

Table of Contents

TestFileDB, a database of test files for use in automated software tests.....	1
Table of Contents.....	1
Why should I use it?.....	1
Where is it?.....	1
Using and abusing the package:.....	1
- How can I access it?.....	1
- How can I browse what's there?.....	2
- How can I access it through options?.....	2
- How can I use it in my project?.....	2
- Can I maintain my own list of files in this way even if they are not for general consumption?...2	2
Editing and adding.....	2
- Obtaining a local copy for editing.....	3
- Editing.....	3
- Database tags:.....	3
- Automatic scripts for adding to the database:.....	3
- Automatic scripts for adding files to CERN-SWTEST/EOS:.....	4
Copying files directly to CERN-SWTEST.....	5
- Commit and use in nightlies.....	5

TestFileDB, a database of test files for use in automated software tests

The TestFileDB is a small python database of PFNs of files used for automated software tests. The database lives within the existing software structure and can thus be used in any project/package.

Table of Contents

Why should I use it?

1. **Inviolability.** The database refers to files by a logical dataset name, or a qmtest name. This allows for an extra layer of obfuscation whereby new files can be chosen or files can be moved to different locations without the test authors needing to change a single line of tests.
2. **Accountability.** Not only does TestFileDB keep the PFNs, but also a whole host of file metadata about who produced it, why, and under which conditions. This vastly simplifies management of test files and debugging.
3. **Usability.** TestFileDB comes with a framework of useful and extensible features since it is a full python module. Migration of files to the standard test element is simplified, inclusion of a file into a qmtest or an options file is simplified, etc. see below.

Where is it?

TestFileDB lives under PRConfig, the DBASE package designed to hold options files and tests for the LHCb Performance and Regressions testing system.

- gitlab: <https://gitlab.cern.ch/lhcb-datapkg/PRConfig>
 - gitlab: <https://gitlab.cern.ch/lhcb-datapkg/PRConfig/blob/master/python/PRConfig/TestFileDB.py>
-

Using and abusing the package:

- How can I access it?

TestFileDB is a full python module including inbuilt help. The actual database is stored in a simple python dictionary within the module, called `test_file_db`. Try:

```
SetupProject LHCB --use PRConfig  
python
```

or

```
lb-run --use=PRConfig LHCB/v42r1 python
```

and then in python:

```
from PRConfig import TestFileDB  
help(TestFileDB)  
print TestFileDB.test_file_db.keys()  
print TestFileDB.test_file_db["2012_raw_default"]
```

- How can I browse what's there?

Either:

1. Read through the python module in SVN:
<https://gitlab.cern.ch/lhcb-datapkg/PRConfig/blob/master/PRConfig/python/PRConfig/TestFileDB.py>
2. Browse through the entries in TestFileDB as shown above
3. Browse through using the browsing script `$PRCONFIGROOT/scripts/BrowseTestFiles.py`. The inbuilt browser works like browsing a directory structure where files are separated into "directories" based on their metadata.
4. Browse manually with the low-level helper commands. This is the basic part that is wrapped in the `BrowseTestFiles.py` script.

```
from PRConfig import TestFileDB
help(TestFileDB)
from PRConfig import TestFileUtils
help(TestFileUtils)
help(TestFileUtils.browseTestFiles)
TestFileUtils.browseTestFiles(TestFileDB.test_file_db)
```

- How can I access it through options?

- The following is completely valid python options:

```
from Configurables import Moore
from PRConfig import TestFileDB
TestFileDB.test_file_db["2012_raw_default"].run(configurable=Moore())
```

- This procedure will add the test files through IOHelper, and configure the configurable Moore with the database tags stored in the test file metadata, and simulation True/False flag.
- The run method has options to control its behaviour, maybe you want to not set the database tags, maybe you want to make sure any existing files from EventSelector are deleted? See the help or read the module:
<https://gitlab.cern.ch/lhcb-datapkg/PRConfig/blob/master/python/PRConfig/TestFileObjects.py#L61>
- This procedure is used in several of the existing qmtest frameworks such as the tests in moore:
<https://gitlab.cern.ch/lhcb/Moore/tree/2018-patches/Hlt/Moore/tests>

- How can I use it in my project?

PRConfig should be available in all LHCb applications. If not you may need to explicitly include it in your lb-run statement:

```
lb-run --use=PRConfig SomeProject/vXrY
```

- Can I maintain my own list of files in this way even if they are not for general consumption?

Yes! Please do. You can create a new database starting from a python dictionary and importing the base class object `testfiles` from `TestFileObjects` `import testfiles`.

Editing and adding

- Obtaining a local copy for editing

This obviously depends where (in which project) you're planning on adding the tests. The example below is for Moore, replace with your version.

```
lb-dev Moore/latest
cd MooreDev_HEAD
git lb-clone-pkg PRConfig
make
```

Edit, commit with release notes.

- Editing

In general make sure:

- Your files are accessible for anyone who may run from the database. Usually this means they need to be at CERN on EOS, castor or in some public directory (DEV).
- You can access the files without needing a grid certificate.
- The files listed are PFNs.
- You have added as much information as possible to the metadata of the files.
- You have not missed any required metadata, the automatic script below will ensure you don't miss anything, or you can check with the TestReadable script:
<https://gitlab.cern.ch/lhcb-datapkg/PRConfig/blob/master/scripts/TestReadable.py>

Remember that this is a full python module, so any valid python to create the entries in the database is allowed, even something procedural.

- Database tags:

One of the key features of the TestFileDB is that it can store the database tags appropriate for reading the files. This is a very useful feature, since the database tags might sit far outside the standard tags for random test files. Please be mindful of this when filling the database. Please use the following priority to decide which tags to use:

1. **The tags used in creation of the file. If there is any way of you finding this out you are encouraged to do so.**
2. A compatible pair of tags which are necessary updates to the tags used to create the file. In case the original tags are not compatible with current software, small updates may be required.
3. **If all else fails, an appropriate pair of tags that you deem appropriate for running on this file can be entered. In case you really cannot find out what was originally used, or what is originally used is completely so obsolete to be entirely irrelevant, then please enter any two compatible tags which you deem appropriate, but only if this is really really necessary.**

- Automatic scripts for adding to the database:

If you already have a python file which adds test files through IOHelper or EventSelector there is a script which will copy this to the TestFileDB and ask you for the metadata.

- gitlab: <https://gitlab.cern.ch/lhcb-datapkg/PRConfig/blob/master/scripts/AddToTestFilesDB.py>

To use this:

1. Prepare local copy as above.
2. Ensure you know the following information about your files:
 - ◆ What database tags were used in their creation? Please do work this out, it is possibly the most valuable piece of metadata about files in TestFileDB.
3. Convert all LFNs to PFNs
4. run the script
 - ◆ `./AddToTestFilesDB.py --help`
 - ◆ `./AddToTestFilesDB.py someoptsfile_or_a_PFN`

You can then check the files are readable for you with the script `TestReadable.py`:
<https://gitlab.cern.ch/lhcb-datapkg/PRConfig/blob/master/scripts/TestReadable.py>

- Automatic scripts for adding files to CERN-SWTEST/EOS:

ACTION RESTRICTED TO RELEASE MANAGERS

For files actually used in our automated nightly tests, especially centrally produced files we need to make sure they are not tied to the validity of a specific user account or production obsolescence cycle. So, please upload your files to the CERN-SWTEST element using the script provided:

- gitlab: <https://gitlab.cern.ch/lhcb-datapkg/PRConfig/blob/master/scripts/MigrateToSWTEST.py>

To use this:

1. Prepare local copy as above.
2. Insert files into the TestFileDB as above, or manually
3. Prepare your environment
 - ◆ You will need an environment with `AppConfig`, `PRConfig`, `LHCbDirac` and `gaudirun.py`. Again the example below is if you are using a base project of Moore

```
SetupProject LHCbDirac
SetupProject Moore v21r0 --use PRConfig --use AppConfig
#next line not always required, depends on the intricacies of cmt
cd PRConfig/cmt
source setup.ssh
```

1. Obtain a grid proxy:
 - ◆ `lhcb_user`: you will be able to migrate user files but not existing production files to CERN_SWTEST
 - ◆ `lhcb_prmgr`: you will be able to migrate existing production files but not user files to CERN_SWTEST

```
lhcb-proxy-init --group=lhcb_prmgr
```

1. Execute the script:
 - ◆ `../scripts/MigrateToSWTEST.py --help`
 - ◆ `../scripts/MigrateToSWTEST.py`

You can then check the files are readable for you with the script `TestReadable.py`:
<https://gitlab.cern.ch/lhcb-datapkg/PRConfig/blob/master/scripts/TestReadable.py>

Copying files directly to CERN-SWTEST

You can copy the files directly to CERN-SWTEST, by using the following commands:

```
lb-run -c best LHCbDirac/prod bash --norc
lhcb-proxy-init -g lhcb_prmgr
dirac-dms-add-file /lhcb/swtest/<subdir>/myfile myfile CERN-SWTEST
```

- Commit and use in nightlies

Don't forget to edit the release notes before you commit! Contact Marco Cattaneo and Marco Clemencic if you want these files to be immediately accessible in the nightlies, and/or request a release of PRConfig by contacting Ben Couturier and Marco Clemencic.

-- RobLambert - 20 Jan 2014

This topic: LHCb > TestFileDB

Topic revision: r17 - 2019-07-17 - ChristopheHaen



Copyright &© 2008-2019 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback