

Future evolution of CMS event display

Alja Mrak Tadel, UC San Diego
HSF Workshop, March 2017

Introduction

Fireworks has been in use for more than 8 years, 80 startups per day

covers a large variety of use cases

support has to continue for 10 - 20 years!

explore ways of decoupling core functionality from GUI and 3D graphics. This naturally leads to server-client separation with following advantages:

- data can be displayed anywhere
- remote data access integration is simplified
- possible to do automatic data and software compatibility matching

Roles of data server

- Even in the future we want a tool that explores data and not just displays some frozen representation of an event. Therefore need a full access to CMS event data model to read it and execute object methods
- Examples of server-client communication from Fireworks
 - event filtering
 - object filtering
 - dynamic table views - custom fields / arbitrary expressions
 - detail views (depending on data tier)

Server-Client example (1)

- **Collection filters:** specific to data type. A tab-complete can be use to validate or query filter expressions

The screenshot displays the cmsShow application interface. The main window shows a particle detector visualization with a central vertex and tracks. The interface includes a menu bar (File, Edit, View, Window, Help), a toolbar with navigation buttons, and a status bar showing 'Run 1', 'Lumi 2', 'Event 51', and 'Wed Dec 31 16:00:00 1969 PST'. A 'Filtering is OFF.' indicator is visible. The left sidebar shows a 'Summary View' with a table of collections and their properties.

Collection	pt	ef
EcalFromPFCand 0	1.1	2.3
EcalFromPFCand 1	5.1	2.2
EcalFromPFCand 2	1.9	1.9
EcalFromPFCand 3	2.2	1.8
EcalFromPFCand 4	27.1	1.8
EcalFromPFCand 5	3.9	1.8
EcalFromPFCand 6	3.1	1.7
EcalFromPFCand 7	1.4	1.7
EcalFromPFCand 8	2.1	1.7
EcalFromPFCand 9	1.2	1.6
EcalFromPFCand 10	1.9	1.5
EcalFromPFCand 11	1.2	1.5
EcalFromPFCand 12	1.4	1.4
EcalFromPFCand 13	1.5	1.4
EcalFromPFCand 14	2.1	1.4
EcalFromPFCand 15	1.3	1.3

The 'Collection Controller' window is open, showing the 'EcalFromPFCands' collection. The 'Filter' tab is selected, and the expression field contains the filter expression: `pt > 1.5 & (abs(pdgid) == 11 || abs(pdgid) == 22 || pdgid == 2)`. A 'Filter' button is visible next to the expression field.

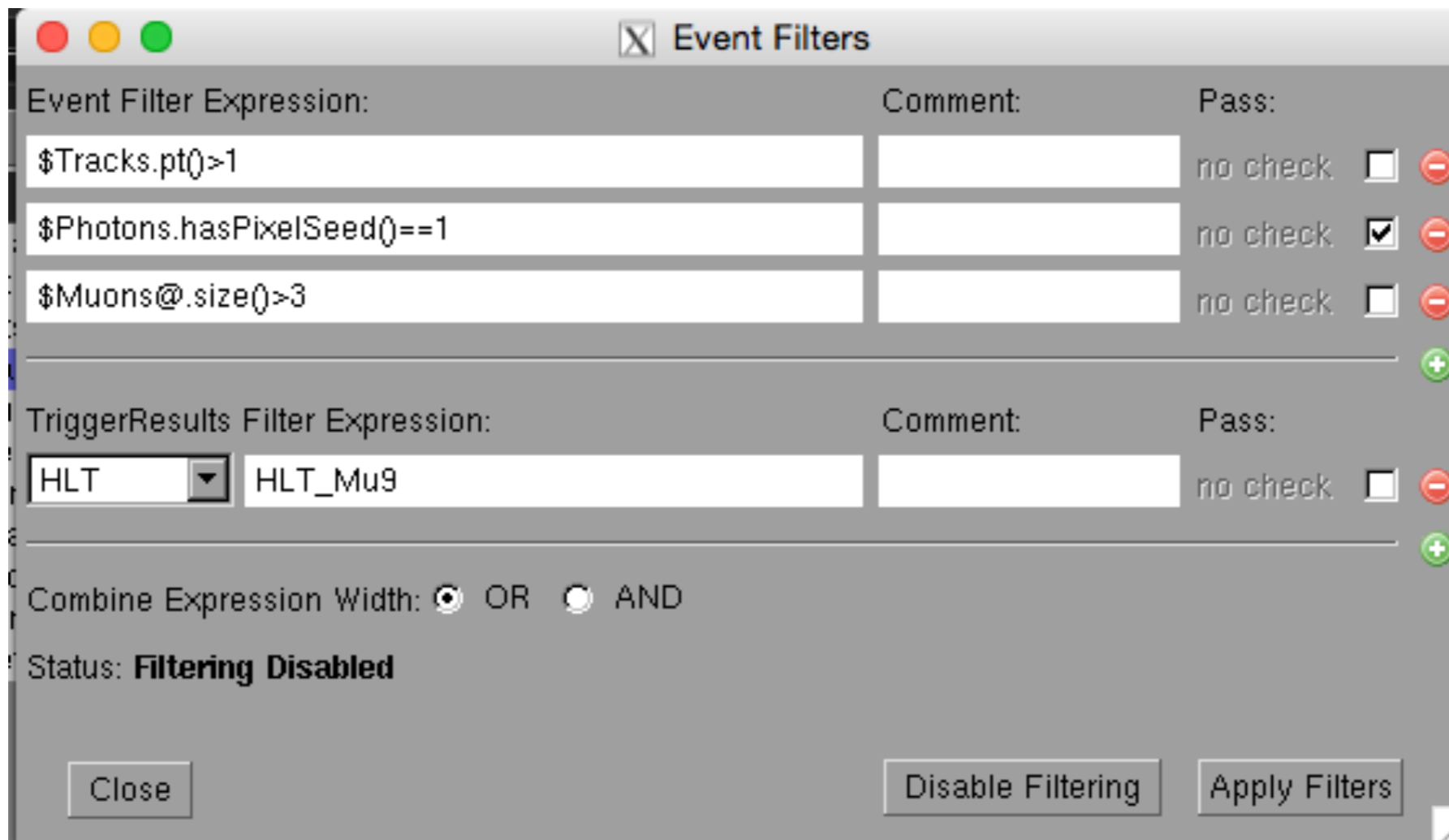
Server-Client example (2)

- **Dynamic table views** : new columns for collection type specific member can be added. A table complete can be used to validate or evaluate expressions

Table															
Collection Muons															
Column editor															
Title	tracker		Expression	isTrackerMuon		Precision	-2						Add	Delete	Modify
	pT	global	tracker	SA	calo	tr pt	eta	phi	matches	d0	d0 / d0Err	charge			
1	13.1	true	true	true	false	13.1	-0.392	1.049	1	0.034	14.787	1			
3	8.8	false	true	false	false	8.8	0.384	-1.677	1	-0.028	-13.355	1			
4	25.5	false	true	false	false	25.5	1.963	0.594	1	0.021	16.186	1			
5	11.3	false	true	false	false	11.3	-0.374	0.989	1	0.027	14.508	1			
6	12.9	false	true	false	false	12.9	-0.323	1.021	1	0.029	15.167	1			
7	15.6	false	true	false	false	15.6	-0.302	0.987	1	0.028	17.197	1			
8	8.5	false	true	false	false	8.5	-0.307	1.070	1	0.030	14.305	1			
9	35.2	false	true	false	false	35.2	1.940	0.573	1	0.017	11.269	-1			
10	18.0	false	true	false	false	18.0	1.965	0.615	1	0.021	17.079	-1			
0	5.1	true	false	true	false	5.1	-0.114	1.409	0	0.038	12.369	-1			
2	6.8	true	false	true	false	6.8	-0.373	1.029	0	-0.038	-1.971	-1			

Server-Client example (3)

- **Event filtering** is used for selection of interesting events based on a complex selection. Anything that can be used as selection in TTree::Draw command in FWLite/Root is accepted.



EVE as object server

- use ROOT package for visualization (EVE) as an object server
 - produce graphics objects
 - perform scene, projection and object state management
- two main options for client implementation:
 - write a simple GUI application (still platform dependent)
 - web browser application

JSROOT as client

- Good base to start with and build upon:
 - integration with ROOT's built-in http server (civet)
 - can already request and render simple Eve classes (tracks and point sets), supports TGeo shapes
- We are working on prototype 0:
 - embed ROOT's http server in Fireworks
 - serve visualization objects to JSROOT client
 - implement next / previous event, etc.
 - basic exploration, ~1 month work allowing us to plan further

Client-Server prototype '0'



Extending Eve & JSROOT (1)

- Prototype '0' limitations:
 - Eve objects were not designed to be streamable. Can fix that by adding a streaming function or exclude in the source members that don't have to be streamed.
 - In some cases visualization primitives are generated in ROOT's GL classes. This problem can be fixed by intermediate layer or additional work on client's side
- Provide general mechanism for interaction with server configuration variables, some of which can be physics object and view dependent
 - filtering a collection
 - change projection parameters (e.g. increase distortion, or change split plane in RhoZ projection)
 - change energy scale (pt->cm for calo towers, jets, and MET)

Extending Eve & JSROOT (2)

design questions

- Common content of method execution request created on the client side
- Common content of update request created on the server side. This one can depend on granularity of updates.
- Sort out details of method execution on server using cling.
- Check ajax messaging be fast enough or maybe use web sockets
- Do we need to introduce other packaging formats than json (e.g. c-json, or google prototype buffers)

Conclusion

- Fireworks is stable and will be used long term.
- Exploring web interfaces using Eve and JSROOT
- limited client-server event display prototype ready until the next workshop
- In parallel, attempt to formulate data-types and server-client protocol for generic HEP visualization