

---

# **decisionengine**

***Release 1.1.1***

**Fermi Research Alliance, LLC.**

**Jun 29, 2021**



## CONTENTS

<b>1</b>	<b>Release Notes</b>	<b>3</b>
<b>2</b>	<b>Developer Documentation</b>	<b>11</b>
<b>3</b>	<b>Jenkins CI pipeline</b>	<b>13</b>
<b>4</b>	<b>Source code</b>	<b>19</b>
<b>5</b>	<b>Indices and tables</b>	<b>29</b>
	<b>Python Module Index</b>	<b>31</b>
	<b>Index</b>	<b>33</b>



The decisionengine\_modules package contains provider specific implementations of base classes provided by the decision engine framework.



## RELEASE NOTES

### 1.1 Release 1.6.2

Patch level (bug fix) release.

Correction map for GlideinWMS Factory Entries added. (#323, PR#324) Fixed imports and a typo.

#### 1.1.1 Full list of commits since version 1.6.1

4f6e313: Correction map implemented too

bdf4324: releasing 1.6.2-0.1.rc1

53c7d6c: getting ready to release 1.6.2

9a9f99d: Fix import (#321)

d502bf1: fix typo

### 1.2 Release 1.6.0

In this release:

- Adapt to decisionengine framework migration to setuptools.
- Drop cmake step before running unit tests.
- Enhanced logging.
- Bug fixes.

#### 1.2.1 Issues fixed in this release

275 : (b639a74)

## 1.2.2 Full list of commits since version 1.5.0

[f93831c](#) : enhanced logging and error handling in util directory (#320)  
[798972f](#) : replaced method\_whitelist with allowed\_methods (as former os deprecated) (#319)  
[b8af4b3](#) : Update Jenkins pipeline configuration (#318)  
[b639a74](#) : Handle KeyError exception (#317)  
[4c9e25e](#) : Add about.py for useful names (#310)  
[48e725f](#) : update unittest document  
[36b799e](#) : Configure DE Modules to use DE Framework with setuptools (#316)  
[45fc425](#) : Move to raw flake8 test action (#315)  
[478b0a5](#) : Update gh-actions to best practices (#311)  
[6285f49](#) : Drop obsolete github action (#309)  
[911fef7](#) : Do a limited pin on dependencies (#308)  
[ab096a8](#) : Sync-up .gitignore (#304)  
[cd7e668](#) : Move to raw github actions (and codecov.io) (#298)  
[4a997d4](#) : Remove building of C++ Logic Engine from unit tests

## 1.3 Release 1.5.0

In this release:

- New AWS bill calculator
- Bug fixes

### 1.3.1 Issues fixed in this release

- [271](#) : Change configuration so the bill-calculator-hep-mapsacosta library doesn't write stuff to stdout ([5410c46](#))
- [283](#) : Change DEAccountConstants.py to allow more variables through ([7a65032](#))
- [293](#) : Unit test tests/test\_job\_q.py is failing ([f5006fb](#))
- [263](#) : Python3 encodes hex strings differently, resulting in malformed htcondor classads ([c16f0](#))
- [289](#) : Travis CI fails after decisionengine transitioned to pybind11 ([2bec1da](#))
- [281](#) : New exception in NERSC newt get\_usage ([398dc42](#))



### 1.3.2 Full list of commits since version 1.4.2

5410c46 : new bill calculator package (#303)  
7a65032 : Resolving issue 283 (#301)  
1a2c3e4 : Fix syntax errors and pylint bits (#300)  
976c4a4 : Minor linting of actions (#297)  
c43f311 : Drop redundant action, pep8speaks is already doing this (#296)  
10691e4 : Add support to use custom docker layer as base of the docker image (#295)  
f5006fb : fix for unit test test\_job\_q issue 293 (#294)  
0b588ef : Update workflow per changes at github (#292)  
7841d19 : Updated Jenkins configuration documentation (#291)  
c08ffce : edited release notes  
c16f0ac : Another fix to malformed strings (#288)  
2bec1da : Fix #289 where TravisCI is missing changes from <https://github.com/HEPCloud/decisionengine/commit/ea7ade57ca825afb0afbfa5ebd49820b16786d8b> (#290)  
8647bb2 : Update to current python test matrix (#287)  
234a78e : update release notes  
398dc42 : decisionengine\_modules/NERSC/util: Added more 5xx codes to retry (#285)  
d2df861 : force docker pull when building the docker container to make sure to use an updated base layer (#280)  
9a78d22 : added 1.4.0 release notes  
5d3b7ec : Updating Dockerfile for unittest (#279)  
e16531e : Provide default for util function. (#278)  
02f1738 : set RPM version number to 1.5.0rc and drop py3 suffix (#277)  
664b9fe : Add timeout support for unit test on Jenkins for DE modules repo (#274)  
e392a3c : Added initial unit test for glide\_frontend\_element.py (#269)  
2f0069f : Updated DE modules Jenkins documentation

## 1.4 Release 1.4.2

In this release:

- Bug fixes to 1.4.1 release

### 1.4.1 Issues fixed in this release

- [282](#) : Decision Engine classad encoding problems make multiple exceptions in the factory ([c16f0ac](#))

### 1.4.2 Full list of commits since version 1.4.1

[c16f0ac](#) : Another fix to malformed strings (#288)

## 1.5 Release 1.4.1

In this release:

- Bug fixes to 1.4.0 release

### 1.5.1 Issues fixed in this release

- [281](#) : New exception in NERSC newt get\_usage ([4d1301e](#))

### 1.5.2 Full list of commits since version 1.4.0

[4d1301e](#) : decisionengine\_modules/NERSC/util: Added more 5xx codes to retry (#285)

## 1.6 Release 1.4.0

In this release:

- Works with python3 glideinWMS package
- Uses new AWS bill calculator
- Improved Jenkins CI pipeline

### 1.6.1 Issues fixes in this release

- [59](#) : Replace AWS Billing and GCE Billing modules with revamped bill Calculator code ([71f25ea](#))
- [246](#) : Modify NERSCAllocation main() so that it can be run standalone ([11ce772](#))
- [263](#) : Python3 encodes hex strings differently, resulting in malformed htcondor classads ([13a24e4](#))
- [255](#) : Add option to config of condor/sources/job\_q.py and slots.py ([6f65af4](#))

## 1.6.2 Full list of commits since version 1.3.0

42f62d3 : remove py3 from package name  
6e56c4d : bump version number  
6433dba : Add support to test any branch in Jenkins for DE modules repo (#276)  
60db1b3 : Accommodate breaking changes to framework's configuration system. (#273)  
13a24e4 : Fix263 (#272)  
a866e47 : Correct 'file\_cache is unavailable when using oauth2client >= 4.0.0' (#267)  
2c82064 : Fix requirements.txt for #266 (#268)  
6f65af4 : A more generic solution for the issue 255 (#265)  
df08cb4 : Speed up tests. (#264)  
057e578 : Update Jenkins pipeline to properly test closing PR (#262)  
a28f272 : Adding Jenkins pipeline documentation (#261)  
71f25ea : T59 (#258)  
90b6580 : Updating Jenkins pipeline configuration (#260)  
b4bd759 : dealing with 2 columnsindividually (#257)  
11ce772 : decisionengine\_modules/NERSC: Can run NerscAllocationInfo.py without (#256)  
4a70465 : doc: add release notes

## 1.7 Release 1.3.0

In this release:

- Introduced Jsonnet based configuration system
- Improved error handling

### 1.7.1 Full list of commits since version 1.2.0

e8c8e24 : Update to split a tuple key into two separate keys, this allows python config file to be JSON compliant config file (#254)  
fbb5d28 : add jsonnet dependency in build/scripts/utlis.sh for unit tests (#252)  
95a0a32 : decisionengine\_modules/glideinwms: Added retry to factory queries (#250)  
77f894a : Aws (#249)  
3be5be1 : Set theme jekyll-theme-slate  
a73e7f8 : Doc (#247)  
f8aa96d : Adding editor config file, see <https://editorconfig.org/> (#248)  
1ab1720 : Issue no. 237 (#245)  
5edc8f9 : Issue 227 (#243)  
e6047f2 : GCE channel goes OFFLINE on a timeout (#242)

63e3627 : Aws billing (#241)  
c3626b9 : 1.1.3/fix (#239)  
ae7c5a6 : Jenkins pipeline improvements (#236)  
310befd : T198 (#235)  
a65886d : Fix import as reported in : <https://github.com/HEPCloud/decisionengin...> (#232)  
93711cc : Run coveralls even if tests fail (#229)  
03d763a : Jenkins pipeline improvements (#230)  
f48d30f : Fix/223 (#228)  
c8aa262 : github ticket 199 (#222)  
0323bda : Address : [https://github.com/HEPCloud/decisionengine\\_modules/issues/224](https://github.com/HEPCloud/decisionengine_modules/issues/224) (#226)  
62e4df6 : Add support to run CI on Jenkins (#221)  
5ab1541 : bump master version to 1.2.0 (for now) (#219)  
bc19c65 : decisionengine\_modules/NERSC: Added retry loop for NERSC API Calls (#220)  
41a50de : Sync up pep8speaks and run\_pylint.sh with decisionengine settings (#218)  
db4634f : silence pylint error (#217)  
1b95141 : Fix whitespace around operator error  
746ea38 : ignore W503  
8a8b5f4 : remove unused variable  
a6668bf : fix PEP8 warnings  
13773ee : address pep8 warnings  
6bea4ca : silence pylint error  
f589895 : Pass sort=True parameter to fix future warning (#215)  
a1d0507 : fixing pep8 warning  
a10bd17 : debugging one import error  
ec501ad : make coveralls.io links work  
deab1a7 : T201 (#204)  
69f2645 : Add coveragerc  
6d8a5f5 : decisionengine\_modules/NERSC: Make Nersc API call backward-compatible with old config (#196)  
a7e0af9 : Only run Github Actions for python3.6 (#24)

## 1.8 Release 1.2.0

In this release:

- Switched to python3
- NERSC newt interface adapted to changed NERSC RESTful API
- Improved unit tests
- Added Jenkins CI pipeline

## 1.9 Patches:

30d928b : clone version 1.2.0 of decisionengine

ae7c5a6 : Jenkins pipeline improvements (#236)

310befd : T198 (#235)

a65886d : Fix import as reported in : <https://github.com/HEPCloud/decisionengine...> (#232)

93711cc : Run coveralls even if tests fail (#229)

03d763a : Jenkins pipeline improvements (#230)

f48d30f : Fix/223 (#228)

c8aa262 : github ticket 199 (#222)

0323bda : Address : [https://github.com/HEPCloud/decisionengine\\_modules/issues/224](https://github.com/HEPCloud/decisionengine_modules/issues/224) (#226)

62e4df6 : Add support to run CI on Jenkins (#221)

5ab1541 : bump master version to 1.2.0 (for now) (#219)

bc19c65 : decisionengine\_modules/NERSC: Added retry loop for NERSC API Calls (#220)

41a50de : Sync up pep8speaks and run\_pylint.sh with decisionengine settings (#218)

db4634f : silence pylint error (#217)

1b95141 : Fix whitespace around operator error

746ea38 : ignore W503

8a8b5f4 : remove unused variable

a6668bf : fix PEP8 warnings

13773ee : address pep8 warnings

6bea4ca : silence pylint error

f589895 : Pass sort=True parameter to fix future warning (#215)

a1d0507 : fixing pep8 warning

a10bd17 : debugging one import error

ec501ad : make coveralls.io links work

deab1a7 : T201 (#204)

69f2645 : Add coveragerc

6d8a5f5 : decisionengine\_modules/NERSC: Make Nersc API call backward-compatible with old config (#196)

[a7e0af9](#) : Only run Github Actions for python3.6 (#24)

## DEVELOPER DOCUMENTATION

### 2.1 Decisionengine\_modules

#### 2.1.1 Prerequisites:

```
yum install postgresql11 postgresql11-server  
sudo localedef -v -c -i en_US -f UTF-8 C.UTF-8 # (needed for tests)
```

#### 2.1.2 Test

```
git clone https://github.com/HEPCloud/decisionengine_modules  
./decisionengine_modules/.github/actions/unittest-in-sl7-docker/entrypoint.sh
```





## JENKINS CI PIPELINE

### 3.1 Decisionengine modules CI with Jenkins pipeline

Jenkins dashboard with Decisionengine modules CI results is available [here](#).

A CI build is triggered any time a PR is created/closed or a commit is made to an existing PR. There are also *nightly CI builds* to test a list of predefined branches.

The Jenkins pipeline runs *pep8*, *pylint* and *unit\_tests* test suites alongside the *rpmbuild* stage. The Jenkins dashboard looks like this:

Jenkins > CI > decisionengine\_modules\_pipeline

## Pipeline decisionengine\_modules\_pipeline

DE modules pipeline

**Back to Dashboard**  
**Status**  
**Changes**  
**Build with Parameters**  
**Delete Pipeline**  
**Configure**  
**Full Stage View**  
**GitHub**  
**Job Config History**  
**Rename**  
**GitHub PR**  
**Pipeline Syntax**  
**GitHub PR Polling Log**  
**Set Next Build Number**

**Last Successful Artifacts**

pep8_mail_results	1.31 KB	<a href="#">view</a>
pep8_pep8_merge258.log	0 B	<a href="#">view</a>
pep8_pyLint_merge258.log	0 B	<a href="#">view</a>
pep8_results_merge258.log	44 B	<a href="#">view</a>
pyLint_mail_results	1.31 KB	<a href="#">view</a>
pyLint_pep8_merge258.log	0 B	<a href="#">view</a>
pyLint_pyLint_merge258.log	0 B	<a href="#">view</a>
pyLint_results_merge258.log	44 B	<a href="#">view</a>
pytest.log	10.78 KB	<a href="#">view</a>
rpmbuild.tar	210.00 KB	<a href="#">view</a>

**Recent Changes**

**Build History**

find

- 189#PR#258** Sep 2, 2020 4:14 PM [#258](#)
- 188#PR#258** Sep 2, 2020 3:31 PM [#258](#)
- 187#PR#258** Sep 2, 2020 3:05 PM [#258](#)
- 186#master** Sep 2, 2020 2:31 AM [#258](#)
- 185#PR#260** Sep 1, 2020 1:38 PM [#260](#)
- #184** Sep 1, 2020 2:31 AM
- #183** Aug 31, 2020 2:31 AM
- #182** Aug 30, 2020 2:31 AM
- #181** Aug 29, 2020 2:31 AM
- #180** Aug 28, 2020 2:31 AM
- #179** Aug 27, 2020 3:54 PM [#258](#)
- #178** Aug 27, 2020 3:54 PM [#258](#)
- #177** Aug 27, 2020 3:39 PM [#258](#)
- #176** Aug 27, 2020 3:39 PM [#258](#)

**Stage View**

Average stage times:  
(Average full run time: ~25min)

	Declarative: Checkout SCM	DE modules tests	pep8	pylint	unit_tests	rpmbuild
	903ms	66ms	4min 28s	22min 26s	10min 36s	3min 16s
<b>189#PR#258</b> Sep 02 16:14 No Changes	1s	72ms	14min 10s	34min 22s	22min 31s	14min 52s
<b>188#PR#258</b> Sep 02 15:31 No Changes	539ms	112ms	4min 53s	24min 50s	11min 13s	6min 3s
<b>187#PR#258</b> Sep 02 15:05 No Changes	850ms	79ms	3min 35s	24min 7s	10min 50s <b>failed</b>	1min 19s
<b>186#master</b> Sep 02 02:31 No Changes	1s	58ms	3min 27s	21min 55s	9min 3s	22s
<b>185#PR#260</b> Sep 01 13:38 1 commit	777ms	54ms	7s <b>failed</b>	25s <b>failed</b>	5s <b>failed</b>	4s <b>failed</b>
<b>#184</b> Sep 01 02:31 No Changes	494ms	36ms	2min 49s	22min 16s	8min 53s	21s
<b>#183</b> Aug 31 02:31 No Changes	618ms	78ms	3min 36s	23min 3s	9min 2s	28s

On the bottom left side there is the list of recent CI builds that are named after the PR or the branch tested.

On the bottom right side the dashboard shows for each CI build detailed status for each test suite.

Hovering the mouse over the *status box* for each CI build stage, a tool-tip with a button to access log details shows up.

Next to the build number the symbol  gives access to a menu with the list of artifacts stored for that build. Those artifacts include logs and the tarball with RPMs.

From the panel on the left side it is possible to access the PR on GitHub by clicking on the PR icon that looks like this  **#258**.

On occasion it could be useful to trigger a manual CI build to test a branch on the official DE GitHub repository or on the user fork. For this purpose, on the top left panel the user can click on the  **Build with Parameters** button, and this panel shows up

# Pipeline decisionengine\_modules\_pipeline

This build requires parameters:

DOCKER_IMAGE	<input type="text" value="vitodb/decision-engine-modules-ci;jenkins"/> Docker image name to use. Default is: vitodb/decision-engine-modules-ci;jenkins
DE_MOD_REPO	<input type="text" value="https://github.com/HEPCloud/decisionengine_modules/"/> Decisionengine modules repo. Default is: https://github.com/HEPCloud/decisionengine_modules/
BRANCH	<input type="text" value="master"/> Branch to test. Default is: master
PYTEST_TIMEOUT	<input type="text" value="300"/> Timeout in seconds for unit_tests (it applies to individual unit test) Default is: 300

Build

the user can modify these parameters to customize what code to test with the CI build.

The *DE\_MOD\_REPO* parameter can point to the user fork or to the main repository.

The *BRANCH* parameter can point to the desired branch to test.

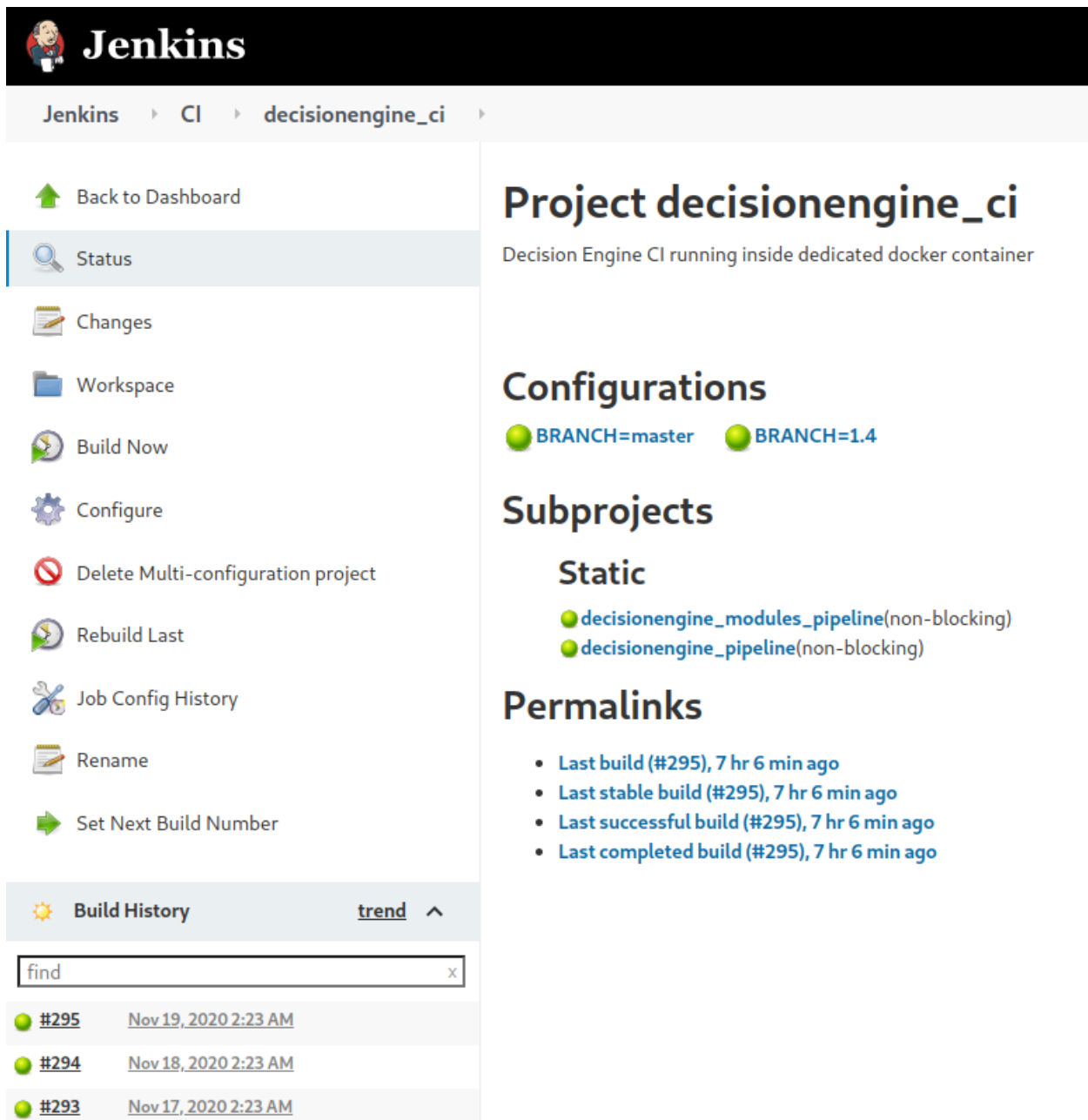
The *PYTEST\_TIMEOUT* parameter is the timeout in seconds for *unit\_tests*.

When ready, by clicking on the *Build* button, the CI build will start.

The [pipeline configuration](#) is part of the decisionengine\_modules repo.

## 3.1.1 Nightly CI build configuration

The nightly CI build for Decisionengine framework uses this [Jenkins project](#) that triggers a CI build using the Jenkins pipeline described above to test a list of predefined branches.



**Jenkins**

Jenkins > CI > decisionengine\_ci

[Back to Dashboard](#)

[Status](#)

[Changes](#)

[Workspace](#)

[Build Now](#)

[Configure](#)

[Delete Multi-configuration project](#)

[Rebuild Last](#)

[Job Config History](#)

[Rename](#)

[Set Next Build Number](#)

**Build History** [trend](#) ^

find

#295	<a href="#">Nov 19, 2020 2:23 AM</a>
#294	<a href="#">Nov 18, 2020 2:23 AM</a>
#293	<a href="#">Nov 17, 2020 2:23 AM</a>

## Project decisionengine\_ci

Decision Engine CI running inside dedicated docker container

## Configurations

BRANCH=master BRANCH=1.4

## Subprojects

### Static

- decisionengine\_modules\_pipeline(non-blocking)
- decisionengine\_pipeline(non-blocking)

## Permalinks

- Last build (#295), 7 hr 6 min ago
- Last stable build (#295), 7 hr 6 min ago
- Last successful build (#295), 7 hr 6 min ago
- Last completed build (#295), 7 hr 6 min ago

Branches to test are defined using the project matrix as shown in the picture below. Each branch in the list (here *master* and *1.4*) spawns an independent CI build.

The screenshot shows the Jenkins Configuration Matrix configuration page. The 'Configuration Matrix' tab is selected. Under the 'User-defined Axis' section, the 'Name' field is set to 'BRANCH' and the 'Values' field is set to 'master 1.4'. There is a red 'X' icon in the top right corner of the axis configuration area and a blue question mark icon next to the values field.

In the *Build* section of the configuration it is set the list of Jenkins subprojects to be triggered, in this case we have *decisionengine\_pipeline* and *decisionengine\_modules\_pipeline*.

The *Parameters* text box is used to override parameters of each Jenkins subproject with a custom value.

In total this Jenkins project triggers 4 CI builds, i.e. 2 branches X 2 Jenkins subprojects.

The screenshot shows the Jenkins Build configuration page. The 'Build' tab is selected. Under the 'Trigger/call builds on other projects' section, the 'Projects to build' field is set to 'decisionengine\_pipeline,decisionengine\_modules\_pipeline'. The 'Block until the triggered projects finish their builds' checkbox is unchecked. Under the 'Predefined parameters' section, the 'Parameters' field is set to 'BRANCH=\${BRANCH}'. There are red 'X' icons in the top right corner of the 'Trigger/call builds on other projects' and 'Predefined parameters' sections, and blue question mark icons next to the 'Projects to build' and 'Parameters' fields.

Finally the *Build Triggers* section is used to setup the schedule for the periodic build, in this case it is scheduled to run at about 2 AM.

Jenkins will choose the actual time depending on the actual load on the system.

General

Advanced Project Options

Source Code Management

Build Triggers

Configuration Matrix

Build Environment

Build

Post-build Actions

### Build Triggers

☐ Trigger builds remotely (e.g., from scripts)

☐ Build after other projects are built

☒ Build periodically

Schedule

H 2 \* \* \*

Would last have run at Wednesday, November 4, 2020 2:23:53 AM CST;  
would next run at Thursday, November 5, 2020 2:23:53 AM CST.

## SOURCE CODE

### 4.1 Welcome to decisionengine\_modules's documentation!

#### 4.1.1 decisionengine\_modules package

##### Subpackages

decisionengine\_modules.AWS package

##### Subpackages

decisionengine\_modules.AWS.publishers package

##### Submodules

decisionengine\_modules.AWS.publishers.AWS\_burn\_rate module

decisionengine\_modules.AWS.publishers.AWS\_figure\_of\_merit module

decisionengine\_modules.AWS.publishers.AWS\_generic\_publisher module

decisionengine\_modules.AWS.publishers.AWS\_price\_performance module

##### Module contents

decisionengine\_modules.AWS.sources package

##### Submodules

decisionengine\_modules.AWS.sources.AWSBurnRateSourceProxy module

decisionengine\_modules.AWS.sources.AWSInstancePerformance module

decisionengine\_modules.AWS.sources.AWSJobLimits module

`decisionengine_modules.AWS.sources.AWSOccupancy` module

`decisionengine_modules.AWS.sources.AWSOccupancyWithSourceProxy` module

`decisionengine_modules.AWS.sources.AWSSpotPrice` module

`decisionengine_modules.AWS.sources.AWSSpotPriceWithSourceProxy` module

`decisionengine_modules.AWS.sources.BillingInfo` module

`decisionengine_modules.AWS.sources.BillingInfoSourceProxy` module

`decisionengine_modules.AWS.sources.DEAccountContants` module

This data is taken from billing and arranged as dictionary of AccountConstants instead of separate class for each billing class. This way is more convenient to use in programmes.

```
class decisionengine_modules.AWS.sources.DEAccountContants.AccountConstants(constants={})
    Bases: object
    info()
```

```
decisionengine_modules.AWS.sources.DEAccountContants.load_constants(constants_file)
    Load constants from file.
```

**Parameters** `constants_file` (file) – configuration file as python dict

**Return type** `AccountConstants`

`decisionengine_modules.AWS.sources.FigureOfMeritSourceProxy` module

`decisionengine_modules.AWS.sources.FinancialParameters` module

`decisionengine_modules.AWS.sources.occupancy_config_sample` module

`decisionengine_modules.AWS.sources.spot_price_config_sample` module

**Module contents**

`decisionengine_modules.AWS.transforms` package

**Submodules**

`decisionengine_modules.AWS.transforms.AwsBurnRate` module

`decisionengine_modules.AWS.transforms.FigureOfMerit` module

**Module contents**



## Submodules

`decisionengine_modules.AWS.LocalCapacity` module

`decisionengine_modules.AWS.pandas_scratch_pad` module

`decisionengine_modules.AWS.pandas_scratch_pad.load_data_frame`(*list\_of\_dicts*)

## Module contents

`decisionengine_modules.GCE` package

## Subpackages

`decisionengine_modules.GCE.publishers` package

## Submodules

`decisionengine_modules.GCE.publishers.GCEBurnRatePublisher` module

`decisionengine_modules.GCE.publishers.GCEFigureOfMerit_publisher` module

`decisionengine_modules.GCE.publishers.GCEPricePerformance_publisher` module

## Module contents

`decisionengine_modules.GCE.sources` package

## Submodules

`decisionengine_modules.GCE.sources.GCEBillingInfo` module

`decisionengine_modules.GCE.sources.GCEBillingInfoSourceProxy` module

`decisionengine_modules.GCE.sources.GCEInstancePerformance` module

`decisionengine_modules.GCE.sources.GCEResourceLimits` module

`decisionengine_modules.GCE.sources.GceBurnRateSourceProxy` module

`decisionengine_modules.GCE.sources.GceFigureOfMeritSourceProxy` module

`decisionengine_modules.GCE.sources.GceOccupancy` module

## Module contents

**decisionengine\_modules.GCE.transforms package**

**Submodules**

**decisionengine\_modules.GCE.transforms.GceBurnRate module**

**decisionengine\_modules.GCE.transforms.GceFigureOfMerit module**

**Module contents**

**Module contents**

**decisionengine\_modules.NERSC package**

**Subpackages**

**decisionengine\_modules.NERSC.publishers package**

**Submodules**

**decisionengine\_modules.NERSC.publishers.NerscFigureOfMerit\_publisher module**

**Module contents**

**decisionengine\_modules.NERSC.sources package**

**Submodules**

**decisionengine\_modules.NERSC.sources.NerscAllocationInfo module**

**decisionengine\_modules.NERSC.sources.NerscAllocationInfoSourceProxy module**

**decisionengine\_modules.NERSC.sources.NerscFigureOfMeritSourceProxy module**

**decisionengine\_modules.NERSC.sources.NerscInstancePerformance module**

**decisionengine\_modules.NERSC.sources.NerscJobInfo module**

**Module contents**

**decisionengine\_modules.NERSC.transforms package**

**Submodules**

**decisionengine\_modules.NERSC.transforms.CompareNerscFactoryJobs module**

**decisionengine\_modules.NERSC.transforms.CompareNerscUserpoolSlots module**

**decisionengine\_modules.NERSC.transforms.NerscFigureOfMerit module**

## Module contents

**decisionengine\_modules.NERSC.util package**

## Submodules

**decisionengine\_modules.NERSC.util.newt module**

Newt API interface

```
class decisionengine_modules.NERSC.util.newt.Newt(password_file, newt_base_url=None,  
                                                  num_retries=0, retry_backoff_factor=0)
```

Bases: object

**\_add\_retries\_to\_session()**

Adds retries to requests Session for requests to NEWT URLs :return: void

**\_login()**

Establishes newt Session :return: void

**get\_queue(system, query=None)**

Returns information about queues controlled by optional query name parameter :param system: string system name (like 'cori', 'edison' ..) :param query: string query (optional) :return: json containing queue information

**get\_status(system=None)**

Returns system status for a given system (if provided) or all systems :param system: string name of the system (optional) :return: json containing system status

**get\_usage(username)**

Returns allocation and usage for the given user using NEWT IRIS API :param username: string username :return: json containing allocation and usage information for a given user

## Module contents

## Module contents

**decisionengine\_modules.graphite package**

## Subpackages

**decisionengine\_modules.graphite.publishers package**

## Submodules

**decisionengine\_modules.graphite.publishers.generic\_publisher module**

## Module contents

## Module contents

decisionengine\_modules.htcondor package

## Subpackages

decisionengine\_modules.htcondor.publishers package

## Submodules

decisionengine\_modules.htcondor.publishers.publisher module

## Module contents

decisionengine\_modules.htcondor.sources package

## Submodules

decisionengine\_modules.htcondor.sources.job\_q module

decisionengine\_modules.htcondor.sources.job\_q\_source\_proxy module

decisionengine\_modules.htcondor.sources.slots module

decisionengine\_modules.htcondor.sources.slots\_source\_proxy module

decisionengine\_modules.htcondor.sources.source module

## Module contents

## Submodules

decisionengine\_modules.htcondor.htcondor\_query module

```
class decisionengine_modules.htcondor.htcondor_query.CondorQ(schedd_name=None,  
                                                             pool_name=None)
```

Bases: *decisionengine\_modules.htcondor.htcondor\_query.CondorQuery*

Class to implement condor\_q

```
_abc_impl = <_abc_data object>
```

```
fetch(constraint=None, format_list=None, condor_config=None)  
Fetch job classads
```

```
class decisionengine_modules.htcondor.htcondor_query.CondorQuery(resource_str, group_attr,  
                                                                    pool_name=None, env=None)
```

Bases: *decisionengine\_modules.htcondor.htcondor\_query.Query*

Fully implemented class for CondorQuery

**\_abc\_impl** = <\_abc\_data object>

**class** decisionengine\_modules.htcondor.htcondor\_query.**CondorStatus**(*subsystem\_name=None*,  
*pool\_name=None*,  
*group\_attr=None*)

Bases: *decisionengine\_modules.htcondor.htcondor\_query.CondorQuery*

Class to implement condor\_status

**\_abc\_impl** = <\_abc\_data object>

**fetch**(*constraint=None*, *format\_list=None*, *condor\_config=None*)

Fetch resource classads and return a list of evaluated classads

**class** decisionengine\_modules.htcondor.htcondor\_query.**Query**

Bases: object

Pure virtual class to have a minimum set of methods defined

**\_abc\_impl** = <\_abc\_data object>

**abstract fetch**(*constraint=None*, *format\_list=None*, *condor\_config=None*)

Fetch the classad attributes specified in the format\_list matching the constraint

**fetch\_stored**(*constraint\_func=None*)

**Parameters constraint\_func** – A boolean function, with only one argument (data el). If None, return all the data.

@rtype: dict @return: Same as fetch(), but limited to constraint\_func(el)==True

**load**(*constraint=None*, *format\_list=None*, *condor\_config=None*)

Fetch the data and store it in self.stored\_data

**exception** decisionengine\_modules.htcondor.htcondor\_query.**QueryError**(*err\_str*)

Bases: RuntimeError

Thrown when there are exceptions using htcondor python bindings or commands

decisionengine\_modules.htcondor.htcondor\_query.**apply\_constraint**(*data*, *constraint\_func*)

Return a subset of data that satisfies constraint\_function If constraint\_func is None, return back entire data

decisionengine\_modules.htcondor.htcondor\_query.**bindings\_friendly\_attrs**(*format\_list*)

Convert the format\_list into attrs that can be used with python bindings Python bindings should take care of the typing

decisionengine\_modules.htcondor.htcondor\_query.**bindings\_friendly\_constraint**(*constraint*)

Convert the constraint to format that can be used with python bindings

decisionengine\_modules.htcondor.htcondor\_query.**eval\_classad\_expr**(*classads*, *format\_list=None*)

Convert the classads into list of classads Guarantees any attribute in format\_list to be added with default value None

decisionengine\_modules.htcondor.htcondor\_query.**list2dict**(*list\_data*, *attr\_name*)

Convert a list to a dictionary and group the results based on attributes specified by attr\_name

decisionengine\_modules.htcondor.htcondor\_query.**resource\_str\_to\_py\_adtype**(*resource\_str*)

Given the resource string return equivalent classad type

decisionengine\_modules.htcondor.htcondor\_query.**split\_collector\_host**(*collector\_host*)

Given the collector\_host string extract primary collector\_host which is the first collector host in the list and sort the secondary collectors. Return a tuple of primary collector and CSV secondary collector.

## Module contents

### decisionengine\_modules.util package

#### Submodules

#### decisionengine\_modules.util.figure\_of\_merit module

Calculate figure of merit

```
decisionengine_modules.util.figure_of_merit.figure_of_merit(performance, running, allowed,  
                                                         idle=None, max_idle=None)
```

#### decisionengine\_modules.util.retry\_function module

```
decisionengine_modules.util.retry_function.retry_on_error(nretries=1, retry_interval=2,  
                                                         backoff=True)
```

```
decisionengine_modules.util.retry_function.retry_wrapper(f, nretries=1, retry_interval=2,  
                                                         backoff=True)
```

Retry on error with parameters of how many times (nretries) and interval in seconds (retry\_interval). If the function to be decorated is an instance method and the values come from a configuration, then this decorator can retrieve these two parameters from instance attributes with the same name. Otherwise, use the default values or pass new values to the decorator.

#### decisionengine\_modules.util.test\_retry\_function module

```
class decisionengine_modules.util.test_retry_function.Dummy(name='Dummy')
```

Bases: object

```
    _func_failure(foo, input2=3)
```

```
    _func_success(foo, input2=3)
```

```
    func_failure(foo, input2=3)
```

```
    func_success(foo, input2=3)
```

```
decisionengine_modules.util.test_retry_function.test_all()
```

#### decisionengine\_modules.util.testutils module

Utils to simplify testing

```
decisionengine_modules.util.testutils.compare_dfs(df1, df2)
```

for some reason df.equals does not work here but if I compare cell by cell it works :type df1: pd.DataFrame :arg df1: data frame instance :type df2: pd.DataFrame :arg df2: data frame instance :rtype: bool - True if equal

```
decisionengine_modules.util.testutils.input_from_file(fname)
```

```
decisionengine_modules.util.testutils.raw_input_from_file(fname)
```

## Module contents

### Submodules

#### decisionengine\_modules.about module

PEP-0396 provides instructions for providing module versions While we are at it, add a few other useful bits

#### decisionengine\_modules.graphite\_client module

```
class decisionengine_modules.graphite_client.Graphite(host='fifemondata.fnal.gov',
                                                       pickle_port=2004)
```

Bases: object

```
send_dict(namespace, data, debug_print=True, send_data=True)
```

send data contained in dictionary as {k: v} to graphite dataset \$namespace.k with current timestamp

```
decisionengine_modules.graphite_client.sanitize_key(key)
```

#### decisionengine\_modules.load\_config module

```
decisionengine_modules.load_config.load(python_file, retries=0, timeout=0)
```

Load constants from file.

**Parameters** `constants_file` (file) – configuration file as python dict

**Return type** object

## Module contents

## 4.2 Indices and tables

- [genindex](#)
- [modindex](#)
- [search](#)





## INDICES AND TABLES

- `genindex`
- `modindex`
- `search`



## PYTHON MODULE INDEX

### d

26

decisionengine\_modules, 27  
decisionengine\_modules.about, 27  
decisionengine\_modules.AWS, 21  
decisionengine\_modules.AWS.pandas\_scratch\_pad,  
21  
decisionengine\_modules.AWS.publishers, 19  
decisionengine\_modules.AWS.sources, 20  
decisionengine\_modules.AWS.sources.DEAccountContants,  
20  
decisionengine\_modules.AWS.sources.occupancy\_config\_sample,  
20  
decisionengine\_modules.AWS.sources.spot\_price\_config\_sample,  
20  
decisionengine\_modules.AWS.transforms, 20  
decisionengine\_modules.GCE, 22  
decisionengine\_modules.GCE.publishers, 21  
decisionengine\_modules.GCE.sources, 21  
decisionengine\_modules.GCE.transforms, 22  
decisionengine\_modules.graphite, 24  
decisionengine\_modules.graphite.publishers,  
24  
decisionengine\_modules.graphite\_client, 27  
decisionengine\_modules.htcondor, 26  
decisionengine\_modules.htcondor.htcondor\_query,  
24  
decisionengine\_modules.htcondor.publishers,  
24  
decisionengine\_modules.htcondor.sources, 24  
decisionengine\_modules.load\_config, 27  
decisionengine\_modules.NERSC, 23  
decisionengine\_modules.NERSC.publishers, 22  
decisionengine\_modules.NERSC.sources, 22  
decisionengine\_modules.NERSC.transforms, 23  
decisionengine\_modules.NERSC.util, 23  
decisionengine\_modules.NERSC.util.newt, 23  
decisionengine\_modules.util, 27  
decisionengine\_modules.util.figure\_of\_merit,  
26  
decisionengine\_modules.util.retry\_function,  
26  
decisionengine\_modules.util.test\_retry\_function,



## Symbols

`_abc_impl` (*decisionengine\_modules.htcondor.htcondor\_query.CondorQ* attribute), 24  
*decisionengine\_modules.htcondor.htcondor\_query.CondorQ* (class in *decisionengine\_modules.htcondor.htcondor\_query*), 24  
`_abc_impl` (*decisionengine\_modules.htcondor.htcondor\_query.CondorQuery* attribute), 25  
*decisionengine\_modules.htcondor.htcondor\_query.CondorQuery* (class in *decisionengine\_modules.htcondor.htcondor\_query*), 24  
`_abc_impl` (*decisionengine\_modules.htcondor.htcondor\_query.CondorStatus* attribute), 25  
*decisionengine\_modules.htcondor.htcondor\_query.CondorStatus* (class in *decisionengine\_modules.htcondor.htcondor\_query*), 24  
`_abc_impl` (*decisionengine\_modules.htcondor.htcondor\_query.Query* attribute), 25  
*decisionengine\_modules.htcondor.htcondor\_query.Query* (class in *decisionengine\_modules.htcondor.htcondor\_query*), 25

`_add_retries_to_session` (*decisionengine\_modules.NERSC.util.newt.Newt* method), 23  
`_func_failure` (*decisionengine\_modules.util.test\_retry\_function.Dummy* method), 26  
`_func_success` (*decisionengine\_modules.util.test\_retry\_function.Dummy* method), 26  
`_login` (*decisionengine\_modules.NERSC.util.newt.Newt* method), 23

## A

`AccountConstants` (class in *decisionengine\_modules.AWS.sources.DEAccountConstants*), 20  
`apply_constraint` (in module *decisionengine\_modules.htcondor.htcondor\_query*), 25

## B

`bindings_friendly_attrs` (in module *decisionengine\_modules.htcondor.htcondor\_query*), 25  
`bindings_friendly_constraint` (in module *decisionengine\_modules.htcondor.htcondor\_query*), 25

## C

`compare_dfs` (in module *decisionengine\_modules.util.testutils*), 26

## D

*decisionengine\_modules* module, 27  
*decisionengine\_modules.about* module, 27  
*decisionengine\_modules.AWS* module, 21  
*decisionengine\_modules.AWS.pandas\_scratch\_pad* module, 21  
*decisionengine\_modules.AWS.publishers* module, 19  
*decisionengine\_modules.AWS.sources* module, 20  
*decisionengine\_modules.AWS.sources.DEAccountConstants* module, 20  
*decisionengine\_modules.AWS.sources.occupancy\_config\_sample* module, 20  
*decisionengine\_modules.AWS.sources.spot\_price\_config\_sample* module, 20  
*decisionengine\_modules.AWS.transforms* module, 20  
*decisionengine\_modules.GCE* module, 22  
*decisionengine\_modules.GCE.publishers* module, 21  
*decisionengine\_modules.GCE.sources* module, 21  
*decisionengine\_modules.GCE.transforms* module, 22  
*decisionengine\_modules.graphite* module, 24  
*decisionengine\_modules.graphite.publishers*

module, 24  
 decisionengine\_modules.graphite\_client  
   module, 27  
 decisionengine\_modules.htcondor  
   module, 26  
 decisionengine\_modules.htcondor.htcondor\_queryfunc\_success()  
   module, 24  
 decisionengine\_modules.htcondor.publishers  
   module, 24  
 decisionengine\_modules.htcondor.sources  
   module, 24  
 decisionengine\_modules.load\_config  
   module, 27  
 decisionengine\_modules.NERSC  
   module, 23  
 decisionengine\_modules.NERSC.publishers  
   module, 22  
 decisionengine\_modules.NERSC.sources  
   module, 22  
 decisionengine\_modules.NERSC.transforms  
   module, 23  
 decisionengine\_modules.NERSC.util  
   module, 23  
 decisionengine\_modules.NERSC.util.newt  
   module, 23  
 decisionengine\_modules.util  
   module, 27  
 decisionengine\_modules.util.figure\_of\_merit  
   module, 26  
 decisionengine\_modules.util.retry\_function  
   module, 26  
 decisionengine\_modules.util.test\_retry\_function  
   module, 26  
 decisionengine\_modules.util.testutils  
   module, 26  
 Dummy (class in decision-  
   engine\_modules.util.test\_retry\_function),  
   26

## E

eval\_classad\_expr() (in module decisio-  
 engine\_modules.htcondor.htcondor\_query),  
 25

## F

fetch() (decisionengine\_modules.htcondor.htcondor\_query.ComplexQuery  
   method), 24  
 fetch() (decisionengine\_modules.htcondor.htcondor\_query.ComplexQueryStatus  
   method), 25  
 fetch() (decisionengine\_modules.htcondor.htcondor\_query.Query  
   method), 25  
 fetch\_stored() (decision-  
   engine\_modules.htcondor.htcondor\_query.Query  
   method), 25

figure\_of\_merit() (in module decisio-  
 engine\_modules.util.figure\_of\_merit), 26  
 func\_failure() (decision-  
   engine\_modules.util.test\_retry\_function.Dummy  
   method), 26  
 (decision-  
   engine\_modules.util.test\_retry\_function.Dummy  
   method), 26

## G

get\_queue() (decision-  
 engine\_modules.NERSC.util.newt.Newt  
   method), 23  
 get\_status() (decision-  
   engine\_modules.NERSC.util.newt.Newt  
   method), 23  
 get\_usage() (decision-  
   engine\_modules.NERSC.util.newt.Newt  
   method), 23  
 Graphite (class in decisio-  
   engine\_modules.graphite\_client), 27

## I

info() (decisionengine\_modules.AWS.sources.DEAccountContants.Account  
   method), 20  
 input\_from\_file() (in module decisio-  
   engine\_modules.util.testutils), 26

## L

list2dict() (in module decisio-  
 engine\_modules.htcondor.htcondor\_query),  
 25  
 load() (decisionengine\_modules.htcondor.htcondor\_query.Query  
   method), 25  
 load() (in module decisio-  
   engine\_modules.load\_config), 27  
 load\_constants() (in module decisio-  
   engine\_modules.AWS.sources.DEAccountContants),  
 20  
 load\_data\_frame() (in module decisio-  
   engine\_modules.AWS.pandas\_scratch\_pad),  
 21

## M

module  
 decisionengine\_modules, 27  
 decisionengine\_modules.about, 27  
 decisionengine\_modules.AWS, 21  
 decisionengine\_modules.AWS.pandas\_scratch\_pad,  
 21  
 decisionengine\_modules.AWS.publishers, 19  
 decisionengine\_modules.AWS.sources, 20  
 decisionengine\_modules.AWS.sources.DEAccountContants,  
 20

decisionengine\_modules.AWS.sources.occupancy\_configs.sample\_adtype() (in module decisio-  
 20 engine\_modules.htcondor.htcondor\_query),  
 decisionengine\_modules.AWS.sources.spot\_price\_configs.sample, (in module decisio-  
 20 engine\_modules.util.retry\_function), 26  
 decisionengine\_modules.AWS.transforms, 20  
 decisionengine\_modules.GCE, 22  
 decisionengine\_modules.GCE.publishers, 21  
 decisionengine\_modules.GCE.sources, 21  
 decisionengine\_modules.GCE.transforms, 22  
 decisionengine\_modules.graphite, 24  
 decisionengine\_modules.graphite.publishers, (in module decisio-  
 24 engine\_modules.util.retry\_function), 26  
 decisionengine\_modules.graphite\_client, (in module decisio-  
 27 engine\_modules.graphite\_client.Graphite  
 method), 27  
 decisionengine\_modules.htcondor, 26  
 decisionengine\_modules.htcondor.htcondor\_query, (in module decisio-  
 24 engine\_modules.htcondor.htcondor\_query),  
 25  
 decisionengine\_modules.htcondor.publishers, 25  
 24  
 decisionengine\_modules.htcondor.sources, (in module decisio-  
 24 engine\_modules.util.test\_retry\_function),  
 26  
 decisionengine\_modules.load\_config, 27  
 decisionengine\_modules.NERSC, 23  
 decisionengine\_modules.NERSC.publishers, 22  
 decisionengine\_modules.NERSC.sources, 22  
 decisionengine\_modules.NERSC.transforms, 23  
 decisionengine\_modules.NERSC.util, 23  
 decisionengine\_modules.NERSC.util.newt, 23  
 decisionengine\_modules.util, 27  
 decisionengine\_modules.util.figure\_of\_merit, 26  
 decisionengine\_modules.util.retry\_function, 26  
 decisionengine\_modules.util.test\_retry\_function, 26  
 decisionengine\_modules.util.testutils, 26

## N

Newt (class in decision-  
 engine\_modules.NERSC.util.newt), 23

## Q

Query (class in decisio-  
 engine\_modules.htcondor.htcondor\_query),  
 25

QueryError, 25

## R

raw\_input\_from\_file() (in module decisio-  
 engine\_modules.util.testutils), 26