Table of Contents

CRAB configuration file

Complete:	Go to SWGuideCrab

CRAB configuration file

For convenience, we suggest to place the CRAB configuration file in the same directory as the CMSSW parameter-set file to be used by CRAB.

The expected default name of the CRAB configuration file is `crabConfig.py`, but of course one can give it any name (respecting always the filename extension `.py` and not adding other dots in the filename), as long as one specifies the name when required (e.g. when issuing the CRAB submission command).

In CRAB3 the configuration file is in Python language. It consists of creating a `Configuration` object imported from the `WMCore` library:

```
import CRABClient
from WMCore.Configuration import Configuration
config = Configuration()
```

Once the `Configuration` object is created, it is possible to add new sections to it with corresponding parameters. This is done using the following syntax:

```
config.section_("<section-name>")
config.<section-name>.<parameter-name> = <parameter-value>
```

CRAB configuration sections

The table below shows what are the sections currently available for CRAB configuration.

Section	Description
`General`	In this section, the user specifies generic parameters about the request (e.g. request name).
`JobType`	This section aims to contain all the parameters of the user job type and related configurables (e.g. CMSSW parameter-set configuration file, additional input files, etc.).
`Data`	This section contains all the parameters related to the data to be analyzed, including the splitting parameters.
`Site`	Grid site parameters are defined in this section, including the stage out information (e.g. stage out destination site, white/black lists, etc.).
`User`	This section is dedicated to all the information relative to the user (e.g. voms information).
`Debug`	For experts use only.

Predefined CRAB configuration file with empty skeleton

To simplify life a bit, CRAB provides a function config that returns a `Configuration` object with pre-defined sections. The function is in the `CRABClient.UserUtilities` module. Users can import and use the function in their CRAB configuration file:

```
from CRABClient.UserUtilities import config
config = config()
```

which, from the point of view of the `Configuration` instance, is equivalent to:

```
from WMCore.Configuration import Configuration
config = Configuration()
```

```
config.section_("General")
config.section_("JobType")
config.section_("Data")
config.section_("Site")
config.section_("User")
config.section_("Debug")
```

CRAB configuration parameters

The table below provides a list of all the available CRAB configuration parameters (organized by sections), including a short description. Mandatory parameters are marked with two stars (**). Other important parameters are marked with one star (*).

Parameter	Type	Description
Section `General`		
`requestName (*)`	string	A name the user gives to it's request/task. In particular, it is used t to create a project directory (named `crab_<requestName>`) where corresponding to this particular task will be stored. Defaults to `<time-stamp>`, where the time stamp is of the form `<YYYYMMDD>_` and corresponds to the submission time. The maximum allowed l 100 characters, according to the formati in RX_TASKNAME. T submission will fail with "Incorrect 'workflow' parameter" if othe characters are used.
`workArea (*)`	string	The area (full or relative path) where to create the CRAB project If the area doesn't exist, CRAB will try to create it using the `mkdi` command. Defaults to the current working directory.
`transferOutputs (*)`	boolean	Whether or not to transfer the output files to the storage site. If se `False`, the output files are discarded and the user can not recover Defaults to `True`.
`transferLogs (*)`	boolean	Whether or not to copy the jobs log files to the storage site. If set the log files are discarded and the user can not recover them. Noti however that a short version of the log files containing the first 10 and the last 3000 lines are still available through the monitoring v Defaults to `False`.
`failureLimit`	integer	The number of jobs that may fail permanently before the entire ta cancelled. Disabled by default. **Note:** a very dangerous parameter expert use, do not touch it unless you are sure of what you are doi
`instance (**)`	string	The CRAB server instance where to submit the task. For users ple `'prod'`.
`activity`	string	The activity name used when reporting to Dashboard. For experts
Section `JobType`		
`pluginName (**)`	string	Specifies if this task is running an analysis (`'Analysis'`) on an ex dataset or is running MC event generation (`'PrivateMC'`).
`psetName (*)`	string	The name of the CMSSW parameter-set configuration file that sh run via `cmsRun`. Defaults to `'pset.py'`.
`generator`	string	This parameter should be set to `'lhe'` when running MC generati LHE files. Automatically set if an `LHESource` is present in the parameter-set.
`pyCfgParams`	list of strings	List of parameters to pass to the CMSSW parameter-set configura as explained here. For example, if set to `['myOption','param1=value1','param2=value2']`, then the jo execute `cmsRun JobType.psetName myOption param1=value1`

		`param2=value2`. **NOTE1:** No blanks allowed in `'param-value'`. double dashes break things, i.e. this works `JobType.pyCfgParams=["arg1=1","arg2=2"]` , but this fails `JobType.pyCfgParams=["--arg1=1" "--arg2=2"]`.
`inputFiles`	list of strings	List of private input files (and/or directories) needed by the jobs. be added to the input sandbox. The input sandbox can not exceed The input sandbox is shipped with each job. The input files will b in the working directory where the users' application (e.g. cmsRu launched regardless of a possible path indicated in this parameter the file name at right of last / is relevant). Directories are tarred a subtree structure is preserved. Please check the FAQ for more det how these files are handled.
`disableAutomaticOutputCollection`	boolean	Whether to disable or not the automatic recognition of output file produced by PoolOutputModule or TFileService in the CMSSW parameter-set configuration. If set to `True`, it becomes the user's responsibility to specify in the `JobType.outputFiles` parameter output files that need to be collected. Defaults to `False`.
`outputFiles`	list of strings	List of output files that need to be collected. If `disableAutomaticOutputCollection = False` (the default), ou produced by PoolOutputModule or TFileService in the CMSSW parameter-set configuration are automatically recognized by CRA don't need to be included in this parameter.
`eventsPerLumi`	integer	When `JobType.pluginName = 'PrivateMC'`, this parameter spe many events should a luminosity section contain. Note that every with a fresh luminosity section, which may lead to unevenly sized luminosity sections if `Data.unitsPerJob` is not a multiple of this parameter. Defaults to 100.
`allowUndistributedCMSSW`	boolean	Whether to allow or not using a CMSSW release possibly not ava sites. Defaults to `False`.
`maxMemoryMB`	integer	Maximum amount of memory (in MB) a job is allowed to use. De 2000.
`maxJobRuntimeMin`	integer	The maximum runtime (in minutes) per job. Jobs running longer amount of time will be removed. Defaults to 1315 (21 hours 55 m see the note about `maxJobRuntimeMin` below. Not compatible wit `Automatic` splitting.
`numCores`	integer	Number of requested cores per job. Defaults to 1. If you increase to run multi-threaded cmsRun, you may need to increase `maxMemo` well. In the CMSSW parameter-set configuration you may require also number of streams to be larger than one per thread, which affects memory consumption too.
`priority`	integer	Task priority among the user's own tasks. Higher priority tasks w processed before lower priority. Two tasks of equal priority will h jobs start in an undefined order. The first five jobs in a task are gi priority boost of 10. Defaults to 10.
`scriptExe`	string	A user script that should be run on the worker node instead of the cmsRun. It is up to the user to setup the script properly to run on node enviroment. CRAB guarantees that the CMSSW environme (e.g. scram is in the path) and that the modified CMSSW paramet configuration file will be placed in the working directory with na `PSet.py`. The user must ensure that a properly named framework file will be written; this can be done e.g. by calling cmsRun withi script as `cmsRun -j FrameworkJobReport.xml -p PSet.py`. Th itself will be added automatically to the input sandbox. Output fil

		produced by PoolOutputModule or TFileService in the CMSSW parameter-set configuration file will be automatically collected (C will look in the framework job report). The user needs to specify output files to be collected in the `JobType.outputFiles` paramet CRAB3AdvancedTopic#Running_a_user_script_with_CRAB for information.
`scriptArgs`	list of strings	Additional arguments (in the form `param=value`) to be passed to specified in the `JobType.scriptExe` parameter. The first argume to the script is always the job number
`sendPythonFolder`	boolean	Determine if the 'python' folder in the CMSSW release ($CMSSW_BASE/python) is included in the sandbox or not. Def `False`.
`sendExternalFolder`	boolean	Determine if the 'external' folder in the CMSSW release ($CMSSW_BASE/external) is included in the sandbox or not. Se https://hypernews.cern.ch/HyperNews/CMS/get/computing-tools/ . Defaults to `False`.
`externalPluginFile`	string	Name of a plug-in provided by the user and which should be run the standard CRAB plug-in `Analysis` or `PrivateMC`. Can not be s together with `pluginName`; is either one or the other. Not supporte
Section `Data`		
`inputDataset (*)`	string	When running an analysis over a dataset registered in DBS, this p specifies the name of the dataset. The dataset can be an official C dataset or a dataset produced by a user.
`allowNonValidInputDataset`	boolean	Allow CRAB to run over (the valid files of) the input dataset give `Data.inputDataset` even if its status in DBS is not VALID. Def `False`.
`outputPrimaryDataset (*)`	string	When running an analysis over private input files or running MC generation, this parameter specifies the primary dataset name that used in the LFN of the output/log files and in the publication data (see Data handling in CRAB).
`inputDBS (*)`	string	The URL of the DBS reader instance where the input dataset is pu The URL is of the form `'https://cmsweb.cern.ch/dbs/prod/<instance>/DBSReader` `instance` can be `global`, `phys01`, `phys02` or `phys03`. The default instance. The aliases `global`, `phys01`, `phys02` and `phys03` in plac whole URLs are also supported (and indeed recommended to avo For datasets that are not of `USER` tier, CRAB only allows to read t `global` DBS.
`splitting (*)`	string	Mode to use to split the task in jobs. When `JobType.pluginName` `'Analysis'`, the splitting mode can either be `'Automatic'` (the d please read the dedicated FAQ), `'FileBased'`, `'LumiBased'`, or `'EventAwareLumiBased'` (for Data the recommended mode is `'Automatic'` or `'LumiBased'`). For `'EventAwareLumiBased'`, C split the task by luminosity sections, where each job will contain number of luminosity sections such that the number of events ana each job is roughly `unitsPerJob`. When `JobType.pluginName =` `'PrivateMC'`, the splitting mode can only be `'EventBased'`.
`unitsPerJob (*)`	integer	Mandatory when `Data.splitting` is not `'Automatic'`, suggests impose) how many units (i.e. files, luminosity sections or events depending on the splitting mode - see the note about `Data.split` below) to include in each job. When `Data.splitting = 'Autom` represents the jobs target runtime in minutes and its minimum all value is 180 (i.e. 3 hours).

totalUnits (*)	integer	Mandatory when JobType.pluginName = 'PrivateMC', in whic parameter tells how many events to generate in total. When JobType.pluginName = 'Analysis', this parameter tells how m (when Data.splitting = 'FileBased'), luminosity sections (w Data.splitting = 'LumiBased') or events (when Data.splitt 'EventAwareLumiBased' or Data.splitting = 'Automatic' - note about "Data.splitting" below) to analyze (after applying the l and/or run range filters).
useParent	boolean	Adds corresponding parent dataset in DBS as secondary input sou Allows to gain access to more data tiers than present in the curren This will not check for parent dataset availability; jobs may fail w errors or due to missing dataset access. Defaults to False.
secondaryInputDataset	string	An extension of the Data.useParent parameter. Allows to specif grandparent dataset in DBS (same instance as the primary dataset secondary input source. CRAB will internally set this dataset as th and will set Data.useParent = True. Therefore, Data.usePare Data.secondaryInputDataset can not be used together *a priori*
lumiMask (*)	string	A lumi-mask to apply to the input dataset before analysis. Can eit URL address or the path to a JSON file on disk. Default to an em (no lumi-sections filter).
runRange (*)	string	The runs and/or run ranges to process (e.g. '193093-193999,198050,199564'). It can be used together with lumi-mask. Defaults to an empty string (no run filter).
outLFNDirBase (*)	string	The first part of the LFN of the output files (see Data handling in Accepted values are /store/user/<username>[/<subdir>*] (th / after <username> can not be omitted if a subdir is not given) a /store/group/<groupname>[/<subgroupname>*] (and /store/local/<dir>[/<subdir>*] if Data.publication = Fa Defaults to /store/user/<username>/. CRAB creates the outLF path on the storage site if needed, do not create it yourself otherw stage-out may fail due to permissions inconsistency.
publication (*)	boolean	Whether to publish or not the EDM output files (i.e. output files p by PoolOutputModule) in DBS. Notice that for publication to be the corresponding output files have to be transferred to the perma storage element. Defaults to True.
publishDBS (*)	string	The URL of the DBS writer instance where to publish. The URL form 'https://cmsweb.cern.ch/dbs/prod/<instance>/DBSWr where instance can so far only be phys03, and therefore it is set default, so the user doesn't have to specify this parameter. The ali in place of the whole URL is also supported.
outputDatasetTag (*)	string	A custom string used in both, the LFN of the output files (even if Data.publication = False) and the publication dataset name (Data.publication = True) (see Data handling in CRAB).
publishWithGroupName	boolean	If Data.outLFNDirBase starts with /store/group/<groupname> groupname instead of the username in the publication dataset nam Data handling in CRAB). This feature allows different users runn same workflow to publish in the same dataset. Defaults to False.
ignoreLocality	boolean	Defaults to False. Set to True to allow the jobs to run at sites reg where the input dataset is hosted (this parameter has effect only w Data.inputDataset is used). Remote file access is done using X parameter Site.whitelist is mandatory and Site.blacklist c used and it is respected. This parameter is useful to allow the jobs other sites when for example a dataset is hosted only on sites whi running CRAB jobs. When set to True, the user must provide also

		white-list of sites where the jobs could run that are physically clo site that hosts the input dataset. For example, if the input dataset i by a site in the USA, a reasonable site white-list would be `['T2_U` while `['T2*']` is NOT reasonable and will most likely result in m than gain
`userInputFiles`	list of strings	This parameter serves to run an analysis over a set of (private) inp as opposed to run over an input dataset from DBS. One has to pro this parameter the list of input files: `Data.userInputFiles = ['` `'file2', 'etc']`, where 'fileN' can be an LFN, a PFN or even a Xrootd redirector. One could also have a local text file containing input files (one file per line; don't include quotation marks nor co and then specify in this parameter the following: `Data.userInput` `open('/path/to/local/file.txt').readlines()`. When this p is used, the only allowed splitting mode is `'FileBased'`. Also, si is no input dataset from where to extract the primary dataset name should use the parameter `Data.outputPrimaryDataset` to define otherwise CRAB will use `'CRAB_UserFiles'` as the primary data This parameter can not be used together with `Data.inputDataset` will not do any data discovery, meaning that most probably jobs w run at the sites where the input files are hosted (and therefore they accessed via Xrootd). But since it is in general more efficient to r jobs at the sites where the input files are hosted, it is strongly reco that the user forces the jobs to be submitted to these sites using th `Site.whitelist` parameter.
Section `Site`		
`storageSite (**)`	string	Site where the output files should be permanently copied to. See t about `storageSite` below.
`whitelist`	list of strings	A user-specified list of sites where the jobs can run. For example: `['T2_CH_CERN','T2_IT_Bari',...]`. Jobs will not be assigned t that is not in the white list. Note that at times this list may not be see this FAQ
`blacklist`	list of strings	A user-specified list of sites where the jobs should not run. Usefu jobs to run on a site where the user knows they will fail (e.g. beca temporary problems with the site). Note that at times this list may respected, see this FAQ
`ignoreGlobalBlacklist`	boolean	Whether or not to ignore the global site blacklist provided by the Board. Should only be used in special cases with a custom whitel blacklist to make sure the jobs land on the intended sites.
Section `User`		
`voGroup`	string	The VO group that should be used with the proxy and under whic should be submitted.
`voRole`	string	The VO role that should be used with the proxy and under which should be submitted.
Section `Debug`		
`oneEventMode`	boolean	For experts use only.
`asoConfig`	list of dictionaries	For experts use only.
`scheddName`	string	For experts use only.
`extraJDL`	list of strings	For experts use only.
`collector`	string	For experts use only.

Note for `Data.splitting = 'EventAwareLumiBased'`

When CRAB does data discovery of the input dataset in DBS, the number of events is only known per input

file (because that's the information available on DBS) and not per luminosity section. CRAB can therefore only estimate the number of events per luminosity section in a given input file as the number of events in the file divided by the number of luminosity sections in the file. Because of that, `Data.unitsPerJob` and `Data.totalUnits` should not be considered by the user as rigorous limits, but as limits applicable on average.

Note for `maxJobRuntimeMin`
We strongly encourage every user to tune their splitting paramenters aiming for jobs to run for a few hours, ideally 8-10 hours, and set the `maxJobRuntimeMin` accordingly. Having many jobs increases the chance of failure, since the number of problems is roughly proportional to the number of run jobs. Moreover, short jobs suffer of start/end overheads resulting in poor CPU/Wall-clock ratio, which impacts negatively CMS and makes it harder to secure additional resources. The large default value for `maxJobRuntimeMin` prevents jobs to be scheduled within shorter site queues which are polled faster.

Note for `storageSite`
In CRAB3 the output files of each job are transferred first to a temporary storage element in the site where the job ran and later from there to a permanent storage element in a destination site. The transfer to the permanent storage element is done asynchronously by a service called AsyncStageOut (ASO). The destination site must be specified in the `Site.storageSite` parameter in the form `'Tx_yy_zzzzz'` (e.g. `'T2_IT_Bari'`, `'T2_US_Nebraska'`, etc.). The official names of CMS sites can be found in the CRIC web page. The user MUST have write permission in the storage site.

Passing CRAB configuration parameters from the command line

It is possible to define/overwrite CRAB configuration parameters by passing them through the command line when the `crab submit` command is executed. Parameters can be set with the convention `<parameter-name>=<parameter-value>` and can be sequentially listed separating them with a blank space. Here is an example on how one would pass the request name and the publication name:

```
crab submit -c my_crab_config_file.py General.requestName=my_request_name Data.outputDatasetTag=m
```

Note: Currently it is only possible to overwrite the parameters that take as value a string, an integer, a float or a boolean. Parameters that take a list can not be overwritten this way.

Converting a CRAB2 configuration file into a CRAB3 configuration file

CRAB3 is essentially new compared to CRAB2; it is not just a re-write. As a consequence, the configuration is different and there is no direct trivial translation that can be done automatically for every CRAB2 configuration file into a CRAB3 one. There is only a basic CRAB3 utility, called `crab2cfgTOcrab3py`, meant to help the user to convert an existing CRAB2 configuration file into a CRAB3 configuration file template. The user has to provide the name of the CRAB2 configuration file he/she wants to convert and the name he/she wants to give to the CRAB3 configuration file (both arguments have default values; `crab.cfg` and `crabConfig.py` respectively).

```
crab2cfgTOcrab3py [crab2confgiName.cfg] [crab3configName.py]
```

Instead of blindly taking the produced CRAB3 configuration file and run it, the user should always inspect the produced file, understand what each parameter means, edit them and add other parameters that might be needed, etc.

Here we give a usage example. Suppose we have the following CRAB2 configuration file with the default name `crab.cfg`:

```
[CRAB]
jobtype                 = cmssw
scheduler               = remoteGlidein
use_server              = 0

[CMSSW]
datasetpath             = /GenericTTbar/HC-CMSSW_5_3_1_START53_V5-v1/GEN-SIM-RECO
dbs_url                 = global
pset                    = my_CMSSW_config.py
number_of_jobs          = 100
events_per_job          = 20
output_file             = output.root

[GRID]
se_white_list           = T2_IT_Bari
se_black_list           = T2_IT_Legnaro
data_location_override  = T2_IT_Bari

[USER]
ui_working_dir          = my_CRAB_project_directory
return_data             = 0
copy_data               = 1
storage_element         = T2_IT_Legnaro
user_remote_dir         = my_remote_directory
publish_data            = 1
publish_data_name       = my_publication_name
dbs_url_for_publication = phys03
```

If we run the tool without specifying any input parameters:

```
crab2cfgTOcrab3py
```

it will create a file `crabConfig.py` with the following content:

```
from WMCore.Configuration import Configuration
config = Configuration()
config.section_('General')
config.General.transferOutputs = True
config.General.requestName = 'my_CRAB_project_directory'
config.section_('JobType')
config.JobType.psetName = 'my_CMSSW_config.py'
config.JobType.pluginName = 'Analysis'
config.JobType.outputFiles = ['output.root']
config.section_('Data')
config.Data.inputDataset = '/GenericTTbar/HC-CMSSW_5_3_1_START53_V5-v1/GEN-SIM-RECO'
config.Data.publication = True
config.Data.unitsPerJob = 20
config.Data.publishDBS = 'https://cmsweb.cern.ch/dbs/prod/phys03/DBSWriter/'
config.Data.splitting = 'EventBased'
config.Data.inputDBS = 'https://cmsweb.cern.ch/dbs/prod/global/DBSReader/'
config.Data.outputDatasetTag = 'my_publication_name'
config.section_('Site')
config.Site.blacklist = ['T2_IT_Legnaro']
config.Site.whitelist = ['T2_IT_Bari']
config.Site.storageSite = 'T2_IT_Legnaro'
```

and it will show the following screen output:

```
Convertion done!
crab2cfgTOcrab3py report:
CRAB2 parameters not YET supported in CRAB3:
   data_location_override,user_remote_dir
CRAB2 parameters obsolete in CRAB3:
   return_data,jobtype,scheduler,use_server
```

As we already emphasized, the template configuration file produced by the `crab2cfgTOcrab3py` utility should not be used before carefully looking into its content. Along this line, one can see for example that the parameter `JobType.outputFiles` was set to `['output.root']`. If `output.root` is defined in the CMSSW parameter-set configuration file in an output module, then it doesn't have to be included in the `JobType.outputFiles` list (although it doesn't harm).

-- AndresTanasijczuk - 02 Oct 2014

This topic: CMSPublic > CRAB3ConfigurationFile
Topic revision: r77 - 2020-08-28 - StefanoBelforte