Table of Contents

3.3.1 Copy and Merge Files

Complete: 🟩
Detailed Review status

Goals of this page:

The goal of this page is to learn how to work with data samples by copying few events from a data file to your local area. You would also learn how to merge data files. Note that the data `ROOT` files are in EDM format, so they are also called EDM files. The full information on finding data samples is given in WorkBookLocatingDataSamples.

Access Data In a CMSSW Job

When you copy a few events to your working directory like in the above example, it is mainly to run a test code quickly or just for the sake of learning etc. But normally you would access the data in your cmsRun job by accessing it in a local storage element (for example, castor at CERN or say at Fermilab). You will read more about it in Section 4.1.1. Before that you would need some experience using cmsRun and understanding `python` configuration files since you will be doing more work than just copying data.

Examples of accessing the data

When you start your analysis, you will locate your data in Data Aggregation System (DAS) which is described in WorkBookLocatingDataSamples. In this page we give instructions on how to get started with a small data sample.

To perform this exercise with the default shell of `bash`:

```
source /cvmfs/cms.cern.ch/cmsset_default.sh
voms-proxy-init -voms cms
```

or on `tcsh`:

```
source /cvmfs/cms.cern.ch/cmsset_default.csh
voms-proxy-init -voms cms
```

If you do not have a grid certificate in the CMS VO, please get one following these instructions: WorkBookStartingGrid

Copy Data Locally

The first example of accessing data is to copy a small amount of data from the local storage element (e.g. castor at CERN) to your own area and study the data directly with FWlite. You may choose different data source looking at DAS and verify the CMSSW version using edm tools. Let us start with a very simple python configuration script as shown below and call it `copy_cfg.py`:

```
import FWCore.ParameterSet.Config as cms

# Give the process a name
process = cms.Process("PickEvent")
```

```
# Tell the process which files to use as the source
process.source = cms.Source ("PoolSource",
          fileNames = cms.untracked.vstring ("/store/relval/CMSSW_5_3_15/RelValPyquen_ZeemumuJets
)

# tell the process to only run over 100 events (-1 would mean run over
#  everything
process.maxEvents = cms.untracked.PSet(
            input = cms.untracked.int32 (100)

)

# Tell the process what filename to use to save the output
process.Out = cms.OutputModule("PoolOutputModule",
         fileName = cms.untracked.string ("MyOutputFile.root")
)

# make sure everything is hooked up
process.end = cms.EndPath(process.Out)
```

Save these lines in a file named `copy_cfg.py`.

Before you run this script, first setup the `CMSSW` release as below: (`cmsrel` command is needed only if you do not have yet the CMSSW_directory) :

```
ssh lxplus.cern.ch
source /cvmfs/cms.cern.ch/cmsset_default.sh
cd ~/scratch0
cmsrel CMSSW_11_1_0
cd CMSSW_11_1_0/src
cmsenv


voms-proxy-init -voms cms
```

and then run the script as follows:

```
cmsRun copy_cfg.py
```

Users intending to reproduce this exercise on LPC machines should log into cmslpc-sl7.fnal.gov with their respective usernames and do instead, on `bash`:

```
source /cvmfs/cms.cern.ch/cmsset_default.sh
voms-proxy-init -voms cms
cd nobackup/
cmsrel CMSSW_11_1_0
cd CMSSW_11_1_0/src/
cmsenv
cmsRun copy_cfg.py
```

and on `tcsh`:

```
source /cvmfs/cms.cern.ch/cmsset_default.csh
voms-proxy-init -voms cms
cd nobackup/
cmsrel CMSSW_11_1_0
cd CMSSW_11_1_0/src/
cmsenv
cmsRun copy_cfg.py
```

When you run this command the output will look like this:

▸ Show result... ▾ Hide result...

```
26-Feb-2014 17:19:17 CET  Initiating request to open file root://eoscms//eos/cms/store/relval/CMS
140226 17:19:17 30642 Xrd: GoToAnotherServer: Going to: lxfsra06a03.cern.ch:1095
26-Feb-2014 17:19:18 CET  Successfully opened file root://eoscms//eos/cms/store/relval/CMSSW_5_3_
Begin processing the 1st record. Run 1, Event 1, LumiSection 666666 at 26-Feb-2014 17:19:21.639 C
Begin processing the 2nd record. Run 1, Event 2, LumiSection 666666 at 26-Feb-2014 17:19:21.640 C
...................
...................
...................
Begin processing the 99th record. Run 1, Event 84, LumiSection 666682 at 26-Feb-2014 17:19:21.852
Begin processing the 100th record. Run 1, Event 85, LumiSection 666682 at 26-Feb-2014 17:19:21.85
26-Feb-2014 17:19:43 CET  Closed file root://eoscms//eos/cms/store/relval/CMSSW_5_3_15/RelValPyqu

=============================================

MessageLogger Summary

 type     category        sev    module        subroutine        count    total
 ---- -------------------- -- ---------------- ----------------  -----    -----
    1 fileAction           -s file_close                             1        1
    2 fileAction           -s file_open                              2        2

 type    category    Examples: run/evt        run/evt          run/evt
 ---- -------------------- ---------------- ---------------- ----------------
    1 fileAction           PostEndRun
    2 fileAction           pre-events       pre-events

Severity    # Occurrences   Total Occurrences
--------    -------------   -----------------
System                  3                   3
```

The execution of the above command will result in copying 100 events from `/store/relval/CMSSW_5_3_15/RelValPyquen_ZeemumuJets_pt10_2760GeV/DQM/PU_STARTHI53V10A_TEST_feb14-` to an output file called `MyOutputFile.root`.

Introduction to `copyPickMerge_cfg.py` and `edmCopyPickMerge`

However, there is a more elegant and simple way to copy events. This elegant way gets rid of modifying the `copy_cfg.py` kind of file every time you need to change the input/output file name or number of events.

You may look inside `copyPickMerge_cfg.py` to find out that it is very similar to the `copy_cfg.py` configuration above, except that it is setup that you can change many options (*e.g.*, the input and output files) from the command line instead of having to edit the file.

The important lines to observe inside copyPickMerge_cfg.py are:

```
fileNames = cms.untracked.vstring (options.inputFiles),
```

takes the name of the input file(s) as a string.

```
input = cms.untracked.int32 (options.maxEvents)
```

is used to specify the number of events to be read/copied, and

```
fileName = cms.untracked.string (options.outputFile)
```

is used to specify the name of the output ROOT file. They serve the same purpose as the following three lines taken from the copy_cfg.py above:

```
...
          fileNames = cms.untracked.vstring ("/store/relval/CMSSW_5_3_15/RelValPyquen_ZeemumuJets
...
            input = cms.untracked.int32 (100)
...
         fileName = cms.untracked.string ("MyOutputFile.root")
...
```

but there is no need to edit this file every time a change is needed, instead, the input parameters are just given from the command line.

You may copy/paste the code lines inside copyPickMerge_cfg.py in your local directory, and you could accomplish the same thing you did with copy_cfg.py above by:

```
cmsRun copyPickMerge_cfg.py inputFiles=/store/relval/CMSSW_5_3_15/RelValPyquen_ZeemumuJets_pt10_2
```

Since part of the beauty of copyPickMerge_cfg.py is that you don't have to edit it, we put it in CVS in CMS.PhysicsTools/Utilities/Configuration. To facilitate using it, there is an edm utility called edmCopyPickMerge, located in the same package, that locates the python configuration copyPickMerge_cfg.py uses it with cmsRun. If you don't initialize the grid environment including the certificate, the data file from which you are trying to copy events should be available locally. If you have the grid certificated initialized as metioned above, i.e.

```
source /cvmfs/cms.cern.ch/cmsset_default.sh
```

```
voms-proxy-init -voms cms
```

the script will try to find the right files for you from a remote storage element, as long as it's not on Tape. Files on Tape are not accessible with this method and must be transferred first with a data management system such as Rucio.

Just type and use edmCopyPickMerge as follows to copy say, 100 events, from a file available locally

```
edmCopyPickMerge \
  inputFiles=/store/relval/CMSSW_5_3_15/RelValPyquen_ZeemumuJets_pt10_2760GeV/DQM/PU_STAR
THI53V10A_TEST_feb14-v3/00000/FE0AF9FB-C196-E311-8678-0025904CF75A.root \  outputFile=MyOutputFil
  maxEvents=100
```

When you execute the above command, the output should look like this.

Show result... Hide result...

```
26-Feb-2014 17:36:16 CET  Initiating request to open file root://eoscms//eos/cms/store/relval/CMS
140226 17:36:16 4176 Xrd: GoToAnotherServer: Going to: lxfsra06a03.cern.ch:1095
26-Feb-2014 17:36:17 CET  Successfully opened file root://eoscms//eos/cms/store/relval/CMSSW_5_3_
Begin processing the 1st record. Run 1, Event 1, LumiSection 666666 at 26-Feb-2014 17:36:20.551 C
Begin processing the 2nd record. Run 1, Event 2, LumiSection 666666 at 26-Feb-2014 17:36:20.552 C
..........................
..........................
..........................
Begin processing the 100th record. Run 1, Event 85, LumiSection 666682 at 26-Feb-2014 17:36:20.77
26-Feb-2014 17:36:42 CET  Closed file root://eoscms//eos/cms/store/relval/CMSSW_5_3_15/RelValPyqu

=============================================

MessageLogger Summary
```

```
 type     category        sev    module        subroutine        count    total
 ---- -------------------- -- ---------------- ----------------  -----    -----
    1 fileAction           -s file_close                             1        1
    2 fileAction           -s file_open                              2        2

 type    category    Examples: run/evt        run/evt          run/evt
 ---- -------------------- ---------------- ---------------- ----------------
    1 fileAction           PostEndRun
    2 fileAction           pre-events       pre-events

Severity    # Occurrences   Total Occurrences
--------    -------------   -----------------
System                  3                   3
```

A successful copying of 100 events will result in an output `ROOT` file called `MyOutputFile_numEvent100.root` . If you do not specify the name of the output file then a file with a default name `output_numEvent100.root` is created. Make sure you have enough disk space to write the file out.

If you do not have the data file located locally, you can also run a Grid Job. For more information on this part and other details have a look at WorkBookPickEvents

Merge EDM files

To merge EDM files, one can again use `edmCopyPickMerge` utility which is in CMSSW, any current version.

To merge several files together:

```
edmCopyPickMerge inputFiles=first.root,second.root,third.root outputFile=output.root maxSize=1000
```

where the input files are `first.root`, `second.root`, and `third.root` and the output file is `output.root` or

```
edmCopyPickMerge inputFiles_load=listOfInputFiles.txt outputFile=output.root maxSize=100000
```

where `listOfInputFiles.txt` is a text file containing a list of input files (one file per line) and `output.root` is the output file and `1000000` is the maximum size of the output file in Kb (*e.g.,* 1000000 Kb = 1 Gb).

Important: In cmsRun, when giving it local files as input, the file names must be prefixed by `file:`. For example, `first.root` would be written `file:first.root`.

How to copy a particular event

Note: `edmPickEvents.py` is a tool that will find the necessary files and run the configuration file below given a dataset name and a list of events.

There is a standard config file that helps you extracting single events from CMS data files. The file and the events can be specified at command line:

```
cmsRun pickEvent_cfg.py inputFiles=file1.root \
       eventsToProcess=123592:334:755009,123592:23:392793,123592:42:79142 \
       outputFile=output.root
```

The config file `pickEvent_cfg.py` is as follows:

Show result... Hide result...

```
import FWCore.ParameterSet.Config as cms
```

```
from FWCore.ParameterSet.VarParsing import VarParsing

options = VarParsing ('analysis')
# add a list of strings for events to process
options.register ('eventsToProcess',
                                  '',
                                  VarParsing.multiplicity.list,
                                  VarParsing.varType.string,
                                  "Events to process")
options.parseArguments()

process = cms.Process("PickEvent")
process.source = cms.Source ("PoolSource",
          fileNames = cms.untracked.vstring (options.inputFiles),
          eventsToProcess = cms.untracked.VEventRange (options.eventsToProcess)
)

process.Out = cms.OutputModule("PoolOutputModule",
        fileName = cms.untracked.string (options.outputFile)
)

process.end = cms.EndPath(process.Out)
```

Note: In `123592:334:755009`, the **first entry** is the RUN number, the **second entry** is the LUMI block number, and the **third entry** the EVENT number. If the specified event is not found, the config file will not complain but will also not write that event to the output. So one needs to know which event to copy. Also make sure you have the privilege to write the output file to a directory like shown above (`output.root`). Also make sure you have enough space to copy.

Important: In cmsRun, when giving it local files as input, the file names must be prefixed by `file:`. For example, `first.root` would be written `file:first.root`.

Find Collision Data

The updated information on

- Main page for Physics Performance Datasets
- 13 TeV collision files of 2015: Collisions2015Analysis
- 8 TeV collision files of 2012: Collisions2012Analysis
- 7 TeV collision files of 2011: Collisions2011Analysis
- 7 TeV collision files of 2010: Collisions2010Analysis
- the first 7 TeV collision files: FirstCollisionsAnalysis.

Review status

Reviewer/Editor and Date (copy from screen)	Comments
MargueriteTonjes - 21 Oct 2020	updated CMSSW version and got rid of afs, added Rucio, still need new RelVal file referenced
XuanChen - 07 Jul 2014	Updated cvs to github
AntonioMorelosPineda - 26-Feb-2014	Update sample file
HengneLiUVa - 24-May-2013	add note of requirement of grid env.
AntonioMorelosPineda - 18-May-2013	Updates to 5_3_7 files

KatiLassilaPerini - 24-Mar-2011	Updates to 4_1_3 files
KatiLassilaPerini 11 Dec 2009	this page now explains how to get some events quickly, all further details are in WorkBookLocatingDataSamples and in WorkBookDataManagementBackground
SudhirMalik- 4 Nov 2009	updated examples to CMSSW_3_3_1, updated DBS snapshots
KatiLassilaPerini - 28 Feb 2008	removed the LPC samples

Detailed comments 7-Nov-2012 ▸ Hide ▾
I went through chapter 3 section 3 subsection 1. The information is relevant and clear.

I updated a coment on DAS, DBS is no longer used.

Responsible: SudhirMalik
Last reviewed by: AntonioMorelosPineda - 18 May 2013

This topic: CMSPublic > WorkBookDataSamples
Topic revision: r59 - 2021-02-18 - unknown