THIS PAGE IS OBSOLETE

workflows

Before running the scripts below you first need to setup the appropriate environment. I use a WMAgent machine and do this:

```
source /data/srv/wmagent/current/apps/wmagent/etc/profile.d/init.sh
source /data/srv/wmagent/current/sw.pre.amaltaro/slc5_amd64_gcc461/cms/dbs3-client/3.2.6a/etc/pro
```

Prestaging

The first step is to subscribe the input dataset to a Tier-1 disk endpoint if necessary. Usually this will be on the same site which has the GEN-SIM on tape, but can be a different Tier-1 if necessary, for example if the custodial Tier-1 has too much work. If there is no custodial site, then subscribe the GEN-SIM to an appropriate Tier-1 based on the current work at each site.

I use the script https://github.com/alahiff/WmAgentScripts/blob/AndrewFixes/listWorkflows.py to produce a list of workflows in the assignment-approved state:

```
python listWorkflows.py
```

The input datasets and custodial sites are also included in the output. Example:

```
bash-3.2$ python listWorkflows.py
jen_a_BTV-Spring14miniaod-00071_00077_v0__141119_181505_4141 /QCD_Pt-170to300_MuEnrichedPt5_Tune4
```

I then manually make the subscriptions to the appropriate sites from the PhEDEx page https://cmsweb.cern.ch/phedex/prod/Request::Create?type=xfer

A script is in testing which automatically subscribes GEN-SIM datasets to disk on custodial Tier-1s (the Phys14DR campaign has mostly been done this way).

Assigning ReDigi workflows

The script https://github.com/alahiff/WmAgentScripts/blob/AndrewFixes/assignWorkflowsAuto.py is used to assign workflows. It's designed to be run twice - firstly as a "dry-run" to ensure everything is fine (e.g. acquisition era, ProcessingString, etc) then again for real.

Example checking assignment of a single workflow:

```
bash-3.2$ python assignWorkflowsAuto.py -w pdmvserv_B2G-Summer12DR53X-00799_00332_v0__141022_1605
Would assign  pdmvserv_B2G-Summer12DR53X-00799_00332_v0__141022_160538_7286  with  Acquisition Er
```

This script needs to be run again with the `-e` option in order to actually assign the workflow.

There are 3 ways the script can be used

- `-w` option: specify a single workflow
- `-f` option: specify a file containing a list of workflows, one per line
- neither of the above: all workflows in the assignment-approved state are obtained from the WMStats API and are considered

To force a workflow to be assigned to a specific site use the `-s` option. If this is not specified, the workflow will be assigned to the Tier-1 which has the complete input dataset on disk. By default the output datasets will be custodial at the Tier-1 that the workflow was assigned to. To specify an alternative Tier-1, use the `-c`

option. If you want to run a workflow at a site that does not have the input data you need to specify the `-o` option to enable xrootd to be used.

Run `python assignWorkflowsAuto.py -h` for more information.

Over the past few weeks I've almost entirely just let it assign all workflows in assignment-approved automatically as soon as it finds that prestaging is complete.

Announcing workflows

I use the script https://github.com/alahiff/WmAgentScripts/blob/AndrewFixes/listWorkflows.py to produce a list of workflows in the closed-out state:

```
python listWorkflows.py closed-out
```

The script https://github.com/CMSCompOps/WmAgentScripts/blob/master/WorkflowPercentage.py can be used to both generate a list of the output datasets and check how complete they are.

However, now I use the script https://github.com/alahiff/WmAgentScripts/blob/AndrewFixes/announceReDigiWorkflows.py and keep the output:

```
python announceReDigiWorkflows.py > announce
```

To get a list of workflows which can be announced:

```
cat announce | grep Announce | awk '{print $2}'
```

and to get the corresponding list of datasets

```
cat announce | grep DATA | awk '{print $2}'
```

The script has the ability to set output datasets to VALID and change the workflow status to announced, but for safety I just have the commented out.

For the case of run-dependent MC, you need to check that the number of jobs created was above the appropriate threshold (500 for PU_RD1 or 2000 for PU_RD2).

If everything is seems ok (generally output datasets should be >=95% and <=100% complete):

- the script https://github.com/CMSCompOps/WmAgentScripts/blob/master/announceWorkflows.py is used to change the status of workflows from closed-out to announced
- the script setDatasetStatusDBS3.py is used to change the status of the output datasets to VALID (`python setDatasetStatusDBS3.py --dataset=name_of_dataset --status=VALID`)
- the script https://github.com/cpausmit/IntelROCCS/blob/master/DataDealer/src/assignDatasetToSite.py is used to subscript AODSIM/MINIAODSIM to Tier-2s (`python assignDatasetToSite.py --dataset=name_of_dataset --exec`)
- a message should then be sent to the Datasets Announcements HyperNews

---

This topic: Sandbox > HandlingWorkflows
Topic revision: r7 - 2015-03-30 - AndrewLahiff

or Ideas, requests, problems regarding TWiki? use Discourse or Send feedback