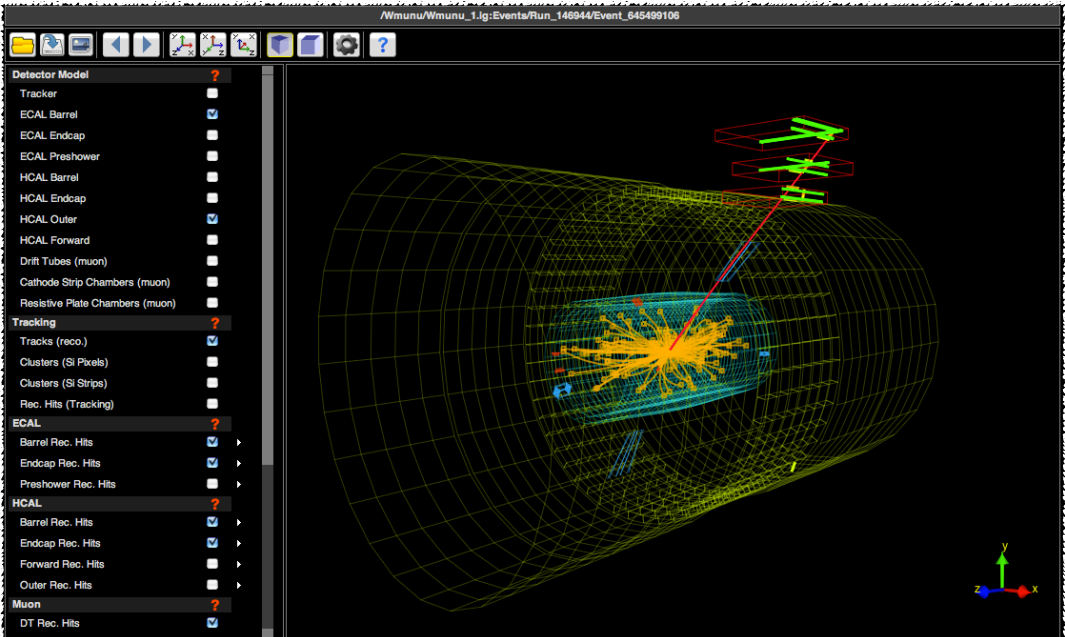
Use-cases

Open format can be used in other applications

- ig format used for public release of CMS data
- ig format is current implementation of “Level 2” data preservation initiative
- Application and data in ig format used in QuarkNet and IPPOG masterclasses
- Application used for creation of images for public and press
- Fork of application developed for DCS (Detector Control System) for CMS



ig file read into SketchUp via ruby plugin



ig files displayed in JavaScript web-based display based on iSpy

Configuration

CSS and XML configuration of styles and views

Configuration of graphical properties such as colors and physical properties such as the minimum energy of calorimetry objects is controlled from a iss file, the syntax of which mimics CSS syntax *e.g.*

```
Event_V2 {
    diffuse-color: rgb(1.0, 1.0, 1.0);
    font-size: 20.0;
    font-family: Arial;
    background: :/images/cms-logo-small.png;
    left: 0.125;
    top: 0.025;
    annotation-level: press;
}
```



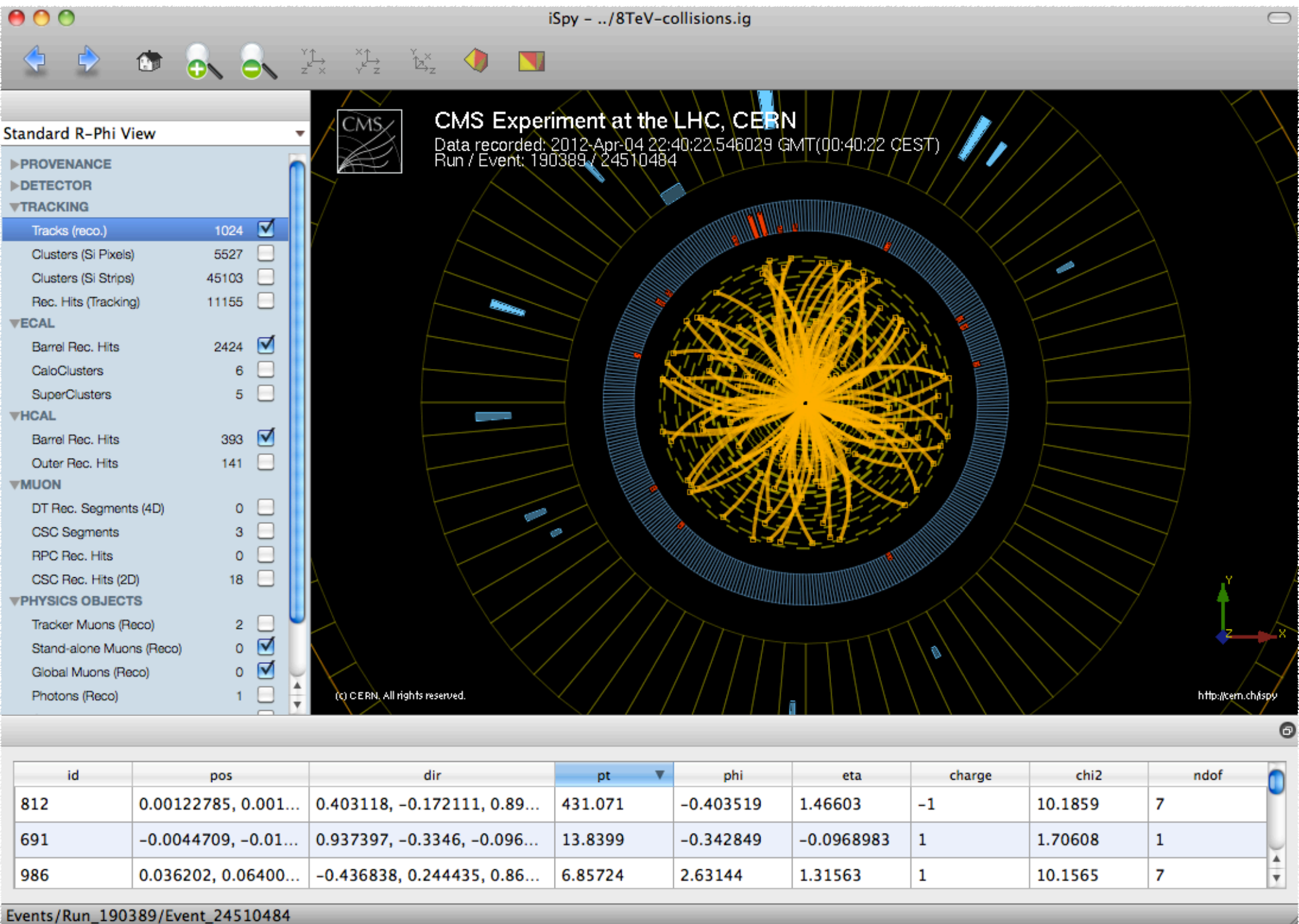
View layout is controlled via an iml file *e.g.*

```
<?xml version="1.0" encoding='UTF-8'?>
<layout>
    <camera position="-18.1, 8.6, 14.0" pointAt="0, 0, 0"
            scale="10.6" orthographic="true" rotating="true">
        <visibilityGroup>
            <view label="Standard 3D View">
                <collection label="Provenance/Event information"
                        spec="Event_V1:time:run:event:ls:orbit:bx"
                        draw="make3DEvent"/>
            </view>
        </visibilityGroup>
    </camera>
</layout>
```

- Default settings for style and views compiled into release
- Users can modify their own iml and iss files and run with iSpy as arguments on command line
- Many default drawing methods are provided (*e.g.* make3DBox, make3DLine, make3DTracks)
- New objects can be drawn with default methods by appropriate addition to iml file: no recompilation needed

Features/Views

- iSpy written using Qt4 and Open Inventor (Coin3D)
- Tree, table, 3D, RPhi, RZ, Lego views
- Simple and minimal controls and menus





Release


iSpy is distributed as fully-bound executable. Simply download and run.

Download

 **Linux (32bit)**
[Download version 1.5.0](#)

 **Linux (64bit)**
[Download version 1.5.0](#)

 **Mac (10.5/10.6/10.7)**
[Download version 1.5.0](#)

 **Windows (Vista or 7)**
[Download version 1.5.0](#)

<http://cern.ch/ispy>

Thanks to S Muzaffar, I Osborne, and L Tuura. L Lopera and Zhang Jinzhong. We acknowledge the support of the NSF and DOE