

# Table of Contents


<b>CRAB3 releases (since October 2014)</b> .....	<b>1</b>
CRAB v3.3.1707 (released on July 11, 2017).....	1
Improvements, enhancements, changes.....	1
Bug fixes.....	1
CRAB v3.3.1706 (released on June 13, 2017).....	1
Improvements, enhancements, changes.....	1
Bug fixes.....	1
CRAB v3.3.1705 (released on May 2, 2017).....	1
Improvements, enhancements, changes.....	1
Bug fixes.....	2
CRAB v3.3.1704 (released on April 4, 2017).....	2
Improvements, enhancements, changes.....	2
Bug fixes.....	2
CRAB v3.3.1703 (released on March 8, 2017).....	2
Improvements, enhancements, changes.....	2
Bug fixes.....	2
CRAB v3.3.1702 (released on February 7, 2017).....	3
Improvements, enhancements, changes.....	3
Bug fixes.....	3
CRAB v3.3.1611 (released on November 14, 2016).....	3
Bug fixes.....	3
CRAB v3.3.1608 (released on August 24, 2016).....	3
New features.....	3
Bug fixes.....	3
CRAB v3.3.1607 (released on July 13, 2016).....	3
New features.....	4
Bug fixes.....	4
CRAB v3.3.1606 (released on June 7, 2016).....	4
New features.....	4
Improvements, enhancements, changes.....	4
Bug fixes.....	4
CRAB v3.3.1605 (released on May 10, 2016).....	4
New features.....	4
Improvements, enhancements, changes.....	4
Bug fixes.....	5
CRAB v3.3.1604 (released on April 12, 2016).....	5
New features.....	5
Improvements, enhancements, changes.....	5
Bug fixes.....	5
CRAB v3.3.1603 (released on March 9, 2016).....	5
New features.....	5
Improvements, enhancements, changes.....	5
Bug fixes.....	6
CRAB v3.3.1602 (released on February 2, 2016).....	6
New features.....	6
Improvements, enhancements, changes.....	6
Bug fixes.....	6
CRAB v3.3.1512 (released on December 3, 2015).....	7
New features.....	7
Improvements, enhancements, changes.....	7
Bug fixes.....	7
CRAB v3.3.1511 (released on November 5, 2015).....	7
Improvements, enhancements, changes.....	7
Bug fixes.....	7

# Table of Contents

## CRAB3 releases (since October 2014)

CRAB v3.3.1510 (released on October 6, 2015).....	7
New features.....	7
Improvements, enhancements, changes.....	8
Bug fixes.....	8
CRAB v3.3.1509 (released on September 1, 2015).....	8
Bug fixes update (released on September 29, 2015).....	8
New features.....	8
Improvements, enhancements, changes.....	9
Bug fixes.....	9
CRAB v3.3.1507 (released on July 7, 2015).....	9
New features.....	10
Improvements, enhancements, changes.....	10
Bug fixes.....	10
CRAB v3.3.1506 (released on June 9, 2015).....	10
New features.....	10
Improvements, enhancements, changes.....	10
Bug fixes.....	10
CRAB v3.3.16 (released on May 5, 2015).....	11
New features.....	11
Improvements, enhancements, changes.....	11
Bug fixes.....	11
CRAB v3.3.15 (released on April 8, 2015).....	11
Bug fixes update (released on April 16, 2015).....	11
New features.....	11
Improvements, enhancements, changes.....	12
Bug fixes.....	12
CRAB v3.3.14 (released on March 3, 2015).....	12
Bug fixes update (released on March 17, 2015).....	12
Improvements, enhancements, changes.....	12
Bug fixes.....	13
CRAB v3.3.13 (released on February 4, 2015).....	13
New features.....	13
Improvements, enhancements, changes.....	13
Bug fixes.....	13
CRAB v3.3.12 (released on December 9, 2014).....	14
Bug fixes update (released on December 19, 2014).....	14
New features.....	15
Improvements, enhancements, changes.....	15
Bug fixes.....	15
CRAB v3.3.11 (released on November 18, 2014).....	16
Bug fixes update (released on November 26, 2014).....	16
New features.....	16
Improvements, enhancements, changes.....	16
Bug fixes.....	17
CRAB v3.3.10 (released on October 5, 2014).....	17
New features.....	17
Improvements, enhancements.....	17
Bug fixes.....	17
Other Topics.....	18

# CRAB3 releases (since October 2014)

Complete:  [Go to SWGuideCrab](#)

## CRAB v3.3.1707 (released on July 11, 2017)

### Improvements, enhancements, changes

- Removed \*old commands (`statusold`, `resubmitold`, `reportold`, `getoutputold`, `getlogold`)
- Events per lumi information of users' output files now being propagated to DBS during publication.
- Improvements for managing the task's lifetime length.

### Bug fixes

- Introduced a check for output file LFNs (maximum length not exceeded and correct format) before the submission of the task.

## CRAB v3.3.1706 (released on June 13, 2017)

### Improvements, enhancements, changes

- Changes to how task status is reported in the CRABClient API.
  - ◆ Intended to make the `status` key in the return status dictionary backwards compatible with the old `crab status` logic.
  - ◆ Documentation can be found here [https://twiki.cern.ch/twiki/bin/view/CMSPublic/CRABClientLibraryAPI#crab\\_status\\_API\\_description](https://twiki.cern.ch/twiki/bin/view/CMSPublic/CRABClientLibraryAPI#crab_status_API_description)
- Introduced a limit to the amount of lumis (100k) that each input dataset block has.
  - ◆ If a dataset contains blocks bigger than 100k lumis, submission will be refused by the CRABServer.

### Bug fixes

- Output publication fixes
- CRABClient code refactoring for downloading files used when constructing overall status. Also addresses the issue when `crab status` would not work outside of CERN.

## CRAB v3.3.1705 (released on May 2, 2017)

### Improvements, enhancements, changes

- Making the new command implementations the default.
  - ◆ The commands that were previously called "`=status2, resubmit2...=`" are now called:
    - ◇ `crab status`
    - ◇ `crab resubmit`
    - ◇ `crab report`
    - ◇ `crab getoutput`
    - ◇ `crab getlog`
  - ◆ The old implementations are still available under these names:
    - ◇ `crab statusold`
    - ◇ `crab resubmitold`
    - ◇ `crab reportold`
    - ◇ `crab getoutputold`

◇ `crab getlogold`

## Bug fixes

- Fix for publication when using the new Oracle implementation and direct stageout.

## CRAB v3.3.1704 (released on April 4, 2017)

### Improvements, enhancements, changes

- `crab status2` now has a `--jobids` option.
  - ◆ Can be used together with the `--long` and / or `===sort` options. Specific jobids or a range of jobids can be passed to only display the `--long` or `--sort` results for those jobs.
- Fail the task and upload a warning if task's webdir upload failed during the bootstrapping step on the grid scheduler.
- Show a better error message when the `Data.userInputFiles` parameter is incorrect.
- Check that the size of the sandbox is under 100MB before uploading, not after.

## Bug fixes

- Renew proxies for completed tasks, previously one could not force resubmission of a completed task after the proxy had expired.
- CRABMon webdir link fix, previously would not work from outside CERN.
- Don't send additional input files list to the server.

## CRAB v3.3.1703 (released on March 8, 2017)

### Improvements, enhancements, changes

- Prototype client commands `status2`, `report2`, `getoutput2`, `getlog2`, `resubmit2`.
  - ◆ The new implementation of the commands should reduce the load on the CRABServer backends. Specifically, `crab status2 --long` should be much faster and should be used instead of the old `crab status --long`.
  - ◆ They rely on a `task_process` that runs alongside each task on the grid scheduler. The process iteratively computes and caches the status of each task, looking only at new information with which the status needs to be updated. This happens every 5 minutes for each task independently, meaning that the information displayed by `status2` (on which the other commands also depend on) may be up to 5 minutes old. However, we do not foresee this to be a problem. Otherwise, one should expect these commands to behave exactly the same as the regular commands.
- New tab in CRAB Monitoring UI with status of transfers for tasks which use OracleASO.
- Print traceback when catching `StageOutMgr` initialisation exceptions.
- Report number of requested cores to Dashboard.

## Bug fixes

- Fix file metadata upload in the `PostJob`.
- Fix issue when the list of additional input files is too large for the database column.

## CRAB v3.3.1702 (released on February 7, 2017)

### Improvements, enhancements, changes

- CRAB now takes e-groups into account when deciding if the user is part of the local site users.
- Allow to purge tasks in SUBMITFAILED fixes,
- Add link to CRABMon in crab status,
- Print used memory in message to user when a job is killed due to memory usage,
- Error summary generation code optimization,
- Metrics based scheduler picker function.

### Bug fixes

- Site blacklist doesn't update on resubmission,
- Task process / status2 fixes,
- Make number of cores in crab config and number of threads in pset consistent,
- Use 'python', not 'python2.6' in dag\_bootstrap\* scripts,
- CRABMon fixes.

## CRAB v3.3.1611 (released on November 14, 2016)

### Bug fixes

- Minor fix for python 2.6 compatibility
- Make Task Worker sleep between queries if CRABServer cannot be reached
- CRABMon minor improvements
- Log working directory file sizes when a job is killed
- Report, resubmit hotfixes

## CRAB v3.3.1608 (released on August 24, 2016)

### New features

- Allow task submission for partial datasets
  - ◆ A message will be displayed notifying the user of this behavior in case only a partial dataset is found. This is mostly relevant for new datasets that are in the process of being distributed and therefore are still incomplete.
- Ignore global blacklist option
  - ◆ If the user is sure that the site(s) is able to run jobs successfully but it is blacklisted for some reason, it is now possible to get around the blacklist by adding a config parameter  
`config.Site.ignoreGlobalBlacklist = True.`

### Bug fixes

- Fix error when the schedd list is empty after retries
- Truncate warning messages

## CRAB v3.3.1607 (released on July 13, 2016)

## New features

### Bug fixes

- Protect against missing LFN in input file.
- Separate Ops monitoring files into another tarball
- Clarify `--priority` option, correct description for `--checksum` option
- Fix resubmit returning 'None' error to the client
- Fix concurrency issue when creating log directory
- Log unexpected errors in subprocesses
- Pick up a new schedd only after retries for current one are exhausted
- `cfipython` folder should now be included if `sendPythonFolder` option is enabled.

## CRAB v3.3.1606 (released on June 7, 2016)

### New features

A new feature intended for HammerCloud operators is available:

- Slow job release for HammerCloud tests.
  - ◆ Instead of submitting multiple tasks per hour for each site, a single task can be used for a longer period of time to continuously release jobs into the site, reducing the overhead load. Documentation on this feature is available [here](#).

### Improvements, enhancements, changes

- All tasks will now be removed after 30 days. Resubmission will not be allowed after 23 days.

### Bug fixes

- Remove some code for backwards compatibility with old tasks and CRAB Clients
- Report local/fallback/direct stageout to dashboard
- Remove unnecessary debug printout in `RenewRemoteProxies`.
- Protect the `TaskWorker` from main loop crashes
- Adjust logic in kill after TM refactoring.
- Protect `PostJob` from empty `defer_num.*.*.txt` file

## CRAB v3.3.1605 (released on May 10, 2016)

### New features

### Improvements, enhancements, changes

- Operators can now kill user tasks, which is useful in rare cases. A reason for the kill will also be provided, which is displayed when doing `crab status`.
- Better input parameter validation for MC generation.
- The problem with files being mysteriously deleted after the transfer should occur much less frequently now.
  - ◆ FTS timeout has been tuned and now compatible with ASO timeout

## Bug fixes

- Save and load startup environment to file. Because of some environment conflicts in certain sites, problems have been observed during transfers, which should now be fixed.
- Improved retrievable error handling
- Do not inject the task to the scheduler if it was not possible to set its status to QUEUED
- Retry pycurl error 35
- Add clusterid to the database, shows if the task was successfully submitted to the scheduler
- Improve error message when the stage out fails to the selected destination

## CRAB v3.3.1604 (released on April 12, 2016)

### New features

### Improvements, enhancements, changes

- New CRABServer state machine. Makes the system more robust in case of cmsweb unavailability and extreme use cases where tasks get stuck in the QUEUED state:
  - ◆ The only visible change for the users is that the KILL and RESUBMIT statuses are now reported as NEW.
- Lightweight CRABClient
  - ◆ The client now works out of the box inside the CMSSW environment. There is now an `init-light.sh` script in `cvmfs` that only does an `export PYTHONPATH = $PYTHONPATH:/path/to/lib` and `export PATH = $PATH:/path/to/bin`, not polluting the environment. You can source it using the `source /cvmfs/cms.cern.ch/crab3/crab_light.sh` command after `cmsenv`.
- CondorStatusService enabled
  - ◆ The worker node now reports job runtime, memory, cpu usage and other details to HTCondor. This helps with monitoring and tracking down issues for the developers.
- Long job exit code now reported to HTCondor using chirp

## Bug fixes

- Added a disk usage check and removal for jobs in case they exceed the allowed limit on the worker node.
- Overwrite CRAB3.zip in `dag_bootstrap_startup.sh`
- TaskWorker tarball build script fixes to adjust to the new WMCore version
- Always use pycurl in client and server, WMCore default is httplib.
- Allow DryRun to use `config.JobType.scriptArgs` parameter.
- `crab uploadlog` now works even if the submission to the CRABServer failed.

## CRAB v3.3.1603 (released on March 9, 2016)

### New features

### Improvements, enhancements, changes

- `crab report` command rewritten to be more detailed, and to improve how the input for recovery tasks are created.
  - ◆ Now it provides information about the lumis in the input dataset at submission time, lumis that the task had to analyse (obtained after applying lumimask/runrange filters), lumis processed by `finished` jobs, information about the lumis in the output dataset.

- ◆ The biggest improvement compared to the previous version is more flexibility in the creation of the "recovery" lumis. You can now create a recovery task out of `failed` jobs without having to wait for the task to be complete. See details about the new parameter `--recovery`
- ◆ Full documentation can be found [here](#).
- ◆ NB the `--dbs` parameter is deprecated now
- For horizontal scalability reasons the CRAB server has been improved and now is able to deal with more than one AsyncStageout backends.
- For Site Administrators: allow using VOMS group for deciding when to send "local pilot" in gWms. The feature is documented [here](#)

## Bug fixes

- Make crab client work with old CMSSW version by cleaning the environment when delegating the proxy, issue described [here](#).

## CRAB v3.3.1602 (released on February 2, 2016)

### New features

### Improvements, enhancements, changes

- Upon request from the submission infrastructure group we will stop using T1s when the PrivateMC plugin is used. This is done to prevent user's jobs overtaking other more important jobs (e.g.: RelVal jobs) now that we deployed some changes regarding how pilot jobs are sent to sites.
- Optimized speed of `crabreport` (credits to Sébastien Brochet who provided the fix)
  - ◆ `crab report` was taking ages when running over a big tasks
- Add submission/resubmission failure message to `crab submit --wait`
- Introduced recoverable/non-recoverable transfer failures in ASO.
  - ◆ Transfers will not blindly retried until failure, but a distinction between failures that can be recovered with a simple transfer resubmission, and transfers that needs a whole job resubmission has been introduced.
- Improvements in the handling of the `cancelled` state have been introduced in ASO.
  - ◆ Will result in a reduced impact of timeouts that in the past have caused undesired deletion of files under some circumstances.

## Bug fixes

- Fixed a bug that was repeatedly causing submission failures to schedulers
  - ◆ Primary reason of the `Failed to connect to schedd. error`. The error is still possible if the scheduler has problems, but a "fake error" caused by a bug in CRAB3 will not happen anymore.
- Fix issue in uploading secondary input files metadata which was causing (some) jobs to fail in the post-processing step when using parents or a secondary input dataset.
- Do not include contributions from secondary input files in the `crab report` output.
  - ◆ Until now, secondary input files (e.g. parents) were included when reporting the number of the files that have been processed and the number of events that have been read.
- Warnings in `crab status` should not appear repeated.
- Avoid showing duplicated warnings in the `crab status`
- Fixed the `= Fix 'ConfigSection' object has no attribute 'outputPrimaryDataset'= error`
- Reasons for high memory usage in crab server backends were unraveled. The `filemetadata` (internal) server API was changed and will not use many big dictionaries anymore but strings. So far so good, no need for server restarts anymore.
- `Stage1` changes to get the code closer to python3 compatibility have been introduced.



- Migration to python2.7 and new gcc493 have been completed

## CRAB v3.3.1512 (released on December 3, 2015)

### New features

- Implement user resubmission of (all) failed publications in the `crab resubmit` command via a new option `--publication`.

### Improvements, enhancements, changes

- Set an upper limit of 100 characters for the request name (`General.requestName`) length.

### Bug fixes

- Fix bug in `crab status` that was causing information from other tasks to leak into the status report.
- Fix bug in the validation of the taskname length.

## CRAB v3.3.1511 (released on November 5, 2015)

### Improvements, enhancements, changes

- The CRAB configuration parameter `Data.secondaryDataset` has been renamed to `Data.secondaryInputDataset`.
- The CRAB configuration parameter `Data.primaryDataset` has been renamed to `Data.outputPrimaryDataset`.
- The CRAB configuration parameter `Data.publishDataName` has been renamed to `Data.outputDatasetTag`.
- Allow resubmission of tasks in `SUBMITFAILED` status.
- In data discovery, use locations from PhEDEx only (DBS only) for input datasets in global (phys0X) DBS instance.
- Improvements in `crab checkusername` command (use bash commands to parse the username from SiteDB output).
- Add protection to handle the case of corrupted "node\_state" file (can happen when schedd spool area is full).
- Always show dashboard URL in `crab status` if the task has been submitted to a schedd.
- Add links in schedd user web directory for two relevant files.
- Do not submit tasks for "connection refused" error.

### Bug fixes

- Fix validation of primary dataset names.

## CRAB v3.3.1510 (released on October 6, 2015)

### New features

- Allow publication in multiple output datasets. (This feature was actually enabled in September release.)
- Allow using a lumi-mask and/or run-range with FileBased splitting.

- New client function `getLumiListInValidFiles` available for users in the `UserUtilities` module. This function allows to retrieve the runs/lumis in (the valid files of) a dataset published in DBS. See `getLumiListInValidFiles`.

## Improvements, enhancements, changes

- Improve files removal from the temp area.
- Use proxied URLs in `crab getlog --short`. This allows to use this command from outside the CERN firewall.
- Add the possibility to check files checksums in `crab get*` commands.
- Add new task statuses (`SUBMITFAILED`, `RESUBMITFAILED` and `KILLFAILED`) for signalling that a failure occurred in executing one of the three actions: submit, resubmit and kill.
  - ◆ This also fixes the following two issues:
    1. Users not being able to use `crab resubmit --jobids` again when a resubmission has just failed (`crab resubmit` without the `--jobids` option would still work).
    2. `crab status` not showing the jobs status for tasks in `FAILED` status after a failure in executing one of the three actions mentioned above.
- Identical sandboxes will have the same ID in the crabcache. It will allow to save some disk space from your quota in the crabcache, especially if you used the crab library.
- Implement correction for split lumis in `EventAwareLumiBased` splitting
- Several improvements to `get(logout)` which did not print error messages in the logfile (and on the screen)

## Bug fixes

- Fix a bug in `LumiBased` splittings when `Nth` and `Nth-1` jobs share the same lumi during lumi correction.
- Fix data management in the `/store/group` area.

## CRAB v3.3.1509 (released on September 1, 2015)

### Bug fixes update (released on September 29, 2015)

- Fix bug that was causing to skip processing any part of a dataset if some blocks were still open in `PhEDEx`, or more in general, if CRAB could not find the locations (host sites) for some blocks.
- Fix environment related issue affecting jobs running in `T1_US_FNAL`.

## New features

- Add support for using a secondary dataset.
- Add `--status` option in `crab tasks` command to allow filtering tasks by status.
- Add a summary with publication failure reasons in `crab status` output.
- Add DBS client to CRAB3 client.
  - ◆ There is no need anymore to source the CRAB2 environment when using DBS client scripts to for example change the status of a dataset in DBS.
- Add `--skip-estimates` option to `crab submit --dryrun`.
  - ◆ This option allows to skip the estimates of job duration and memory consumption. In this case, the dry run will show only the jobs splitting information.
- Add basic validation of the `CMSSW` parameter-set configuration: `process.source` must be defined.

## Improvements, enhancements, changes

- Deprecated GlideMon web service for monitoring CRAB user jobs. Removed pointer to GlideMon URL in `crab status` output.
- Do direct stageout from the worker node to the destination storage when the execution site is the same as the destination storage site.
- Use proxied URLs for the job and post-job log files linked from dashboard.
  - ◆ This allows them to be accessed from outside CERN.
- Implement post-jobs execution requeueing.
  - ◆ This avoids post-jobs waiting to start execution due do the limit on the number of concurrent running post-jobs. With the new post-job execution requeueing feature, a job that has finished executing in the worker node will immediately start running the post-job step, and each post-job will execute again (if necessary) every 30 minutes.
  - ◆ In most cases, the request for transferring files from the worker node to the destination storage is made from the worker node and the post-job only monitors the status of the transfers. Having post-jobs queued waiting to start execution was creating cases where a job stays in `transferring` status for (many) hours even after the corresponding files have finished transferring. The post-jobs requeueing avoids these cases. However, since there is a defer time interval of 30 minutes, a job may still stay in `transferring` (or `transferred`) status after transfers have finished and until the deferred post-job executes again (this is for up to a maximum of 30 minutes).
- Consider as a fatal errors (i.e. don't automatically resubmit the job) the following stageout failures: a) failure in creating a directory in the destination storage, b) not enough space in the destination storage, and c) SE is down.
- Check if user has write permission into the destination storage before injecting the transfers requests.
- Improvement of many error messages. A never ending story...
- Remove jobs that are idle for more than a week.
- Enhance `crab checkwrite` command to also check it can create and delete a directory in the destination storage.
- When passing arguments to the `crab submit` command, assume the first argument is always the CRAB configuration file name.
- Add Grid scheduler name to the `crab status` output.

## Bug fixes

- Fix `crab kill`.
- Fix bug in reporting killed jobs to dashboard.
- Fix environment issue when running `scriptExe`: unpack the user input sandbox in the `$CMSSW_BASE` directory.
- Fix the retrieval of valid files from DBS for datasets that are not in status `VALID` or `PRODUCTION`.
  - ◆ This fixes the issue referred to by the red comment in the July release notes below.
- Set the job status into `transferred` only after all files-to-transfer in the job have been successfully transferred.
  - ◆ Previously, the job status was being set to `transferred` when at least one of the files-to-transfer in the job has been successfully transferred.
- Fix a bug in the calculation of duplicate lumis reported by `crab report`.
- Fix the behaviour of `crab status --json`.
- Fix bug that was causing to have duplicate lines in the `crab.log` file when an unhandled exception occurred.

## CRAB v3.3.1507 (released on July 7, 2015)

## New features

- Allow publishing using a groupname instead of the username in the dataset name.
  - ◆ A new boolean parameter `Data.publishWithGroupName` was introduced in the CRAB configuration. The parameter defaults to `False`. If set to `True` and the output LFN directory path (`Data.outLFNDirBase`) starts with `/store/group/<groupname>`, the `groupname` is used in the publication dataset name.
- Add new functions `setConsoleLogLevel` and `getConsoleLogLevel` and a constant `LOGLEVEL_MUTE` to allow users to control the CRAB logging level to the console. Add new function `getLoggers` for users to retrieve the CRAB loggers. These functions are intended to be used in the context of the CRAB client library API. Documentation in `CRAB3UserFunctions`.

## Improvements, enhancements, changes

- A new boolean parameter `Data.allowNonValidInputDataset` was introduced in the CRAB configuration to allow CRAB to run over (the valid files of) an input dataset that is not in `VALID` status in DBS.
  - ◆ The only purpose of this parameter is for users to explicitly acknowledge that they know the dataset is not in `VALID` status. Up to CRAB v3.3.1506 this was not required.
  - ◆ It was noticed however that, due to a recent change in `WMCORE`, CRAB is not able to run over datasets that are not in status `VALID` or `PRODUCTION` even if the files itself are in valid status. We plan to fix this in CRAB v3.3.1509.

## Bug fixes

- Report correct exit code to dashboard for jobs killed in the worker node because of excessive memory/cpu/disk usage.
- Report correct number of events processed when a job runs on multiple input files.
  - ◆ Up to CRAB v3.3.1506, the report included only the number of events processed in the last input file.
- Provide a script to overcome the CRAB3 and CMSSW environment conflicts. See this FAQ.

## CRAB v3.3.1506 (released on June 9, 2015)

### New features

- Implement the use of `Data.totalUnits` for `EventAwareLumiBased` splitting. In this case, `Data.totalUnits` refers to events (same as `Data.unitsPerJob`).

### Improvements, enhancements, changes

- Add `--wait` option in `crab resubmit` command. This option has similar behaviour as in the `crab submit` command: wait until the (re)submission request has been submitted to the grid.
- Add `--force` option in `crab resubmit` command. This option is necessary for resubmitting successfully finished jobs.
- Build the CRAB3 site blacklist automatically using input information from the Site Status Board. Until CRAB v3.3.16 the CRAB3 site blacklist had to be updated by hand by a CRAB3 operator.

### Bug fixes

- Don't submit a task if the lumi-mask and run-range (given in the `Data.lumiMask` and `Data.runRange` CRAB configuration parameters) are incompatible (i.e. their intersection is null). Until CRAB v3.3.16

the full dataset is analyzed in that case.

- In ASO, fix the "parents migration problem" when publishing a dataset that has more than one parent dataset.
  - ◆ When publishing a user dataset in DBS<sub>phys03</sub> database, the parent datasets must be migrated to the DBS<sub>phys03</sub> database (if they were not registered there already). Until now ASO was migrating only the primary input parent dataset, and so if the dataset to be published had more than one parent dataset (e.g. when using a secondary input dataset for pile-up), publication was failing.

## CRAB v3.3.16 (released on May 5, 2015)

### New features

- A new CRAB configuration parameter `JobType.disableAutomaticOutputCollection` has been introduced. Documented in CRAB configuration parameters.

### Improvements, enhancements, changes

- The CRAB configuration parameter `allowNonProductionCMSSW` has been renamed to `allowUndistributedCMSSW`.
- The CRAB configuration parameter `outLFN` has been renamed to `outLFNDirBase`.
- Cap job runtime (wall-time) request at 2800 minutes (46h:40m), i.e. set to 2800 if user asks for more. Also give a warning if a task requests more than 2800 minutes of job runtime or more than 2500 MB of memory.
- Format of the error summary changed to not print the error messages; to see the error messages use the new option `--verboseErrors`. If there are different error messages for a given exit code, only the 3 most frequent error messages will be shown. Make sure you use `--verboseErrors` and try to figure out the problem yourself before asking for support.

### Bug fixes

- Fix `crab resubmit` command options, which were not working. Options have also been renamed. See `crab resubmit`.
- Fix `crab status` to report the correct exit codes especially when jobs terminate due to exceeding wall-time or memory limit. This should make this FAQ obsolete.
- Fix the sorting of job ids by exit code in `crab status --sort=exitcode` (by the way, there is no need anymore to use `--long` with `--sort`).
- Log files for all jobs should be accessible again in Dashboard.

## CRAB v3.3.15 (released on April 8, 2015)

### Bug fixes update (released on April 16, 2015)

- Fix bug that was causing errors in the import of (some) CMSSW parameter-set configurations in PrivateMC workflows.

### New features

- Introduced `--dryrun` option in the client for estimating splitting parameters. We encourage you to try to use `crab submit --dryrun crabconfig.py` when submitting the task, review what happened, and then do `crab proceed` if everything is ok.

- Added `crab getlog --short` to get the truncated version of the log saved on the schedd. Can be used even if `General.transferLogs = False` (the default) in the CRAB configuration file. It works only at CERN, because of firewall issues that are being solved.
- Give priority to tasks submitted first.
- Automatically detect if a MC workflow has an LHE input (no need to set the `config.generator` parameter anymore).

## Improvements, enhancements, changes

- Improved (re)selection of the scheduler in case of failures at task submission time: certain types of failure that are related to one specific scheduler failure will be overcome by selecting a new scheduler.
- Always use long exit code for dashboard reports (8bit exit code was used in some instances)

## Bug fixes

- Fix the way we count processed files in crab report

## CRAB v3.3.14 (released on March 3, 2015)

### Bug fixes update (released on March 17, 2015)

- Fix bug in stage out wrapper that was causing, under certain rare circumstances, to not transfer the output file and still show the job as successfully completed.
- Add a (sub-)priority parameter to the jobs so that jobs from tasks with equivalent priority are run respecting the submission time order.
- Fix exit code report to dashboard (report the "long" exit code instead of `exit_code % 256`).

## Improvements, enhancements, changes

- Change task name to not include anymore the scheduler name.
  - ◆ The new format of the task name is `YYMMDD_hhmmss:<username>_crab_<requestName>`. The previous format was `YYMMDD_hhmmss_<scheduler>:<username>_crab_<requestName>`. Users that are relying on the previous format in their private scripts should take a note of this change.
- `crab uploadlog` now works even if the `.requestcache` file is not present in the CRAB project directory.
- The CRAB configuration parameter `Data.userInputFile` has been renamed to `Data.userInputFiles`, and it doesn't take a text file as input anymore, but a python list of the input files. See CRAB configuration parameters for more details.
- Show job exit code in `crab status --long`.
- When staging out into a `/store/group/` area, don't require anymore the `username` to be part of the path. Thus, the new allowed locations for files stage out are `/store/user/<username>` and `/store/group/<groupname>`, where `<username>` is as always the username of the CERN primary account as registered in SiteDB (`/store/local/<something>` is also allowed if publication is turned off).
- Improvement of many error messages.
- Improvements in the CRAB Server backend.
- Generate audio bell with `crab submit --wait`.

## Bug fixes

- Fix a bug in `crab checkusername` that was causing error to retrieve username from SiteDB under certain environments.
- Fix `crab status` to show the status of the latest job retries when a task is in `QUEUED` state after the user did a resubmission.

## CRAB v3.3.13 (released on February 4, 2015)

### New features

- `multicrab` by means of the CRAB library:  
<https://twiki.cern.ch/twiki/bin/view/CMSPublic/CRABClientLibraryAPI>
- WMCORE "event aware lumi splitting" is now available in CRAB3.

### Improvements, enhancements, changes

- Don't even do local stageout into a temporary storage for (log and/or output) files if they were not requested to be transferred to a permanent storage. This also means that a user will not be able to retrieve the (log and/or output) files from the temporary storage if he/she didn't requested the files to be transferred to a permanent storage.
- Check whether the `Data.publishDataName` CRAB configuration parameter (and also the `Data.primaryDataset` parameter in case of a MC generation task), which define (part of) the output dataset name for publication in DBS, satisfy the publication dataset name rules from DBS. If not, don't submit the task and return an appropriate message to the user.
- Restrict the LFN where the `crab checkwrite` command can check users' write permission to the allowed LFNs for stageout.
  - ◆ When CRAB stages out files to a given storage site, it can do it only into an allowed LFN according to the CMS LFN namespace policy. (CRAB checks at submission time if the LFN for stageout provided by the user is an allowed one, and if it is not refuses to submit the jobs and returns an appropriate error message to the user). On the other hand, the `crab checkwrite` command was able to check user's write permission into any given LFN. Since this new release, the `crab checkwrite` command will reject checking write permissions into an LFN that is not allowed for stageout.
- Rename the client module `client_utilities` to `ClientUtilities`. Rename the client module `client_exceptions` to `ClientExceptions`.
- Move all the client functions that are exposed to users (`getUsernameFromSiteDB`, `getFileFromURL`, `config`) to a new module `UserUtilities`.
  - ◆ Users should import those functions with `from CRABClient.UserUtilities import <function>` instead of `from CRABClient.client_utilities import <function>`.
- Rename `getBasicConfig` function to `config`.
- New client function `getFileFromURL` available for users in the `UserUtilities` module. As its name suggests, this function allows to retrieve a file from a URL. See `getFileFromURL`.
- It is possible to type `--` and `tab` in a bash shell to see and autocomplete the `crab` commands.
- In `crab status`, show the output dataset(s) name(s) even if the publication information is not available.

### Bug fixes

- Fix the counting of the total number of files to publish in the `crab status` output.
- Fix the counting of the number of processed files in the `crab report` output (it was showing "0 files have been processed").

## CRAB v3.3.12 (released on December 9, 2014)

### Bug fixes update (released on December 19, 2014)

This is a patch release (version 3.3.12.patch1) of the CRAB Server backend component and the CRAB client that fixes some bugs listed below.

- Implement work-around fix in CRAB for oracle bug triggered when uploading a long file metadata to the CRAB server.
  - ◆ The oracle error was/is happening in jobs that analyze a large amount of lumis. The symptom is that all jobs are shown as `failed` with exit code 0 and failure at the post-processing (in `crab status`, Dashboard and GlideMon). In this case jobs are automatically retried (up to 2 times), but fail at every retry. On the other hand, output files are actually present in the user's destination storage. Furthermore, in all the cases we have investigated, the file metadata was actually present in the CRAB server database, meaning that publication (which needs the file metadata of the outputs) was able to run. Therefore, we ended up telling to users that they can ignore the failure if the publication was ok (or of course, in case the outputs were not intended to be published, if they were present in the destination storage). Eventually we were suggesting to use a finer job splitting in the CRAB configuration file.
  - ◆ The fix implemented in CRAB consists in: whenever get back an error uploading a file metadata, check if the file metadata is available in the CRAB server or not. If it is, ignore the error and proceed normally.
- Fix publication for output files that are staged out directly from the worker node to the destination storage (i.e. output files not staged out via the ASO component).
  - ◆ Direct stageout is the fallback method used by CRAB3 in case the initial stageout to the temporary storage in the running site fails. (This was happening last week for example with jobs running at CERN.) There was a bug preventing the publication for those output files. `crab status` was showing the jobs in `finished` status (which is correct) and publication in `idle` status all the time (it is ok to have the publication in `idle` status for even some hours, but shouldn't be so for more than 8 hours).
- The `crab report` command was enhanced to also work when the task produces output files of a type other than EDM.
  - ◆ Before, this command was only working if the task produced an output file of EDM type (i.e. produced an output file via `PoolOutputModule`). Now it also works for tasks that, within `cmsRun`, produce an output file via `TFileService` or (in principle) via any other module. The most important output from the `crab report` command are the luminosity summary json files (`lumiSummary.json` and `missingLumiSummary.json`). Before, the file `lumiSummary.json` was produced by parsing from the framework job report produced by `cmsRun` the list of luminosity sections reported in the "output" section of the report, and that section existed (if we are not wrong) only for output files produced via `PoolOutputModule`. Now, the file `lumiSummary.json` is produced by parsing the list of luminosity sections reported in the "input" section of the framework job report, which should be valid for all output files produced in the workflow and should be the same list as in the "output" section.
  - ◆ Warning: The information shown by `crab report` about the number of files processed is not correct when the task doesn't produce an EDM output file. `crab report` will show "0 files have been processed". On the other hand, the information about number of events processed and written should be always correct.
  - ◆ Warning: Tasks (that produce an output file of type other than EDM) submitted before this deployment, will not benefit from this patch. Users will need to submit a new task if for them it's critical to get the correct `lumiSummary.json` file.
  - ◆ Feedback in case of inconsistencies in `crab report` is very much appreciated.
- Fix bug in `crab status` command showing `Output dataset:`  
`/FakeDataset/fakefile-FakePublish-5b6a581e4ddd41b130711a045d5fecb9/USER` when an output file of type other than EDM is produced.



- Don't clean the environment when running `scriptExe`.
  - ◆ The problem here was that, by cleaning the environment, some critical information were lost, like the `X509_USER_PROXY` which defines the user's proxy file location. As a consequence, jobs were not able to access for example remote input files via `xrootd`.
  - ◆ We kindly ask users to report back to us in case this fix has introduced some other unexpected problem.

## New features

- A new parameter named `JobType.eventsPerLumi` has been introduced in the CRAB configuration file.
  - ◆ This parameter allows users to specify how many events should a luminosity section contain when generating MC events. The previous behaviour was to create one luminosity section per job.
- A new parameter named `Data.useParent` has been introduced in the CRAB configuration file.
  - ◆ This parameter has the same behaviour as `use_parent` in CRAB2.

## Improvements, enhancements, changes

- Don't inject transfer requests to ASO until the local stage out for all the files in the job were successful.
- When staging out the files from the worker node, use the same stage out policy (either local stage out or remote stage out) for all the files.
  - ◆ In practice: 1) try local stage out first and, if it succeeds for all files, inject transfer requests to ASO; 2) if local stage out fails for at least one file, clean the local temporary storage (keep the logs archive so that the user can retrieve it with `crab getlog`) and try direct stage out for all files to the permanent storage in the site specified by the user; 3) if the remote stage out fails for at least one file, clean the destination directory in the remote permanent storage and the job is failed (will be automatically retried).
- Move the upload of the logs archive file metadata from the post-job to the stage out wrapper in the worker node, and do the upload even if the job fails.
  - ◆ This allows the user to retrieve the logs (with `crab getlog`) even if the job fails.
- Implement the new CMS global LFN namespace policy.
  - ◆ Users can only write into `/store/user/<username>/` or `/store/group/<groupname>[/<subgroupname>]*/<username>/`, where `username` is the user's username registered in SiteDB (which in turn is the user's CERN primary account username). If publication is off, the user can also write into `/store/local/<dirname>[/<subdirname>]*`.
- The command line option `-t/--task` has been renamed to `-d/--dir`.
- Report input number of events analyzed by terminal jobs. See <https://hypernews.cern.ch/HyperNews/CMS/get/computing-tools/272.html>.
- Improved reporting of jobs removed in the worker nodes because of memory/wallclock/disk limits are hit:
  - ◆ the exit code and remove reason will also be shown during `crab status now`
- Improved the algorithm used for checking that datasets have no lumis split across files
  - ◆ it is faster now, the CRAB3 server backend should not get stuck processing some user's task

## Bug fixes

- Don't retry a job in case the job wrapper exit code is 80000 (or 128).
- Update locations of log/output files when the jobs that generates them are retried. Less failures in `crab gelog` and `crab getoutput` commands should be observed now.

## CRAB v3.3.11 (released on November 18, 2014)

### Bug fixes update (released on November 26, 2014)

This is a transparent release (compatible with current CRAB client version 3.3.11) of the ASO and CRAB Server backend components that fixes some minor bugs listed below.

- Save the luminosity sections that a job should analyze into a json file.
  - ◆ Before, this list of luminosity sections were directly saved in the condor input file, with a limit of 128000 characters, and indirectly limiting how many luminosity sections a job could analyze. Now jobs can accept any list of luminosity sections to run on.
- Change (relax) the check on the CMSSW release name.
  - ◆ Before CRAB was accepting only `CMSSW(_[0-9X]){3}(_[a-zA-Z0-9_-]+)?`. Now it accepts `CMSSW[a-zA-Z0-9_-]*`.
- Save only the first 1K lines and the last 3K lines of the cmsRun stdout to the job log.
- Fix ASO submission problem to FTS.
- Improved error messages for some corner cases (e.g.: datasets only on tape)

### New features

- MC generation from LHE files.
- The `crab status` command shows an error summary for failed jobs.
- Improved report of jobs in transferring state (a new state called `transferred` has been added).

### Improvements, enhancements, changes

- Error messages from the server backend and the server frontend have been improved to better tell user what is going on and what they have to do.
- Some of the configuration parameters have been renamed.
  - ◆ E.g. `Data.saveLogs` was renamed to `Data.transferLogs`, `Data.transferOutput` was renamed to `Data.transferOutputs`, `JobType.outlfn` was renamed to `JobType.outLFN`, `Data.dbsUrl` was renamed to `Data.inputDBS`, `Data.publishDbsUrl` was renamed to `Data.publishDBS`.
- Don't exit the PostJob until all the file transfers are in a terminal state.
- Improve screen output of `getoutput` and `getlog` commands when `--dump` option is used.
  - ◆ Print not only the PFN of the files that would be retrieved, but also the LFN, and organize the output by jobs.
- Renamed option `--skip-proxy` to `--proxy` in all client commands.
- Renamed `checkHNname` command to `checkusername`.
- Renamed client function `getHyperNewsName` to `getUsernameFromSiteDB`.
  - ◆ And the renamed function `getUsernameFromSiteDB` has been improved: a) avoid error when scram is not available, b) check existence of valid proxy, c) improve messages to user.
- Use the new client function `getUsernameFromSiteDB` in the `checkwrite` command when it needs to retrieve the username.
  - ◆ When the `--lfn` option is not specified, the `checkwrite` command assumes that the user wants to check its write permission in `/store/user/<username>`. In that case the command retrieves the username from SiteDB. Until now it was doing it via the client class `CredentialInteractions` (which in turn uses `SiteDBJSON` from `WMCORE`). Now it uses the client function `getUsernameFromSiteDB`.
- Disable the default automatic kill of tasks.
  - ◆ CRAB was killing tasks for which at least 10% of the jobs had a fatal error.
- Disabled CMSSW MemoryCheck warnings  
(<https://hypernews.cern.ch/HyperNews/CMS/get/computing-tools/150.html>)

- Always return the output dataset name, even after the temporary data have been deleted from the database.

## Bug fixes

- People should not get the "This CRAB server is not configured to publish; no publication status is available" error anymore. See <https://hypernews.cern.ch/HyperNews/CMS/get/computing-tools/218/1/1/1.html>
- Removed blocking calls in the condor python bingings that were causing occasional DatabaseUnavailable problems.
  - ◆ Under some circumstances crab commands were immediately returning DatabaseUnavailable errors. This should not happen again, please report it if you see this happening.
- Don't make a transfer request for a file that is already in transfer.
  - ◆ This situation was sometimes happening when jobs were being automatically retried by CRAB and the PostJob left the transfers go on even if it decided to retry the job. This bug was causing the these jobs were reported as successfully finished, but the corresponding (output or log) file was actually missing in the permanent storage.
- Fix job-retry-count indexing problem in `postjob.<jobid>.<retrycount>.txt` files.
  - ◆ The bug was causing that PostJob log files were missing for jobs from manual resubmission. The user was seeing "Post-job is currently queued." even if the post-processing step was finished.
- Strip `file:` from the output file names in `jobReport.json`.
  - ◆ When a user was adding the protocol prefix `file:` in the output file names in the pset, this was passed to the job report and cmscp was unable to match the local file names in the job report.

## CRAB v3.3.10 (released on October 5, 2014)

### New features

- Possibility to run a user script instead of plain `cmsRun`.
  - ◆ Two new parameters have been added to support this functionality, namely `JobType.scriptExe` and `JobType.scriptArgs`. A short documentation can be found in the twiki CRAB3 configuration file.

### Improvements, enhancements

- The `crab getlog` and `crab getoutput` commands are now able to retrieve files from the temporary storage when the corresponding transfers to a permanent storage were disabled (to remember: the transfer of the logs is disabled by default).
- Check if a dataset is on tape when looking for the locations of the dataset.
- Automatically upload the CRAB log file to the CRABCache for unhandled exceptions.
- Various improvements in the `crab2cfgT0crab3py` script, so to better translate a CRAB2 configuration file into a CRAB3 configuration file.
- Do not print colors if the standard output if it is redirected to a file.

### Bug fixes

- Add file size to job report for the additional output files.
  - ◆ The file size (for the additional output files) in the job report was so far set to 0. This information is used for example by the `crab getoutput` command to determine what is the "expected" size of the output files. When a user executes `crab getoutput`, the command first checks if the corresponding output files already exist in the local destination directory (the

`results` subdirectory in the task CRAB project directory). If they already exist, the command compares the actual size of the files with the "expected" size, and if the sizes differ it retrieves the file again from the corresponding storage. Having the file size set to 0 in the job report had the obvious undesired behaviour that `crab getoutput` was always retrieving all the additional output files again whenever the command was executed.

- Pass `CMSRunAnalysis` exit code as argument to `cmscp` (the wrapper the takes care of local transfers). Use it as the exit code in `cmscp` if not 0.
  - ◆ The exit code reported by Dashboard in case of job failure should be now consistent with the exit code reported in the job log file.
- Strip the `file:` string from file names where the user specified `file:` in the CMSSW parameter-set or the CRAB3 configuration files.
  - ◆ This bug was causing that files specified by the user with a `file:` prefix were not found by CRAB.

## Other Topics

- Local Resource Provisioning: the access to local resources such as sites with no grid access (e.g.: CAF at CERN, LPC at Fermilab), opportunistic sites, etc, is not possible yet with CRAB3. The CRAB3 and the Submission Infrastructure teams are evaluating different proposal of how to better integrate this use case in the CRAB3 client/server architecture (<https://twiki.cern.ch/twiki/bin/view/CMSPublic/CompOpsLocalSubmission>). The idea is to include this type of submission into the central system, improving the CRAB2 direct submission model. A timeline for implementing the use case has not been set yet.

-- AndresTanasijczuk - 15 Oct 2014

---

This topic: CMSPublic > CRAB3Releases

Topic revision: r109 - 2017-07-11 - EmilisAntanasRupeika



Copyright &© 2008-2020 by the contributing authors. All material on this collaboration platform is the property of the contributing authors.

Ideas, requests, problems regarding TWiki? Send feedback