

## PlotUtils Management - Feature #22856

### Add xrootd compatibility to PU::ChainWrapper

07/02/2019 09:16 AM - Benjamin Messerly

<b>Status:</b>	Resolved	<b>Start date:</b>	07/02/2019
<b>Priority:</b>	Normal	<b>Due date:</b>	
<b>Assignee:</b>	Benjamin Messerly	<b>% Done:</b>	100%
<b>Category:</b>		<b>Estimated time:</b>	0.00 hour
<b>Target version:</b>		<b>Spent time:</b>	3.00 hours

#### Description

Currently, PU::ChainWrapper uses globs to add root files to its underlying TChain. This may interfere with xrootd functionality. Or it may not. Andrew O seemed to have opinions about this. It may be a simple fix.

Related tasks, which I'm not going to make an issue for: we may want to expand the PU::DefaultCVUniverse to hold a TChain, instead of a CHW.

#### History

##### #1 - 07/02/2019 09:30 AM - Andrew Olivier

Here's what I've learned about this issue so far:

- When using xrootd, I suspect that UNIX glob() won't pick up the directories used because it doesn't "know anything" about the xrootd protocol. I think the same would be true if I gave it a http URL or something. I have not yet tried to find symptoms of this problem though.
- I think TChain uses TSystem to wrap both UNIX globbing and xrootd globbing as documented for ROOT 6 at line 365 of [https://root.cern.ch/doc/master/TChain\\_8cxx\\_source.html#l00346](https://root.cern.ch/doc/master/TChain_8cxx_source.html#l00346). lirc, there was once a comment somewhere in the MergeTool about why we use UNIX (POSIX?) glob() instead of letting TRegex or some other ROOT component do it. I think it had something to do with ROOT not supporting everything glob() does.
- The xrootd API, documented at <http://xrootd.org/doc/xrdcl-docs/xrdcldocs.pdf>, seems to give me all of the tools I'd need to write a glob() function except the regular expression engine. The c++11 <regex> header or TRegex is probably more than sufficient for that. Then, we'd just have to write a FIFO/recursive function to keep doing matches in each directory.
- Since we'd probably be issuing lots of queries to xrootd, I'd be a little worried about latency piling up when getting the list of files to run over. If that actually happens, maybe we can grab "chunks" of the filesystem at once that we might not even use? Seems like a problem to ignore until we actually have it, but it should be mentioned in comments.

My views on action items:

- Low-hanging fruit: Just add a check for an "xroot" protocol on URLs in ChainWrapper and use TSystem to fix the problem to first order right now. Should have no impact on current glob() functionality.
- Find out why we don't use TSystem for glob()ing within the "local" filesystem. If there's still a reason that we feel strongly about, it would be a nice feature to eventually write a glob()-like function that uses xrootd instead. I'm interested in solving that problem if no one else wants to do it. This could take a while to test though.

##### #2 - 07/05/2019 03:16 PM - Andrew Olivier

- Status changed from New to Work in progress

Wrote a ROOT macro to demonstrate that ChainWrapper cannot find files specified with xrootd URLs. When ChainWrapper gets any(?) file name, it feeds it to glob(). If glob() fails to find any files, ChainWrapper exit(1)s immediately (also something we should fix). Test script will be committed with an executable suitable to be used in a "make test" target when I think I've solved the problem. I'll hold off on committing for a while so as not to interfere with testing of standalone PlotUtils.

##### #3 - 07/11/2019 06:59 PM - Andrew Olivier

- % Done changed from 0 to 100

- Status changed from Work in progress to Resolved

From Slack #PlotUtils:

I have a first draft of ChainWrapper ready that:

- 1) Lets me use xrootd URLs to Add() files. This includes a first draft of glob()-like functionality with different wildcard syntax for now.
- 2) Still default to exit()ing on failing to Add() any files, but there is now a compile-time constant that could cause an exception to be thrown instead. If you build PlotUtils to throw the exception, you have the option to catch it and maybe try to get files a different way.

This should only add functionality and not affect any existing analysis code that uses ChainWrapper. I later figured out how to get "grep-style" regular expressions to work in addition to what I think is PCRE. I have some simple test scripts too, but I don't yet know how to deploy them. Note that you

can use `PlotUtils::glob()` even if you're not using `ChainWrapper` by including `"PlotUtils/ROOTglob.h"` and linking against `libplotutils`.

2) doesn't make any concrete change yet. It's just a proposal that I wrote an implementation for and that users can enable by undefining a preprocessor macro.