

Table of Contents

CREAM-CE metrics and WN probes.....	1
CREAM-CE metrics.....	1
emi.wms.CREAMCE-JobState.....	1
The message transfer agent (MTA).....	1
Output files.....	2
Third-party WN checks.....	2
nagrun.sh.....	3
emi.wms.CREAMCE-JobMonit.....	4
WN Probes.....	4
WN-csh.....	5
WN-softver.....	5
WN-brokerinfo.....	5
Test.....	6
State + Monit + Cancel.....	6
State + Monit without notification.....	7
State + Monit with notification.....	10

CREAM-CE metrics and WN probes

CREAM-CE metrics

These metrics are used to test worker nodes submitting ad-hoc jdls which run some grid's checks.

1. **emi.cream.CREAMCE-JobState**. Submits grid job to CREAMCE using a given WMS. Accepts passive check updates from emi.wms.CREAMCE-JobMonit.
2. **emi.cream.CREAMCE-JobMonit**. Monitors grid jobs submitted to CREAM-CE.
3. **emi.cream.CREAMCE-JobCancel**. Cancel active grid jobs.
4. **emi.cream.CREAMCE-JobSubmit**. Passive check. Holds terminal status of job submission to CREAM-CE.

emi.wms.CREAMCE-JobState

Submit a grid job to a given CREAM-CE through a WMS. These are the generic parameters:

--wms <wms>	WMS to be used for job submission. If not given, default WMPProxy end-points defined on the UI will be used.
--jdl-templ <file>	JDL template file (full path). Default: <emi.cream.ProbesLocation>/CREAM-jdl.template
--jdl-retrycount <val>	JDL RetryCount (Default: 0).
--jdl-shallowretrycount <val>	JDL ShallowRetryCount (Default: 1).
--timeout-job-discard <sec>	Discard job after the timeout. (Default: 21600)
--prev-status <0-3>	Previous Nagios status of the metric.

This is the default jdl template used for submission:

```
Type="Job";
JobType="Normal";
Executable = "<jdlExecutable>";
StdError = "gridjob.out";
StdOutput = "gridjob.out";
Arguments = "<jdlArguments>";
InputSandbox = {"<jdlInputSandboxExecutable>", "<jdlInputSandboxTarball>"};
OutputSandbox = {"gridjob.out", "wnlogs.tgz"};
RetryCount = <jdlRetryCount>;
ShallowRetryCount = <jdlShallowRetryCount>;
Requirements = other.GlueCEInfoHostName == "<jdlReqCEInfoHostName>";
```

The message transfer agent (MTA)

To manage checks on the worker node it is used the executable `nagrund.sh` (see below). The arguments are dynamically composed by the metrics translating the ones given to the probes. In particular results from the worker nodes are sent via a message transfer agent (MTA) to Message Brokers. The code for MTA `mta-simple` is located under `<WN_codebase>/bin/` and implementation in `<WN_codebase>/lib/python2.3/site-packages/mig`.

The MTA :

- tries to establish connection to a broker (either a given one or found via discovery in IS and successive application of ranking)
- takes messages from directory based queue and sends them to the broker.

Messages with metric results are stored in outgoing messages queue by a Nagios handler `handle_service_check` invoked by Nagios after execution of each check. The parameters to manage resource broker are:

<code>--mb-destination <dest></code>	Mandatory parameter (if <code>--no-mb</code> is not specified). The destination queue/topic on Message Broker to publish to.
<code>--mb-uri <URI></code>	Message Broker URI. If not given, MB discovery will be performed on WN to find working MB. Format for <URI>: [failover:/^(<uri>,[...][\]) <uri> - stomp://FQDN:port/ or http://FQDN/message (Default: service discovery on WN.)
<code>--mb-network <net></code>	Brokers network for broker discovery on WN. (Default: PROD)
<code>--mb-no-discovery</code>	Do not do broker discovery on WN. If a given <URI> is not accessible, WN part of the framework will exit with UNKNOWN. (Default: if <URI> is not given or not accessible from WN perform broker discovery in <net>.)
<code>--mb-choice <best or random></code>	How to choose MB on WN. 'best' - min response time. (Default: best)
<code>--no-mb</code>	Do not send results messages to Message Broker

Note that the last option `--no-mb` disables messages transfer; in that case results e-mail messages can be found in the job's output file `wnlogs.tgz`.

Output files

Default JDL's output sandbox defines two files that will be taken from WN

```
OutputSandbox = {"gridjob.out", "wnlogs.tgz"};
```

- `gridjob.out` contains logging output of WN job as seen by WMS job wrapper. I.e., `stdout` and `stderr` from the testing framework launching script on WN (`nagrun.sh`).
- `wnlogs.tgz` contains the following directories from WN: `//nagios/{var,tmp}`. The framework's messaging and Nagios logging and debugging is stored there.
 - ◆ when writing probes for WN, one can direct output into some files in that directories - they will be brought to UI in `wnlogs.tgz`.

On Nagios Server/UI `OutputSandbox` is stored per CE under `/var/lib/gridprobes//emi.cream/CREAMCE//jobOutput*` directories. `jobOutput.LAST` contains last historical output from WN.

Third-party WN checks

To describe which checks must be executed on the worker node the following parameters should be used:

<code>--add-wntar-nag <d1,d2,...></code>	Comma-separated list of top level directories with Nagios compliant directories structure to be added to tarball to be sent to WN.
<code>--add-wntar-nag-nosam</code>	Instructs the metric not to include standard SAM WN probes and their Nagios config to WN tarball. (Default: WN probes are included)
<code>--add-wntar-nag-nosamcfg</code>	Instructs the metric not to include Nagios configuration for SAM WN probes to WN tarball. The probes themselves and respective Python packages, however, will be included.
<code>--wnjob-location <dir></code>	Full path to directory containing WN scheduler. (Default: <code><emi.cream.ProbesLocation>/wnjob</code>)

NOTE: with --add-wntar-nag <d1,d2,...> parameter the respective "Nagios compliant directories structure" should look like this:

```

|-- etc
|   |-- wn.d
|       |-- org.my
|           |-- commands.cfg
|           |-- services.cfg
|-- probes
    |-- org.my
        |-- check_A
        |-- check_B
        |-- checks_lib.sh
    
```

- probes/org.my/* should contain your probes/checks
- etc/wn.d/org.my/ should contain file(s) with .cfg extension with Nagios command and service objects definitions (optionally, service dependencies definitions). In your etc/wn.d/org.my/*.cfg files please use the following paths defining Nagios macros and the framework template names:
 - ◆ \$USER3\$ macro defining path to /probes/ directory on WN. Usage:

```

define command{
    command_name    check_A1
    command_line    $USER3$/org.my/check_A
}
    
```

- ◆ <wnjobWorkDir> will be substituted with the job's working directory on WN. Handy if your check requires and creates a working directory. Possible usage (assumes -w instructs check_A to create <wnjobWorkDir>/mygridprobes directory):

```

define command{
    command_name    check_A2
    command_line    $USER3$/org.my/check_A -w <wnjobWorkDir>/mygridprobes
}
    
```

For this particular part of Nagios objects configuration and macros please see the Nagios documentation for resources configuration.

With these last parameters you can manage some timeouts in WNs:

--timeout-wnjob-global <sec>	Global timeout for a job on WN. (Default: 600)
--wn-verb <0-3>	Metrics verbosity level on WN. [-v <VERBOSITY>] (Default: 0)
--wn-verb-fw <0-3>	Framework verbosity level on WN (Default: 1)

- --wn-verb on WN the given value is substituted for <VERBOSITY> in Nagios metric definition *.cfg files.
- --wn-verb-fw is used by nagrun.sh for setting its debugging output as well as setting logging and debugging output of messages publishing client (Message Transfer Agent - MTA) and Nagios.
 - ◆ >= 2 - 'debug'
 - ◆ == 1 - 'info'
 - ◆ == 0 - 'warn'

nagrun.sh

On WN (as specified in JDL with Executable = "<jdlExecutable>") nagrun.sh script (on Nagios UI located in <WN_codebase>) is used to

- set required environment variables
- launch messages transfer agent (MTA) - metric results publisher to message brokers
- make substitutions in templated (mainly Nagios) configuration files

- launch and monitor Nagios
- after all metrics are executed (or on timeout) terminate Nagios and MTA
- do on-exit cleanup

Parameters to nagrun.sh specified by Arguments = "" define what and how should be launched.

```
usage: nagrun.sh -v <vo> -d <dest> [-b <broker_uri>]
      [-n <broker_network>] [-t <timeout>] [-w <fw_verb>]
      [-z <metric_verb>] [-f <fqan>] [-i <host:port,..>] -B -R -N -h -m
-v and -d (if not -m) are mandatory paramters. Defaults:
<broker_network> - PROD
<timeout> - 600 sec
<metrics_verb> - 0
<fw_verb> - 1 (2 - messages, 3 - Nagios config/stats/debug)
-f <fqan> - VOMS FQAN
-B - don't do broker discovery
-R - take MB randomly; by default sort by min response time
-N - don't run WN tests
-m - don't use mta service to transfer messages
```

In most cases the parameters is the translation of corresponding ones given to emi.cream.CREAMCE-JobState metric.

emi.cream.CREAMCE-JobState	nagrun.sh
--mb-network <net> Brokers network for broker discovery on WN.	-n <broker_network>
--mb-uri <URI> Message Broker URI.	-b <broker_uri>
--mb-destination <dest> queue/topic on MB to publish to.	-d <dest>
--mb-no-discovery Do not do broker discovery on WN.	-B
--mb-choice <best or random> How to choose MB on WN.	-R
--vo <name> Virtual Organization.	-v <vo>
--vo-fqan <name> VOMS primary attribute as FQAN.	-f <fqan>
--wn-verb <0-3> Metrics verbosity level on WN.	-z <metric_verb>
--wn-verb-fw <0-3> Framework verbosity level on WN.	-w <fw_verb>
--timeout-wnjob-global <sec> Global timeout for a job on WN.	-t <timeout>
--no-mb	-m

emi.wms.CREAMCE-JobMonit

Monitors status of all submitted jobs (as defined in activejob.map files) and updates states of emi.cream.CREAMCE-JobState and emi.wms.CREAMCE-JobMonit metrics. Acts as a babysitter for all grid jobs submitted by emi.cream.CREAMCE-JobState. emi.cream.CREAMCE-JobState and emi.cream.CREAMCE-JobMonit are updated (as passive checks) either via Nagios command file or NSCA. It accepts these parameters:

```
--timeout-job-global <sec>      Global timeout for jobs. Job will be cancelled and dropped
                                if it is not in terminal state by that time. (Default: 3300)
--timeout-job-waiting <sec>     Time allowed for a job to stay in Waiting with 'no compatible
                                resources'. (Default: 2700)
--timeout-job-discard <sec>     Discard job after the timeout. (Default: 21600)
--timeout-job-schedrun <sec>    Scheduled/Running states timeout. (Default: 19800)
--hosts <h1,h2,..>             Comma-separated list of CE hostnames to run the monitor on.
```

WN Probes

Using the default wntag directory `emi.wn` these probes are performed on the worker node using the wrapper `samtest-run` :

- WN-csh
- WN-softver
- WN-brokerinfo

WN-csh

Checking if CSH works running the command: `/bin/csh -c "env|sort" > env-csh.txt` and then checking if the variable `PATH` is defined. Accept only the parameter: `debug`.

Example of a message sent as output:

```
serviceURI: cream-30.pd.infn.it:8443/cream-pbs-creamtest2
hostName: localhost.localdomain
serviceFlavour: CE
siteName: FAKE-SITE
metricStatus: OK
metricName: emi.wn.WN-Csh
summaryData: OK
gatheredAt: cream-wn-030.pn.pd.infn.it
timestamp: 2011-09-22T15:25:36Z
nagiosName: emi.wn.WN-Csh-dteam
role: site
voName: dteam
serviceType: emi.wn.WN
detailsData: Checking if CSH works\nTest: OK.\n
```

WN-softver

Detects the version of software which is really installed on the WN. To detect the version `lcg-version`, `glite-version` commands and the `cat` of the `/etc/emi-version` file are tried and if the commands are not available the script exits with an error.

Example of a message sent as output:

```
serviceURI: cream-30.pd.infn.it:8443/cream-pbs-creamtest2
hostName: localhost.localdomain
serviceFlavour: CE
siteName: FAKE-SITE
metricStatus: OK
metricName: emi.wn.WN-SoftVer
summaryData: OK: EMI 1.2.0-1
gatheredAt: cream-wn-030.pn.pd.infn.it
timestamp: 2011-09-22T15:25:37Z
nagiosName: emi.wn.WN-SoftVer-dteam
role: site
voName: dteam
serviceType: emi.wn.WN
detailsData: Installed software version\n+ type=unknow\n+ mwver=error\n+ type -f glite-version\n/
```

WN-brokerinfo

Check if BrokerInfo works. The procedure is the following:

- Firstly check if BrokerInfo file is defined in `$GLITE_WMS_RB_BROKERINFO`, `$GLITE_WL_RB_BROKERINFO` or `$EDG_WL_RB_BROKERINFO` variables

CreamProbe < CREAM < TWiki

- Then try to get CE host name using `edg-brokerinfo getCE` or `glite-brokerinfo getCE` command respectively. If previous command result value is different from 0 test is failed.

Example of a message sent as output:

```
serviceURI: cream-30.pd.infn.it:8443/cream-pbs-creamtest2
hostName: localhost.localdomain
serviceFlavour: CE
siteName: FAKE-SITE
metricStatus: OK
metricName: emi.wn.WN-Bi
summaryData: OK: getCE: cream-30.pd.infn.it:8443/cream-pbs-creamtest2
gatheredAt: cream-wn-030.pn.pd.infn.it
timestamp: 2011-09-22T15:25:35Z
nagiosName: emi.wn.WN-Bi-dteam
role: site
voName: dteam
serviceType: emi.wn.WN
detailsData: Checking if BrokerInfo works\nBrokerInfo file: /home/dteam009/home_cre30_262167654/C
```

Test

To test the probe you have to create a valid proxy.

State + Monit + Cancel

First you have to "submit" a jdl:

```
/usr/libexec/grid-monitoring/probes/emi.cream/CREAMCE-probe --vo <vo> -x <path of the proxy> -H  
<CREAM-ce hostname> -m emi.cream.CREAMCE-JobState --wms <WMS hostname> --no-mb
```

```
[ale@cream-48 ~]$ /usr/libexec/grid-monitoring/probes/emi.cream/CREAMCE-probe --vo dteam -x /tmp/  
OK: [Submitted]  
OK: [Submitted]
```

Connecting to the service `https://cream-45.pd.infn.it:7443/glite_wms_wmproxy_server`

```
===== glite-wms-job-submit Success =====
```

The job has been successfully submitted to the WMPProxy
Your job identifier is:

```
https://cream-45.pd.infn.it:9000/fvNWcPJ6nVXAQJqhuU29-g
```

The job identifier has been saved in the following file:
`/var/lib/gridprobes/dteam/emi.cream/CREAMCE/cream-19.pd.infn.it/jobID`

```
=====
```

You can "monitor" the job:

```
/usr/libexec/grid-monitoring/probes/emi.cream/CREAMCE-probe --vo <vo> -x <path of the proxy> -H  
<CREAM-ce hostname> -m emi.cream.CREAMCE-JobMonit --pass-check-dest active
```

```
[ale@cream-48 ~]$ /usr/libexec/grid-monitoring/probes/emi.cream/CREAMCE-probe --vo dteam -x /tmp/  
metric results >>> <cream-19.pd.infn.it,emi.cream.CREAMCE-JobState-dteam>  
OK: [Scheduled] https://cream-45.pd.infn.it:9000/fvNWcPJ6nVXAQJqhuU29-g  
OK: [Scheduled] https://cream-45.pd.infn.it:9000/fvNWcPJ6nVXAQJqhuU29-g  
Testing from: cream-48.pd.infn.it
```

CreamProbe < CREAM < TWiki

```
DN: /C=IT/O=INFN/OU=Personal Certificate/L=Padova/CN=Alessio Gianelle/CN=proxy
VOMS FQANs: /dteam/Role=NULL/Capability=NULL, /dteam/NGI_IT/Role=NULL/Capability=NULL
glite-wms-job-status https://cream-45.pd.infn.it:9000/fvNWcPJ6nVXAQJqhuU29-g
```

```
===== glite-wms-job-status Success =====
BOOKKEEPING INFORMATION:
```

```
Status info for the Job : https://cream-45.pd.infn.it:9000/fvNWcPJ6nVXAQJqhuU29-g
Current Status:      Scheduled
Status Reason:      unavailable
Destination:        cream-19.pd.infn.it:8443/cream-lsf-creamcert2
Submitted:          Wed Nov  9 16:43:31 2011 CET
```

```
=====
OK: Jobs processed - 1
OK: Jobs processed - 1
[Scheduled] : 1|jobs_processed=1;; DONE=0;; RUNNING=0;; SCHEDULED=1;; SUBMITTED=0;; READY=0;; WAITING=0
```

Before it finishes you can "cancel" it:

`/usr/libexec/grid-monitoring/probes/emi.cream/CREAMCE-probe --vo <vo> -x <path of the proxy> -H <CREAM-ce hostname> -m emi.cream.CREAMCE-JobCancel`

```
[ale@cream-48 ~]$ /usr/libexec/grid-monitoring/probes/emi.cream/CREAMCE-probe --vo dteam -x /tmp/
OK: job cancelled
OK: job cancelled
Testing from: cream-48.pd.infn.it
DN: /C=IT/O=INFN/OU=Personal Certificate/L=Padova/CN=Alessio Gianelle/CN=proxy
VOMS FQANs: /dteam/Role=NULL/Capability=NULL, /dteam/NGI_IT/Role=NULL/Capability=NULL
Job cancellation request sent:
glite-wms-job-cancel --noint -i /var/lib/gridprobes/dteam/emi.cream/CREAMCE/cream-19.pd.infn.it/
Job bookkeeping files deleted.
```

You can verify that it works correctly checking the status of the job.

```
[ale@cream-48 ~]$ glite-wms-job-status https://cream-45.pd.infn.it:9000/fvNWcPJ6nVXAQJqhuU29-g
```

```
===== glite-wms-job-status Success =====
BOOKKEEPING INFORMATION:
```

```
Status info for the Job : https://cream-45.pd.infn.it:9000/fvNWcPJ6nVXAQJqhuU29-g
Current Status:      Cancelled
Destination:        cream-19.pd.infn.it:8443/cream-lsf-creamcert2
Submitted:          Wed Nov  9 16:43:31 2011 CET
```

State + Monit without notification

As before submit a job disabling the messages transfer (option `--no-mb`):

`/usr/libexec/grid-monitoring/probes/emi.cream/CREAMCE-probe --vo <vo> -x <path of the proxy> -H <CREAM-ce hostname> -m emi.cream.CREAMCE-JobState --wms <WMS hostname> --no-mb`

Then you can monitor the job until it ends:

`/usr/libexec/grid-monitoring/probes/emi.cream/CREAMCE-probe --vo <vo> -x <path of the proxy> -H <CREAM-ce hostname> -m emi.cream.CREAMCE-JobMonit --pass-check-dest active`

At the end when job finished the execution of the `emi.cream.CREAMCE-JobMonit` metrics should trigger also a `get_output`

State + Monit + Cancel

CreamProbe < CREAM < TWiki

```
[ale@cream-48 ~]$ /usr/libexec/grid-monitoring/probes/emi.cream/CREAMCE-probe --vo dteam -x /tmp/metric results >>> <cream-19.pd.infn.it,emi.cream.CREAMCE-JobSubmit-dteam>
OK: success.
OK: success.
```

```
Testing from: cream-48.pd.infn.it
DN: /C=IT/O=INFN/OU=Personal Certificate/L=Padova/CN=Alessio Gianelle/CN=proxy
VOMS FQANs: /dteam/Role=NULL/Capability=NULL, /dteam/NGI_IT/Role=NULL/Capability=NULL
glite-wms-job-status https://cream-45.pd.infn.it:9000/qAAznOmFalzWi5eTYrjAlQ
```

```
===== glite-wms-job-status Success =====
BOOKKEEPING INFORMATION:
```

```
Status info for the Job : https://cream-45.pd.infn.it:9000/qAAznOmFalzWi5eTYrjAlQ
Current Status:      Done (Success)
Exit code:           0
Status Reason:      Job Terminated Successfully
Destination:        cream-19.pd.infn.it:8443/cream-lsf-creamcert2
Submitted:           Wed Nov  9 18:11:03 2011 CET
=====
```

```
Getting job output: OK.
```

```
metric results >>> <cream-19.pd.infn.it,emi.cream.CREAMCE-JobState-dteam>
OK: success.
OK: success.
```

```
Testing from: cream-48.pd.infn.it
DN: /C=IT/O=INFN/OU=Personal Certificate/L=Padova/CN=Alessio Gianelle/CN=proxy
VOMS FQANs: /dteam/Role=NULL/Capability=NULL, /dteam/NGI_IT/Role=NULL/Capability=NULL
glite-wms-job-status https://cream-45.pd.infn.it:9000/qAAznOmFalzWi5eTYrjAlQ
```

```
===== glite-wms-job-status Success =====
BOOKKEEPING INFORMATION:
```

```
Status info for the Job : https://cream-45.pd.infn.it:9000/qAAznOmFalzWi5eTYrjAlQ
Current Status:      Done (Success)
Exit code:           0
Status Reason:      Job Terminated Successfully
Destination:        cream-19.pd.infn.it:8443/cream-lsf-creamcert2
Submitted:           Wed Nov  9 18:11:03 2011 CET
=====
```

```
Getting job output: OK.
```

```
OK: Jobs processed - 1
OK: Jobs processed - 1
Done : 1|jobs_processed=1;; DONE=1;; RUNNING=0;; SCHEDULED=0;; SUBMITTED=0;; READY=0;; WAITING=0;
```

You can verify that it works correctly checking the status of the job.

```
[ale@cream-48 ~]$ glite-wms-job-status https://cream-45.pd.infn.it:9000/qAAznOmFalzWi5eTYrjAlQ
```

```
===== glite-wms-job-status Success =====
BOOKKEEPING INFORMATION:
```

```
Status info for the Job : https://cream-45.pd.infn.it:9000/qAAznOmFalzWi5eTYrjAlQ
Current Status:      Cleared
Status Reason:      user retrieved output sandbox
Destination:        cream-19.pd.infn.it:8443/cream-lsf-creamcert2
Submitted:           Wed Nov  9 18:11:03 2011 CET
=====
```

To check that metrics on the worker node run correctly you can edit the output files in

CreamProbe < CREAM < TWiki

```
gatheredAt: cream-wn-007.pn.pd.infn.it
timestamp: 2011-11-09T17:11:31Z
nagiosName: emi.wn.WN-Csh-dteam
role: site
voName: dteam
serviceType: emi.wn.WN
detailsData: Checking if CSH works\nTest: OK.\n
EOT
```

Of the **emi.wn.WN-SoftVer** metrics:

```
serviceURI: cream-19.pd.infn.it:8443/cream-lsf-creamcert2
hostName: localhost.localdomain
serviceFlavour: CE
siteName: INFN-EMITESTBED
metricStatus: OK
metricName: emi.wn.WN-SoftVer
summaryData: OK: gLite 3.1.0
gatheredAt: cream-wn-007.pn.pd.infn.it
timestamp: 2011-11-09T17:11:32Z
nagiosName: emi.wn.WN-SoftVer-dteam
role: site
voName: dteam
serviceType: emi.wn.WN
detailsData: Installed software version\n+ type=unknow\n+ mwver=error\n+ type -f glite-version\n
EOT
```

Of the **emi.wn.WN-Bi** metrics:

```
serviceURI: cream-19.pd.infn.it:8443/cream-lsf-creamcert2
hostName: localhost.localdomain
serviceFlavour: CE
siteName: INFN-EMITESTBED
metricStatus: OK
metricName: emi.wn.WN-Bi
summaryData: OK: getCE: cream-19.pd.infn.it:8443/cream-lsf-creamcert2
gatheredAt: cream-wn-007.pn.pd.infn.it
timestamp: 2011-11-09T17:11:30Z
nagiosName: emi.wn.WN-Bi-dteam
role: site
voName: dteam
serviceType: emi.wn.WN
detailsData: Checking if BrokerInfo works\nBrokerInfo file: /home/dteam017/home_cre19_460125504/C
EOT
```

State + Monit with notification

To test also the mechanism of messages transfer you need to install a Message Broker.

Then you can "submit" a job using this command:

```
/usr/libexec/grid-monitoring/probes/emi.cream/CREAMCE-probe --vo <vo> -x <path of the proxy> -H  
<CREAM-ce hostname> -m emi.cream.CREAMCE-JobState --wms <WMS hostname> --mb-uri  
<Message Broker URI> --mb-destination <Message Broker destination>
```

```
[ale@cream-48 ~]$ /usr/libexec/grid-monitoring/probes/emi.cream/CREAMCE-probe --vo dteam -x /tmp/
OK: [Submitted]
OK: [Submitted]
```

```
Connecting to the service https://cream-45.pd.infn.it:7443/glite_wms_wmproxy_server
```

```
===== glite-wms-job-submit Success =====
```

CreamProbe < CREAM < TWiki

The job has been successfully submitted to the WMPProxy
Your job identifier is:

<https://cream-45.pd.infn.it:9000/3sqXvSstpobzaQhTzWNH4Q>

The job identifier has been saved in the following file:
/var/lib/gridprobes/dteam/emi.cream/CREAMCE/cream-19.pd.infn.it/jobID

=====

Again you can, as usual, monitor the job until it terminates:

/usr/libexec/grid-monitoring/probes/emi.cream/CREAMCE-probe --vo <vo> -x <path of the proxy> -H <CREAM-ce hostname> -m emi.cream.CREAMCE-JobMonit --pass-check-dest active

At the end you can do the same checks as in the previous test, but also you can check the log of the Message Broker Server to see if it receives the messages, as in this example:

```
2011-11-10 11:17:36,856 [Thread-2] coilmq.server.socketserver.StompRequestHandler - DEBUG - Processed
content-length:1020
ROC:UNDEFINED
sitename:INFN-EMITESTBED
destination:/tmp/msg
persistent:true
nagios_host:localhost.localdomain
role:site
```

```
serviceURI: cream-19.pd.infn.it:8443/cream-lsf-creamcert2
hostName: localhost.localdomain
serviceFlavour: CE
siteName: INFN-EMITESTBED
metricStatus: OK
metricName: emi.wn.WN-Bi
summaryData: OK: getCE: cream-19.pd.infn.it:8443/cream-lsf-creamcert2
gatheredAt: cream-wn-007.pn.pd.infn.it
timestamp: 2011-11-10T10:17:36Z
nagiosName: emi.wn.WN-Bi-dteam
role: site
voName: dteam
serviceType: emi.wn.WN
detailsData: Checking if BrokerInfo works\nBrokerInfo file: /home/dteam017/home_cre19_378000412/C
EOT
```

```
2011-11-10 11:17:37,854 [Thread-2] coilmq.server.socketserver.StompRequestHandler - DEBUG - Processed
content-length:397
ROC:UNDEFINED
sitename:INFN-EMITESTBED
destination:/tmp/msg
persistent:true
nagios_host:localhost.localdomain
role:site
```

```
serviceURI: cream-19.pd.infn.it:8443/cream-lsf-creamcert2
hostName: localhost.localdomain
serviceFlavour: CE
siteName: INFN-EMITESTBED
metricStatus: OK
metricName: emi.wn.WN-Csh
summaryData: OK
gatheredAt: cream-wn-007.pn.pd.infn.it
timestamp: 2011-11-10T10:17:37Z
nagiosName: emi.wn.WN-Csh-dteam
role: site
voName: dteam
```

CreamProbe < CREAM < TWiki

```
serviceType: emi.wn.WN
detailsData: Checking if CSH works\nTest: OK.\n
EOT
```

```
2011-11-10 11:17:38,854 [Thread-2] coilmq.server.socketserver.StompRequestHandler - DEBUG - Process
content-length:659
ROC:UNDEFINED
sitename:INFN-EMITESTBED
destination:/tmp/msg
persistent:true
nagios_host:localhost.localdomain
role:site
```

```
serviceURI: cream-19.pd.infn.it:8443/cream-lsf-creamcert2
hostName: localhost.localdomain
serviceFlavour: CE
siteName: INFN-EMITESTBED
metricStatus: OK
metricName: emi.wn.WN-SoftVer
summaryData: OK: gLite 3.1.0
gatheredAt: cream-wn-007.pn.pd.infn.it
timestamp: 2011-11-10T10:17:38Z
nagiosName: emi.wn.WN-SoftVer-dteam
role: site
voName: dteam
serviceType: emi.wn.WN
detailsData: Installed software version\n+ type=unknow\n+ mwver=error\n+ type -f glite-version\nng
EOT
```

This topic: [CREAM > CreamProbe](#)

Topic revision: r7 - 2011-11-10 - AlessioGianelle



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