

---

# **NorduGrid ARC Documentation**

***Release 6.0***

**NorduGrid Collaboration**

**Aug 29, 2018**



## CONTENTS:

<b>1</b>	<b>End-user documentation</b>	<b>1</b>
1.1	Try ARC6 or towards distributed computing in few minutes . . . . .	1
1.2	ARC6 Installation Guide . . . . .	5
1.3	ARC Configuration Reference Document . . . . .	12
1.4	Working with RunTime Environments in ARC6 . . . . .	119
1.5	Deploying ARCHERY for Project/VO . . . . .	125
<b>2</b>	<b>Documentation for developers</b>	<b>131</b>
2.1	Packages and Services Naming changes in ARC6 . . . . .	131
2.2	General arc.conf python configuration parser . . . . .	131
2.3	LRMS shell-backends overview for developers . . . . .	136
<b>3</b>	<b>ARC Utilities command line reference</b>	<b>143</b>
3.1	ARC Control Tool . . . . .	143
3.2	ARC Configuration Parser . . . . .	158



## END-USER DOCUMENTATION

### 1.1 Try ARC6 or towards distributed computing in few minutes

Scared of distributed computing complexities?

With ARC6 you can setup *Computing Element* and try the common distributed computing workflows just in few minutes!

ARC6 comes with so-called *zero configuration* included and works out of the box without any configuration at all.

#### 1.1.1 Step 1. Enable Nordugrid ARC6 repos

**Warning:** ALL THIS PART IS FOR DEVELOPERS-ORIENTED TESTING ONLY AND SHOULD BE REPLACED WITH `yum install http://download.nordugrid.org/packages/nordugrid-release/releases/6-alpha/centos/el7/x86_64/nordugrid-release-6.0.0-alpha.el7.centos.noarch.rpm` AT THE TIME OF ALPHA RELEASE

At the time of writing ARC6 is available as a [nightly builds packages](#) for many Linux distribution.

To add nightlies repo to your system<sup>1</sup> create a `/etc/yum.repos.d/nordugrid-nightly.repo` with the following content:

```
[nordugrid-nightly]
name=Nordugrid ARC Master Nightly Builds - $basearch
baseurl=http://builds.nordugrid.org/nightlies/packages/nordugrid-arc/master/
↪$arcnightly/centos/el7/$basearch
enabled=1
gpgcheck=0
```

And setup a CRON job (e.g. `/etc/cron.daily/update-arc-nightly-time.sh`) to fetch the latest nightlies date:

```
#!/bin/bash
curl -s http://builds.nordugrid.org/nightlies/packages/nordugrid-arc/master/ | sed ↪
↪-n 's/^\.*<a.*>\(.*\)\\</a>.*$/\1/p' | sort | tail -1 > /etc/yum/vars/arcnightly
```

#### 1.1.2 Step 2. Install A-REX

ARC Resource Execution service (A-REX) is a core component that manages authentication, authorization and job life cycle. It is enough to have A-REX installed to have minimal computing element:

---

<sup>1</sup> Package installation examples in this guide assumes you are running RHEL-based distributions (TODO: someone test on Ubuntu/Debian and add examples)

```
[root ~]# yum -y install nordugrid-arc-arex
```

### 1.1.3 Step 3. Run A-REX

To start ARC services just run the:

```
[root ~]# arcctl service start --as-configured
```

You can check the A-REX is running with:

```
[root ~]# arcctl service list
arc-acix-index           (Not installed, Disabled, Stopped)
arc-acix-scanner        (Not installed, Disabled, Stopped)
arc-arex                 (Installed, Enabled, Running)
arc-datadelivery-service (Not installed, Disabled, Stopped)
arc-gridftpd            (Not installed, Disabled, Stopped)
arc-infosys-ldap        (Not installed, Disabled, Stopped)
```

---

**Note:** arcctl tool automate many ARC CE operations and designed with bash-completion in mind. If you would like to go futher this quick start guide it is advised to have completion enabled:

```
[root ~]# yum install bash-completion python-argcomplete
[root ~]# activate-global-python-argcomplete
```

### 1.1.4 Step 4. Generate test user certificate and run job

Grid services and users authentication is haviely rely on cryptography and uses certificates for each entity. ARC6 comes with Test Certificate Authority on board that can issue the test user certificates easiely.

You can test submission from the host running A-REX or from any other host in the network.

#### Submitting jobs from another host via the network

Zero configured A-REX comes with EMI-ES and REST interfaces enabled. It runs on port 443, so make sure it is not firewalled to be used from client host.

**Install ARC client tools on the client host::** [root ~]# yum -y install nordugrid-arc-client

On the **A-REX host** generate user test certificate:

```
[root ~]# arcctl test-ca usercert --export-tar
User certificate and key are exported to testcert-08272152.tar.
To use test cert with arc* tools on the other machine, copy the tarball and run_
↳following:
  tar xf testcert-08272152.tar
  source arc-test-certs/usercerts.sh
```

Transfer the tarball to the client host and execute the commands suggested in the previous command output:

```
[user ~]$ tar xf /tmp/testcert-08272152.tar
[user ~]$ source arc-test-certs/usercerts.sh
```

Single Sign-On token for grid-infrastructure access is so-called *proxy-certificate* that needs to be generated in the following way:

```
[user ~]$ arcproxy
Your identity: /DC=org/DC=nordugrid/DC=ARC/O=TestCA/CN=Test Cert 08272152
Proxy generation succeeded
Your proxy is valid until: 2018-08-28 09:54:24
```

You can start with information query about your newly installed ARC computing element<sup>2</sup>:

```
[user ~]$ arcinfo -c https://arc.example.org/arex
Computing service:
  Information endpoint: https://arc.example.org:443/arex
  Information endpoint: https://arc.example.org:443/arex
  Information endpoint: https://arc.example.org:443/arex
  Submission endpoint: https://arc.example.org:443/arex (status: ok, interface: ↪
↪org.ogf.glue.emies.activitycreation)
  Submission endpoint: https://arc.example.org:443/arex (status: ok, interface: ↪
↪org.ogf.bes)
  Submission endpoint: https://arc.example.org:443/arex (status: ok, interface: ↪
↪org.nordugrid.arcrest)
```

**A simple job can be submitted with arctest tool:** [user ~]\$ arctest -J 2 -c https://arc.example.org/arex Submitting test-job 2: &( executable = "/usr/bin/env" )( stdout = "stdout" )( stderr = "stdout" )( gmlog = "gmlog" )( jobname = "arctest2" )( clientxrsl = "&( executable = ""/usr/bin/env"" )( jobname = ""arctest2"" )( stdout = ""stdout"" )( join = ""yes"" )( gmlog = ""gmlog"" )" ) Client version: nordugrid-arc-20180822231219 Test submitted with jobid: https://arc.example.org:443/arex/NhlKDmsmeEtnPSAtDmVmuSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm

**The job status can be checked with:** [user ~]\$ arcstat https://arc.example.org:443/arex/NhlKDmsmeEtnPSAtDmVmuSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm Job: https://arc.example.org:443/arex/NhlKDmsmeEtnPSAtDmVmuSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm

Name: arctest2 State: Running

Status of 1 jobs was queried, 1 jobs returned information

**Job stdout[#]\_:** [user ~]\$ arcctl https://arc.example.org:443/arex/NhlKDmsmeEtnPSAtDmVmuSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm GRIDMAP=/dev/null HOSTNAME=arc.zero TMPDIR=/tmp GLOBUS\_LOCATION=/usr <output omitted>

### Submitting jobs from A-REX host

You can use EMI-ES and REST in the same way as when submitting from the dedicated client host. The only difference that you can use `arcctl test-ca usercert` without `--export-tar` option due to files transfer is not needed.

However when running locally to A-REX you are able to use *internal* submission interface that communicate directly with A-REX via filesystem<sup>3</sup>:

```
[root ~]# arctest -S org.nordugrid.internal -J 2 -c localhost
Submitting test-job 2:
&( executable = "/usr/bin/env" )( stdout = "stdout" )( stderr = "stdout" )( gmlog ↪
↪= "gmlog" )( jobname = "arctest2" )( clientxrsl = "&( executable = ""/usr/bin/env
↪"" )( jobname = ""arctest2"" )( stdout = ""stdout"" )( join = ""yes"" )( gmlog =
↪""gmlog"" )" )
Client version: nordugrid-arc-20180822231219
Test submitted with jobid: file:///var/spool/arc/sessiondir/
↪zTVNDmd80EtnPSAtDmVmuSEmABFKDmABFKDmxiHKDmABFKDm0JAfan

[root ~]# arcstat file:///var/spool/arc/sessiondir/
↪zTVNDmd80EtnPSAtDmVmuSEmABFKDmABFKDmxiHKDmABFKDm0JAfan
Job: file:///var/spool/arc/sessiondir/
↪zTVNDmd80EtnPSAtDmVmuSEmABFKDmABFKDmxiHKDmABFKDm0JAfan
```

<sup>2</sup> Examples uses `arc.example.org` as a domain name for A-REX host

<sup>3</sup> Not working due to bug #3741

```
Name: arctest2
State: Accepted
```

```
Status of 1 jobs was queried, 1 jobs returned information
```

The following workflow is the same.

## Step 5. Play more with ARC computing element

Another point that every admin should look into - the logs and directories that ARC computing element uses. The brief list of the relevant paths can be obtained from:

```
[root ~]# arcctl config brief
ARC Storage Areas:
  Control directory:
    /var/spool/arc/jobstatus
  Session directories:
    /var/spool/arc/sessiondir
  Scratch directory on Worker Node:
    Not configured
  Additional user-defined RTE directories:
    Not configured
ARC Log Files:
  A-REX Service log:
    /var/log/arc/arex.log
  A-REX Jobs log:
    /var/log/arc/arex-jobs.log
  A-REX Helpers log:
    /var/log/arc/job.helper.errors
  A-REX WS Interface log:
    /var/log/arc/ws-interface.log
  Infosys Infoproviders log:
    /var/log/arc/infoprovider.log
```

To work with jobs on A-REX side look into `arcctl job4`. Operations includes but not limited to:

- **Listing jobs::** [root ~]# arcctl job list NhlKDmsmeEtnPSAtDmV-muSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm zb0LDm7RfEtnPSAtDmVmuSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm <output omitted>  
  
[root ~]# arcctl job list -long NhlKDmsmeEtnPSAtDmVmuSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm FINISHED arctest2 /DC=org/DC=nordugrid/DC=ARC/O=TestCA/CN=Test Cert 08272152 zb0LDm7RfEtnPSAtDmVmuSEmABFKDmABFKDm2PJKDmDBFKDme1CYXm FINISHED arctest2 /DC=org/DC=nordugrid/DC=ARC/O=TestCA/CN=Test Cert 08272234 <output omitted>
- **Job general information::** [root ~]# arcctl job info NhlKDmsmeEtnPSAtDmV-muSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm Name : arctest2 Owner : /DC=org/DC=nordugrid/DC=ARC/O=TestCA/CN=Test Cert 08272152 State : FINISHED LRMS ID : 16890 Modified : 2018-08-27 22:06:34
- **Job log::** [root ~]# arcctl job log NhlKDmsmeEtnPSAtDmVmuSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm 2018-08-27T22:03:34Z Job state change UNDEFINED -> ACCEPTED Reason: (Re)Accepting new job 2018-08-27T22:03:34Z Job state change ACCEPTED -> PREPARING Reason: Starting job processing 2018-08-27T22:03:34Z Job state change PREPARING -> SUBMIT Reason: Pre-staging finished, passing job to LRMS 2018-08-27T22:03:36Z Job state change SUBMIT -> INLRMS Reason: Job is passed to LRMS 2018-08-27T22:06:34Z Job state change INLRMS -> FINISHING Reason: Job finished executing in LRMS 2018-08-27T22:06:34Z Job state change FINISHING -> FINISHED Reason: Stage-out finished.

---

<sup>4</sup> Not working due to bugs #3742, #3743



- A-REX logs that mentioning the job::** [root ~]# arcctl job log NhlKDmsmeEtnPSAtDmV-muSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm -service ### /var/log/arc/arex.log: [2018-08-27 22:03:34] [Arc] [INFO] [16298/4] NhlKDmsmeEtnPSAtDmV-muSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm: State: ACCEPTED: parsing job description [2018-08-27 22:03:34] [Arc] [INFO] [16298/4] NhlKDmsmeEtnPSAtDmV-muSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm: State: ACCEPTED: moving to PREPARING [2018-08-27 22:03:34] [Arc] [INFO] [16298/4] NhlKDmsmeEtnPSAtDmV-muSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm: State: PREPARING from ACCEPTED [2018-08-27 22:03:34] [Arc] [INFO] [16298/4] NhlKDmsmeEtnPSAtDmV-muSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm: State: SUBMIT from PREPARING [2018-08-27 22:03:34] [Arc] [INFO] [16298/4] NhlKDmsmeEtnPSAtDmV-muSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm: state SUBMIT: starting child: /usr/share/arc/submit-fork-job [2018-08-27 22:03:36] [Arc] [INFO] [16298/4] NhlKDmsmeEtnPSAtDmV-muSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm: state SUBMIT: child exited with code 0 [2018-08-27 22:03:36] [Arc] [INFO] [16298/4] NhlKDmsmeEtnPSAtDmV-muSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm: State: INLRMS from SUBMIT [2018-08-27 22:06:34] [Arc] [INFO] [16298/4] NhlKDmsmeEtnPSAtDmV-muSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm: Job finished [2018-08-27 22:06:34] [Arc] [INFO] [16298/4] NhlKDmsmeEtnPSAtDmV-muSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm: State: FINISHING from INLRMS [2018-08-27 22:06:34] [Arc] [INFO] [16298/4] NhlKDmsmeEtnPSAtDmV-muSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm: State: FINISHED from FINISHING [2018-08-27 22:06:34] [Arc] [ERROR] [16298/4] NhlKDmsmeEtnPSAtDmV-muSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm: Job monitoring stop requested with 1 active references ### /var/log/arc/ws-interface.log: [2018-08-27 22:05:32] [Arc.A-REX] [INFO] [16298/42] GET: id NhlKDmsmeEtnPSAtDmV-muSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm path stdout [2018-08-27 22:05:49] [Arc.A-REX] [INFO] [16298/43] GET: id NhlKDmsmeEtnPSAtDmV-muSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm path stdout
- Getting job attributes::** [root ~]# arcctl job attr NhlKDmsmeEtnPSAtDmV-muSEmABFKDmABFKDm2PJKDmBBFKDmxDyQbm jobname arctest2

Now you are ready to arc6\_install\_guide!

## 1.2 ARC6 Installation Guide

### 1.2.1 PREREQUISITES

#### Choosing the host

It is assumed that ARC CE is installed on the top of existing Linux computing cluster. There are many Linux distributions [are supported](#). It can be installed on complete virtual computing cluster environment in the cloud.

ARC is not intrusive to existing system. We suggest to deploy ARC CE on the dedicated (virtual) machine connected to the cluster network and filesystem. In the limited number of cases it is possible to communicate with cluster over SSH from the completely independent remote node.

#### Plan for storage areas

Several storage areas are necessary for jobs execution and data storing. You should mount/export following directories:

- session directory
- datastaging cache directory (if planned)
- decide to what extent to use not cross-mounted scratch directory on the worker nodes

[TODO: what is session dir] [TODO: what is cache] [TODO: local scratch]

### Local resource management system (LRMS)

Install and configure your LRMS. ARC supports a variety of LRMS backends:

- fork (default) - execute jobs on the ARC CE host. Targeted for testing and development but not for real production jobs.
- condor - uses HTCondor powered HTC resource
- slurm - SLURM cluster
- pbs - any flavor of PBS batch system, including Torque and PBSPro
- ll - Load Leveler
- lsf - Load Sharing Facility
- sge - Oragle Grid Engine (formerly Sun Grid Engine)
- boinc - work as a gateway to BOINC volunteer computing

Check you are able to submit jobs from ARC CE host.

You may consider to setting up dedicated queues to use with ARC CE (e.g. per-VO queues).

Please also NOTICE that in some cases (depends on LRMS) you need to share batch system logs directories with ARC CE. [TODO: link to fulldoc]

### Configure OS accounts

Plan for local account (or account pools) that will be used to execute jobs on the worker nodes.

This accounts should be also available on ARC CE.

Please note that ARC services are run as root on the ARC node and switch to this local account when processing job data staging and job execution. This process is called *mapping*.

## 1.2.2 INSTALLATION

Install ARC CE core packages from repositories. [TODO: build from source link]. [TODO: metapackage name].

Grid security heavily relies on PKI and anything requires certificates/keys including ARC CE and users:

- for testing purposes Test-CA and host certificates signed by Test-CA are included [TODO: arcctl]
- for production usage please obtain certificate signed by one of the [IGTF accredited CA](#).

ARC CE needs IGTF CA certificates deployed to verify users and other services like storage elements. To deploy IGTF CA certificates to ARC CE host run<sup>1</sup>:

```
arcctl deploy igtf-ca classic
```

## 1.2.3 CONFIGURATION

Configuration of ARC CE is done by means of modifying pre-shipped 'zero configuration' available at `/etc/arc.conf`.

The purpose of this 'zero configuration' is to offer a minimalistic working computing element out-of-the box right after packages installation with zero additional configuration needed.

For production deployment you need to customize the configuration in accordance to your setup.

---

<sup>1</sup> Use `--installrepo` argument to enable repositories with IGTF CA certificates if ARC is not installed from the Nordugrid repos.

The most common configuration steps are the following:

## Configure AuthZ

AuthZ rules defines who can execute jobs on the computing element.

ARC CE authorization rules are configured with [authgroup] blocks.

In the shipped configuration the [authgroup: all] is defined that match any user.

Authgroup can be applied per-interface ([arex/ws/emies], [gridftpd/jobs]) and per-queue with allowaccess option in corresponding block.

### Example configuration 1

To authorize single (or several) person by certificate subject name (SN):

1. Create authorization group in `arc.conf` and specify SN directly with `subject` keyword or refers to a file that contains list of SNs:

```
[authgroup: staticdn]
subject = /O=Grid/O=Big VO/CN=Main Boss

[authgroup: dnfromfile]
file = /etc/grid-security/local_users
```

2. Apply authgroup to target interface of queue:

```
[gridftpd/jobs]
allowaccess = staticdn dnfromfile
```

### Example configuration 2

To filter access based on VOMS certificate attributes define one or more [authgroup] blocks using voms keyword.

To verify VO membership signature ARC CE needs so-called list of certificates (LSC) files that can be installed by `arcctl`.

Example configuration for atlas VO<sup>2</sup>:

1. Deploy LSC files:

```
arcctl deploy voms-lsc atlas --egi-vo
```

2. Create authorization group in `arc.conf`:

```
[authgroup: atlas]
voms = atlas * * *
```

3. Apply authgroup to target interface of queue:

```
[queue: atlas]
allowaccess = atlas
```

For more information about possible authgroup options, including LCAS integration please read ARC CE System Administrator manual.

<sup>2</sup> It this example and following there is simplified configuration, actual config in most cases includes different authgroups for different VO groups and roles.

### Configure mapping

Every grid-user should be mapped to local account to start processes and access files.

In shipped zero configuration all users are mapped to the same `nobody` account that will work with local forking only.

You have several common options to map grid-users.

### Accounts pool

The most transparent, secure and flexible recommended method is to map authorized users to accounts pool (so-called ARC simple pool method).

In this approach every authorized (by specified `[authgroup]`) user will be dynamically mapped to one of the available accounts.

Available pool account names are stored one per line in the `pool` file inside the directory. Leased names are stored in the other files placed in the same directory and can be reassigned to the other users after 10 days of inactivity.

*Example configuration for atlas:*

1. Create an account pool:

```
mkdir -p /etc/grid-security/pool/atlas
for u in atlas{001..100}; do echo $u >> /etc/grid-security/pool/atlas/pool;
↔done
```

2. Configure mapping in `arc.conf`<sup>3</sup>:

```
[mapping]
unixgroupmap=atlas simplepool /etc/grid-security/pool/atlas
```

### Legacy grid-mapfile based mapping

Legacy grid-mapfile based mapping is *not recommended* for the typical production loads.

In this approach users are mapped to local account based on certificate DN only. Mapping rules are stored line-by-line in so-called grid-mapfiles that describes which user is mapped to which account, for example:

```
"/O=Grid/O=NorduGrid/OU=uiio.no/CN=Aleksandr Konstantinov" user1
"/O=Grid/O=NorduGrid/OU=hep.lu.se/CN=Oxana Smirnova" user2
```

In the simplest legacy case ARC can use grid-mapfile for both authorization and mapping decisions instead of or in addition to `[authgroup]`-based.

Normally grid-mapfiles are referred in `arc.conf` as `[userlist]` objects that can be used as a source for `authZ` and mapping.

To generate mapfiles automatically and keeping it up to date (from e.g. VOMS database) `nordugridmap` utility can be used and configured with `[nordugridmap]` block.

### Using external LCMAPS rules

ARC can run external plugin to map users. To comply the several production loads ARC ships with build-in LCMAPS plugin.

LCMAPS itself should be installed and configured separately and beyond the scope of this guide. Consult ARC CE Sysadm Manual [TODO].

---

<sup>3</sup> atlas is the name used in `[authgroup: atlas]`

## Provide LRMS-specific information

One more critical configuration step is to supply ARC CE with relevant information regarding your LRMS specifics.

### Specify your LRMS type

In the `arc.conf` there is a dedicated `[lrms]` block that defines the type of your LRMS and several options related to the tuning behaviour. For example to instruct ARC to use SLURM use the following config:

```
[lrms]
lrms = slurm
slurm_use_sacct = yes
```

### Specify queues

In addition to specifying LRMS itself you should list all queues you want to expose via ARC CE using `[queue: name]` blocks.

### Configure A-REX

The ARC Resource-coupled EXecution service (A-REX) is a core service for the execution of compute jobs.

### Enable job management interfaces

A-REX has several job management interfaces available. You can control which of them are enabled configuring the corresponding blocks

**EMI-ES** `[arex/ws/emies]`

**RESTful** `[arex/rest]`

**Gridftp** `[gridftpd/jobs]`

**Internal** This interface is available implicitly

### Enable data services

ARC comes with powerful data-staging framework called DTR. [TODO: purpose of datastaging]

Define `[arex/data-staging]` to enable data-staging capabilities.

TODO: advantages of having cache TODO: choose and consider to share cachedir Configure `[arex/cache]`

### RunTime Environments

RunTime Environments can modify job execution cycle and used for advertising available software or features.

ARC ships several RTEs that already available to be used and classified as system-defined.

You can add extra directories with so-called user-defined RTEs using the `runtime_dir` configuration option in `[arex]` block.

In the ARC6 both system- and user-defined directories are local to ARC CE and *SHOULD NOT* be shared to worker nodes.

To use one of the installed RTEs you should additionally **enable** this RTE with *ARC Control Tool*. For example to enable system-defined ENV/PROXY RTE run:

```
arcctl rte enable ENV/PROXY
```

More details on operating RunTime Environments can be found in *Working with RunTime Environments in ARC6*.

### Information system

ARC CE information system aimed to collect and publish information to clients to be used for matchmaking and/or monitoring the state and stats of the resource.

It is mandatory to configure the information system for production case, like WLCG computing element.

### Defining general information

There are many information schemas and renderings of data available to comply any existing standards. There are several blocks that used to defined information depending on schemas:

**[infosys]** The most common block that enables internal information collection from ARC CE host and LRMS

**[infosys/cluster]** The common information about the whole cluster, including description of calculated total CPUs values.

**[queue: name]** For the heterogeneous clusters most of the information in the `[infosys/cluster]` block can be redefined on per-queue basis.

**[infosys/glue2]** Configures the GLUE2-specific values and enables internal glue2 rendering.

**[infosys/ldap]** Enables LDAP/BDII dedicated services to publish information via LDAP protocol.

**[infosys/glue2/ldap]** Enables GLUE2-schema LDAP rendering of the collected information.

**[infosys/nordugrid]** Enables LDAP rendering of the collected information according to the Nordugrid schema.

**[infosys/glue1]** Configures the GLUE1.x-schema specific values and enables LDAP rendering of GLUE1.x.

**[infosys/glue1/site-bdii]** Enables and configures GLUE1.x site-bdii functionality.

### Hinting clients about authorized VOs

TODO: describe advertizedvo per/cluster per/queue

### Accounting

ARC CE has build-in functionality to publish usage statistics to the SGAS and APEL centralized accounting services with the `jura` tool.

[TODO: some common flow and archiving necessity for republishing]

### Publishing to SGAS

TODO: example

### Publishing to APEL

TODO: example

## Additional ARC services for advanced use-cases

### Datadelivery service

TODO: description TODO: [arex] conf

### Candypond

TODO: description TODO: develop, ship, and describe here candypond RTE

### ACIX

TODO: describe ACIX (scanner, index, brocker)

## 1.2.4 CONFIGURE FIREWALL

Different ARC CE services opens a set of ports that should be allowed in the firewall configuration.

To generate iptables configuration based on `arc.conf` run:

```
arcctl deploy iptables-config
```

## 1.2.5 ENABLE AND RUN SERVICES

To enable and run all services as configured in `arc.conf` run:

```
arcctl service enable --as-configured --now
```

## 1.2.6 TEST BASIC FUNCTIONALITY

To test the job submission on the same host as A-REX the internal interface can be used:

```
arcsub -S org.nordugrid.internal to inject the jobs directly
```

To test the submission via any of regular interfaces you can use ARC clients on the other machine and run:

```
arcstat
arctest
arcsub
```

To diagnose the ARC CE service you can interact with `arcctl` in many different ways. Some common examples follows:

- Check which ARC services are enabled and running:

```
arcctl service list
```

- TODO:

```
arcctl job list
arcctl job attr 1s1MDm6kspnr005upx6UuPqABFKDmABFKDmPRIKDmABFKDmzyFsJm lrmsid
arcctl job log 1s1MDm6kspnr005upx6UuPqABFKDmABFKDmPRIKDmABFKDmzyFsJm
arcctl job log 1s1MDm6kspnr005upx6UuPqABFKDmABFKDmPRIKDmABFKDmzyFsJm --lrms
arcctl job log 1s1MDm6kspnr005upx6UuPqABFKDmABFKDmPRIKDmABFKDmzyFsJm --service
arcctl accounting stats
```

[TODO] Links to some production configs as examples? Publish some configs when we will create new ARC 6 configs from old ARC 5 for people.

## 1.3 ARC Configuration Reference Document

### 1.3.1 Configuration structure

This is the `arc.conf` REFERENCE DOCUMENT defining the configuration blocks and configuration options for the ARC services.

The `arc.conf` configuration file consists of the following blocks:

```
[common]
[authgroup:allowedusers]
[mapping]
[lrms]
[lrms/ssh]
[arex]
[arex/cache]
[arex/cache/cleaner]
[arex/data-staging]
[arex/ws]
[arex/ws/emies]
[arex/ws/cache]
[arex/ws/candypond]
[arex/ws/argus]
[arex/rest]
[arex/jura]
[arex/jura/archiving]
[arex/jura/sgas:neic_sgas]
[arex/jura/apel:egi_prod_apel]
[arex/ganglia]
[gridftpd]
[gridftpd/jobs]
[gridftpd/filedir]
[infosys]
[infosys/ldap]
[infosys/nordugrid]
[infosys/glue2]
[infosys/glue2/ldap]
[infosys/glue1]
[infosys/glue1/site-bdii]
[infosys/cluster]
[queue:gridlong]
[datadelivery-service]
[acix-scanner]
[acix-index]
[userlist:biusers]
[nordugridmap]
```

A block configures an ARC service, a service interface, a utility or a subsystem. Enabling (turning on) a functionality, a service or an interface requires the presence of the appropriate configuration block. To disable a service or an interface, simply delete or comment out the related `arc.conf` block (you may need to rerun the corresponding startup script).

As an example, in order to set up a minimalistic ARC CE offering no external interfaces you need to configure at least the `[common]`, `[mapping]`, `[arex]`, `[lrms]`, `[infosys]` and `[queue:name]` blocks.

As another example, an ARC-based data offloader would require the `[common]` and the `[datadelivery-service]` blocks.



A block is identified by its block header. A block header may consist of keywords and optionally block identifiers. Keywords may be separated by / and used to label subblocks (e.g. [arex/jura]), while block identifiers are separated by : from keywords. For example, in the [queue:short] block header 'queue' is a keyword while 'short' is an identifier, e.g. the name of the queue. Block headers must be UNIQUE.

A block starts with a unique [keyword:identifier] blockheader and ends where the next block starts, that is at the next [blockheader] directive.

A block may have sub-blocks e.g. the various interfaces of the AREX service are configured via sub-blocks (e.g. [arex/ws]). When a sub-block is enabled then the corresponding parent block must also appear in the arc.conf file.

Configuration blocks contain (config option, config value) pairs following the syntax in single line:

```
config_option=value element [optional value element]
```

**Note:** quotes around the configuration value(s) must NOT be used any longer.

**Note:** the arc.conf is CASE-SENSITIVE!

Space handling syntax in arc.conf for configuration lines:

```
(stripped space)option(stripped space)=(stripped space)value(saved_
↪space)(value)(stripped space)
```

and for block headers:

```
[keyword:(stripped space)space is allowed within identifier(stripped space)]
```

Detailed textual definition:

1. All trailing and leading spaces on each configuration line are stripped and ignored. This applies both to block headers and block content.
2. All spaces around the = sign in option=value kind of string (after 'a' is applied) are stripped and ignored. For example line hostname = myhost.info is treated as identical to hostname=myhost.info.
3. In block headers of [keyword] kind (after 'a' is applied) no additional spaces are allowed around keyword and inside keyword.
4. In block headers of [keyword:identifier] kind (after 'a' is applied) no additional spaces are allowed around keyword and inside keyword. The spaces ARE allowed around and inside identifier part. Those around identifier are stripped and ignored. Those inside identifier are treated as its integral part and are preserved.

Mandatory configuration options are indicated by an asterix prefix to the option name e.g: \*mandatory\_configuration. Mandatory options with undefined values will result in service stop during the startup process.

Each of the configuration options have well-defined default that is specified in this reference file. The default can take either a pre-set value, a special substitution or the keyword undefined. Configuration options within an enabled block take their default values in case they are missing (or commented out). Configuration parameters with undefined defaults takes no values. Furthermore, configuration options within disabled blocks takes no values either.

Configuration blocks related to authorization are ORDER-DEPENDENT! The authorization blocks [authgroup:name] MUST be defined before used in the blocks such as [mapping], [arex] or [gridftp/jobs]. The same rule applies to defining legacy [userlist:name] blocks. Furthermore, the order of the authorization blocks itself may have influence over authorization decisions!

## 1.3.2 Configuration blocks

### [common] block

Common configuration affecting all ARC components, usually related to networking or security or service behaviour. The common block options may be overridden by the specific sections of the components later. The [common] always appears at the beginning of the config file. The config options set within this block are available for all the other blocks thus shared by the different components of ARC.

#### hostname

*Synopsis:* hostname = string

*Description:* The FQDN of the frontend on which the ARC services are deployed.

*Default:* \$EXEC{hostname -f}

*Example:*

```
hostname=myhost.org
```

**Warning:** CHANGE1: modified semantics, not mandatory any longer

#### http\_proxy

*Synopsis:* http\_proxy = url

*Description:* The http proxy server. This setting affects all client HTTP(s) requests that initiated by ARC core services, including data staging, SAML communications, and pushing SGAS accounting records. This variable is similar to setting the ARC\_HTTP\_PROXY environmental variable.

*Default:* undefined

*Example:*

```
http_proxy=proxy.mydomain.org:3128
```

**Warning:** CHANGE4: moved here from the old [grid-manager] block

#### x509\_host\_key

*Synopsis:* x509\_host\_key = path

*Description:* (previously x509\_user\_key) Server credential location. Sets the full path to the host private key. These variables are similar to the GSI environment variable X509\_USER\_KEY. If indicated, the variable can be set individually for each service/component in the corresponding block.

*Default:* /etc/grid-security/hostkey.pem

*Example:*

```
x509_host_key=/etc/grid-security/hostkey.pem
```

**Warning:** CHANGE5: renamed

### x509\_host\_cert

*Synopsis:* x509\_host\_cert = path

*Description:* (previously x509\_user\_cert) Server credential location. Sets the full path to the host public certificate. These variables are similar to the GSI environment variable X509\_USER\_CERT. If indicated, the variable can be set individually for each service/component in the corresponding block.

*Default:* /etc/grid-security/hostcert.pem

*Example:*

```
x509_host_cert=/etc/grid-security/hostcert.pem
```

**Warning:** CHANGE6: renamed

### x509\_cert\_dir

*Synopsis:* x509\_cert\_dir = path

*Description:* Location of trusted CA certificates. This variable is similar to the GSI environment variable X509\_CERT\_DIR. If indicated, the variable can be set individually for each service/component in the corresponding block.

*Default:* /etc/grid-security/certificates

*Example:*

```
x509_cert_dir=/etc/grid-security/certificates
```

### x509\_voms\_dir

*Synopsis:* x509\_voms\_dir = path

*Description:* the path to the directory containing \*.lsc files needed for verification of VOMS service signature in the proxy-certificate.

*Default:* /etc/grid-security/vomsdir

*Example:*

```
x509_voms_dir=/etc/grid-security/vomsdir
```

### voms\_processing

*Synopsis:* voms\_processing = keyword

*Description:* Defines how to behave if errors in VOMS AC processing detected. The following keywords are supported:

**relaxed** use everything that passed validation.

**standard** same as relaxed but fail if parsing errors took place and VOMS extension is marked as critical. This is a default.

**strict** fail if any parsing error was discovered

**noerrors** fail if any parsing or validation error happened.

*Allowed values:* relaxed, standard, strict, noerrors

*Default:* standard

*Example:*

```
voms_processing=strict
```

### [authgroup:name] block

These configuration blocks contain authorization rules. These rules determine which authorization group a user belongs to. Then, the access control of ARC services are implemented via specifying the [authgroup:name] to which certain permissions are granted. For more info please read [Security Framework of ARC at `http://www.nordugrid.org/documents/arc-security-documentation.pdf`](http://www.nordugrid.org/documents/arc-security-documentation.pdf) The authgroup should not be mistaken for a virtual organisation (VO). An authgroup may match a single VO if only a single check (rule) on VO membership is performed. It is however more common to allow multiple VOs in a single authgroup.

**IMPORTANT:** Rules in an authgroup are processed in their order of appearance. The first matching rule decides the membership of the user to the authgroup being evaluated and the processing STOPS within that authgroup. This does not mean that the same user is not processed for the next authgroup: all [authgroup:name] blocks are evaluated, even if a user already has a match with one of the earlier groups.

There are positively and negatively matching rules. If a rule is matched positively then the user tested is accepted into the respective group and further processing is stopped. Upon a negative match the user would be rejected for that group - processing stops too. The sign of rule is determined by prepending the rule with + (for positive) or - (for negative) signs. + is default and can be omitted. A rule may also be prepended with ! to invert result of rule, which will let the rule match the complement of users. That complement operator (!) may be combined with the operator for positive or negative matching.

**Warning:** TODO: consider implementing Roger's suggestion and introduce a rule variable

All the objects used in the rules (e.g. other authgroups or userlists) MUST be defined before it may be used.

**Warning:** CHANGE31: renamed the block.

name authgroup\_name - This optional parameter specifies the name of the authgroup. This must be the same as the one in the [authgroup:name] block name. If this parameter not set then the name of the subblock is used instead, for example [authgroup:allowedusers]

**Warning:** CHANGE186: DELETED. use the name from the block header instead!

### subject

*Synopsis:* subject = certificate\_subject

*Description:* Rule to match specific subject of user's X.509 certificate. No masks, patterns and regular expressions are allowed.

This option in **multivalued**.

*Default:* undefined

*Example:*

```
subject=/O=Grid/O=Big VO/CN=Main Boss
subject=/O=Grid/O=Big VO/CN=Deputy Boss
```

## file

*Synopsis:* file = path

*Description:* Processes a list of DNs stored in an external file one per line and adds those to the authgroup.

This option in **multivalued**.

*Default:* undefined

*Example:*

```
file=/etc/grid-security/local_users
file=/etc/grid-security/atlas_users
```

**Warning:** CHANGE32: Modified semantics. The external file should only contain DNs, no complex rules. No need to change the code.

## voms

*Synopsis:* voms = vo group role capabilities

*Description:* Match VOMS attribute in user's credential. Use \* to match any value.

This option in **multivalued**.

*Default:* undefined

*Example:*

```
voms=nordugrid Guests * *
voms=atlas students prodman *
```

## authgroup

*Synopsis:* authgroup = group\_name [group\_name ...]

*Description:* (previously group) Match user already belonging to one of specified authgroups. The authgroup referred here must be defined earlier in configuration file. Multiple authgroup names may be specified for this rule. That allows creating hierarchical structure of authorization groups like 'clients' are those which are 'users' and 'admins'.

This option in **multivalued**.

*Default:* undefined

*Example:*

```
authgroup=local_admins
authgroup=local_admins remote_users
```

**Warning:** CHANGE33: renamed

## userlist

*Synopsis:* userlist = ulist\_name [ulist\_name ...]

*Description:* (previously vo) Match user belonging to ulist\_name defined in an earlier [userlist:ulist\_name] block. Multiple userlist names are allowed for this rule.

This option in **multivalued**.

*Default:* undefined

*Example:*

```
userlist=biouers
```

**Warning:** CHANGE34: renamed

## plugin

*Synopsis:* plugin = timeout path [argument ...]

*Description:* Run external executable or function from shared library. Rule is matched if plugin returns 0. In arguments following substitutions are supported:

%D - subject of certificate %P - path to proxy

ARC ships with LCAS plugin that can be enabled with following plugin configuration. For more information about configuring LCAS itself see section 'Using LCAS/LCMAPS' in the ARC Computing Element System Administrator Guide.

This option in **multivalued**.

*Default:* undefined

*Example:*

```
plugin=10 /usr/libexec/arc/arc-lcas %D %P liblcas.so /usr/lib64 /etc/lcas/lcas.db
```

lcas library directory database - Call LCAS functions to check rule.

**Warning:** CHANGE35: DELETED. The same can be done with the plugin parameter

remote URL ... - Check user's credentials against remote service. Only

**Warning:** CHANGE36: DELETED. Feature is not used for ages. Code simplification.

all - Matches any user identity. This variable requires no arguments or values.

*Default:* undefined

*Example:*

```
all=
```

## [mapping] block

This block defines the grid-identity to local unix identity mapping rules used by various ARC components.

**Warning:** CHANGE8: This is a new block

## gridmap

*Synopsis:* gridmap = path

*Description:* The gridmap file location. gridmap-based mapping is not recommended any longer.

*Default:* /dev/null

*Example:*

```
gridmap=/etc/grid-security/grid-mapfile
```

## unixmap

*Synopsis:* unixmap = [unixname][:unixgroup] rule

*Description:* A more sophisticated way to map Grid identity of client to local account. If client matches 'rule' it's assigned specified unix identity or one generated by rule. Mapping commands are processed sequentially and processing stops at first successful one (like in [authgroup] section). For possible rules read ARC Computing Element. System Administrator guide manual. All rules defined in [authgroup] section can be used. There are also additional rules which produce not only yes/no result but also give back user and group names to which mapping should happen. The way it works is quite complex so it is better to read full documentation. For safety reasons if sophisticated mapping is used it is better to finish mapping sequence with default mapping to nonexistent or safe account. ARC ships with LCMAPS plugin that can be enabled with following unixmap configuration. For more information about configuring LCMAPS itself see section 'Using LCAS/LCMAPS' in the full documentation.

This option in **multivalued**.

*Default:* undefined

*Example:*

```
unixmap=nobody:nobody all
unixmap=* mapplugin 30 /usr/libexec/arc/arc-lcmaps %D %P liblcmaps.so /usr/lib64 /
↳etc/lcmaps/lcmaps.db arc
```

## unixgroupmap

*Synopsis:* unixgroupmap = authgroup rule

*Description:* (previously unixgroup) Mapping rule only for users belonging to

specified authorization 'authgroup'. It is similar to an additional filter for unixmap command which filters out all users not belonging to specified authorization group. Only rules which generate unix user and group names may be used in this command. Please read ARC Computing Element System Administrator Guide for more information.

This option in **multivalued**.

*Default:* undefined

*Example:*

```
unixgroupmap=users simplepool /etc/grid-security/pool/users
```

**Warning:** CHANGE170: RENAMED

## unixlistmap

*Synopsis:* unixlistmap = userlist\_name rule

*Description:* (previously unixvo) Mapping rule only for users belonging to specified userlist defined via the [userlist:name] block. Only rules which generate unix identity name may be used in this command. Please read ARC Computing Element. System Administrator Guide for more information. This command is similar to 'unixgroupmap' described above and exists for convenience for setups which base mapping on userlists.

This option in **multivalued**.

*Default:* undefined

*Example:*

```
unixlistmap=ATLAS unixuser atlas:atlas
```

**Warning:** CHANGE9: RENAMED. modified option: It uses the name from the [userlist:name] block

## [userlist:name] block

**Warning:** TODO: move to the end just before the [nordugridmap]

The [userlist:name] blocks are used to define userlists and configure how those are generated by the nordugridmap utility, including the optional user mapping information. The userlist is identified with the generated file that is stored in the outfile. Please note that behaviour of the nordugridmap external utility can be further modified by the optional [nordugridmap] block (see next block). Note that the [userlist:name] block by itself does not affect authorization. In order to define auth rules userlists can be referenced within the [authgroup] blocks by the userlist=name parameter. Also, the generated outfile can be used via the 'file' auth rule of the [authgroup] block. The order of this block in arc.conf may influence authorization decision. This block must appear before the [arex] block.

**Warning:** CHANGE17: new blockname that better reflects the purpose of the config block. This block is not defining any VO!

id blockid - specifies the unique configuration block id (this does not affect nordugridmap utility)

**Warning:** CHANGE18: DELETED should be removed from code and config now that the userlist blocks have unique names

vo vo\_name - This optional parameter specifies the name of the userlist. This must be the same as the one in the [userlist:name] block name.

**Warning:** CHANGE19: DELETED. use the name from the block header instead!

## outfile

*Synopsis:* outfile = path



*Description:* (previously file) The full path of the GENERATED file that contains the userlist (with optional mapping info). If the same file specified as output for different [userlist:name] blocks then nordugridmap will automatically merge entries following the order of the blocks.

*Default:* \$VAR{ [common/mapping] gridmap }

*Example:*

```
outfile=/etc/grid-security/lists/atlas-users
```

**Warning:** CHANGE20: renamed from file

## source

*Synopsis:* \*source = url

*Description:* the URL of the VO database which is used to generate the userlist. The nordugridmap will use this URL to automatically generate and keep up-to-date userlist (mapfile) specified by the 'outfile' attribute.

url is a multivalued attribute, several sources can be specified and all the users from those sources will be merged into the same file. The source URLs are processed in the given order.

Currently supported URL types are:

**http(s)://** URL to plain text file. File should contain a list of DNs.

**voms(s)://** URL to VOMS-Admin interface

**file://** local file (stand-alone or dynamically generated by nordugridmap). File should contain a list of DNs with optional mapped unixid: user DN [mapped user ID] Result of optional mapped unixid processing depend on mapuser\_processing option settings.

**userlist://** reference to another [userlist/name] configuration block

**nordugrid** add Nordugrid VO members

You can use either userlist:// or file:// entries to specify dependencies between [userlist/name] blocks, but using userlist:// is a recommended way.

For each separate source URL it is possible to override some parameters value. You can use the following syntax to perform this:

```
"source=URL < parameter1=value1 parameter2=value2"
```

You can override the following parameters:

**mapped\_unixid** for http(s),voms(s),ldap and file URLs

**cache\_enable** for http(s),voms(s),ldap and file URLs

**voms\_method** for voms(s) URLs

**mapuser\_processing** for file URLs with mapped\_unixid='<unixid>' overridden (control mapped\_unixid overriding behaviour for URL)

This option is **multivalued**.

*Default:* undefined

*Example:*

```
source=vomss://voms.ndgf.org:8443/voms/nordugrid.org
source=vomss://lcg-voms.cern.ch:8443/voms/atlas?/atlas/Role=VO-Admin < mapped_
->unixid=atlasadmin
source=vomss://kuiken.nikhef.nl:8443/voms/gin.ggf.org < voms_method=get
source=http://www.nordugrid.org/developers.dn
```

```
source=file:///etc/grid-security/privileged_users.dn
source=userlist://biusers
source=nordugrid
```

**Warning:** CHANGE21: MODIFIED options: edg-mkgridmap source option is removed (obsolete technology); vo source option is renamed as userlist.

## mapped\_unixid

*Synopsis:* mapped\_unixid = unixid

*Description:* The local UNIXID which is optionally used in the generated outfile by the nordugridmap utility.

If any of the sources have already provided mapping information (`file://` or `userlist://`) behavior depends on ‘mapuser\_processing’ from the [nordugridmap] block:

**mapuser\_processing = overwrite** ignore already provided mapping and apply mapped\_unixid for all sources

**mapuser\_processing = keep** apply mapped\_unixid only for sources that does not already has mapping information

If the mapped\_unixid config parameter is not specified or has empty value, then behavior depends on the value of allow\_empty\_unixid from the [nordugridmap] block:

**allow\_empty\_unixid = yes** empty value will be used for mapped\_unixid which means that nordugridmap will generate only the list of DNs without mapping (consider using mapuser\_processing = overwrite along with this option or sources that does not provide previously defined mapping information)

**allow\_empty\_unixid = no** then nordugridmap will skip users without mapping information

*Default:* nobody

*Example:*

```
mapped_unixid=gridtest
```

**Warning:** CHANGE22: Modified. The empty value should work the same way as missing parameter. set the default to nobody

## voms\_fqan\_map

*Synopsis:* voms\_fqan\_map = fqan unixid

*Description:* The local UNIXID which is used to map voms(s) sources with specific FQAN given.

Several voms\_fqan\_map can be specified for a [userlist/name] block. For each voms(s) sources in [userlist/name] block and every voms\_fqan\_map record separate source record will be automatically generated with mapped\_unixid overwritten to specified one.

Sources are generated in a given voms\_fqan\_map order. Original voms(s) source URL are processed LAST.

This allows to simplify configuration, especially in redundancy cases when several VOMS servers are used for the same VO.

This option in **multivalued**.

*Default:* undefined

*Example:*

```
voms_fqan_map=/atlas/Role=VO-Admin atlasadmin
voms_fqan_map=/atlas/Role=production atlasprod
```

require\_issuerdn yes/no - YES would map only those DNs obtained from the urls...

**Warning:** CHANGE23: DELETED. no valid use-case for this feature any longer.

## filter

*Synopsis:* filter = ACL string

*Description:* An ACL filter for the nordugridmap utility. Multiple allow/deny statements are possible. The fetched DNs are filtered against the specified rules before they are added to the generated outfile.

\* can be used as a wildcard. You may run the nordugridmap with the `--test` command line option to see how the filters you specified work.

If at least one allow filter is specified implicit deny is used at the end of ACL. If only deny filters are present - implicit allow used at the end.

This option in **multivalued**.

*Default:* undefined

*Example:*

```
filter=deny *infn*
filter=allow *Nordugrid*
```

## [lrms] block

This block specifies the characteristics of the Local Resource Manager System (batch system) underneath the ARC CE. This block contains all the lrms-specific parameters and information. Configuration values in this block are available for A-REX, the backends, accounting and infosys ARC subsystems.

ARC support the most common LRMS flavours. For detailed description of recommended LRMS setups for ARC please refer to ARC CE sysadmin guide: <http://www.nordugrid.org/documents/arc-ce-sysadm-guide.pdf>

**Warning:** CHANGE10: This is a new block. Contains parameters previously set in [common], [infosys], [cluster], [queue]

## lrms

*Synopsis:* lrms = lrmstype [defaultqueue]

*Description:* Sets the type of the LRMS (queue system) and optionally the default queue name. **ONLY ONE LRMS IS ALLOWED. MULTIPLE lrms ENTRIES WILL TRIGGER UNEXPECTED BEHAVIOUR.**

For lrmstype, the following values can be chosen:

- fork - simple forking of jobs to the same node as the server
- sge - (Sun/Oracle) Grid Engine
- condor - Condor
- pbs - PBS (covers all the PBS flavours e.g. OpenPBS, PBSPro, ScalablePBS, Torque)

- lsf - LSF
- ll - LoadLeveler
- slurm - SLURM
- boinc - Boinc
- slurmpy - new generation SLURM scripts (provide access to slurm via ssh too)

PBS has many flavours, ARC currently supports OpenPBS, PBSPro, ScalablePBS and Torque (the official name for ScalablePBS). There is no need to specify the flavour or the version number of the PBS, simply write 'pbs'. Similarly, there is no need to specify (Sun/Oracle) Grid Engine versions and flavours.

The optional `defaultqueue` parameter specifies the name of an existing LRMS queue in the cluster that will be used by AREX as the default queue to submit grid jobs in case the job descriptions submitted to the ARC CE do not specify a requested queue. This queue name must match one of the `[queue:queue_name]` blocks.

*Allowed values:* fork, sge, condor, pbs, lsf, ll, slurm, boinc, slurmpy

*Default:* fork

*Example:*

```
lrms=pbs gridlong
lrms=slurm
```

**Warning:** CHANGE11: new lrms type values added (slurmpy).

### lrmsconfig

*Synopsis:* lrmsconfig = text

*Description:* An optional free text field to describe the configuration of your Local Resource Management System (batch system). The value is published in the infosys, and is not used otherwise.

*Default:* undefined

*Example:*

```
lrmsconfig=single job per processor
```

**Warning:** CHANGE12: moved here from the `[cluster]` block

### defaultmemory

*Synopsis:* defaultmemory = number

*Description:* The LRMS memory request of job to be set by the LRMS backend scripts, if a user submits a job without specifying how much memory should be used. The order of precedence is: job description -> default-memory. This is the amount of memory (specified in MB) that a job will request.

*Default:* undefined

*Example:*

```
defaultmemory=512
```

**Warning:** CHANGE13: moved here from the [cluster] block

### nodename

*Synopsis:* nodename = path

*Description:* Redefine the command to obtain hostname of LRMS worker node. By default the value is defined on buildtime and depend on the OS. In most cases /bin/hostname -f will be used.

*Default:* undefined

*Example:*

```
nodename=/bin/hostname -s
```

**Warning:** CHANGE: moved here from the [arex] block

### gnu\_time

*Synopsis:* gnu\_time = path

*Description:* Path to the GNU time command on the LRMS worker nodes. If time command exists on the node, jobscript will write additional diagnostic information.

*Default:* /usr/bin/time

*Example:*

```
gnu_time=/usr/bin/time
```

**Warning:** CHANGE: moved here from the [arex] block

### pbs\_bin\_path

*Synopsis:* pbs\_bin\_path = path

*Description:* The path to the qstat,pbsnodes,qmgr etc PBS binaries, no need to set if PBS is not used

*Default:* /usr/bin

*Example:*

```
pbs_bin_path=/usr/bin
```

### maui\_bin\_path

*Synopsis:* maui\_bin\_path = path

*Description:* Sets the path of the maui commands like showbf.

**Warning:** CHANGE: REMOVED, not used any longer

### pbs\_log\_path

*Synopsis:* pbs\_log\_path = path

*Description:* The path of the PBS server logfiles which are used by A-REX to determine whether a PBS job is completed. If not specified, A-REX will use qstat for that.

*Default:* /var/spool/pbs/server\_logs

*Example:*

```
pbs_log_path=/var/spool/pbs/server_logs
```

### pbs\_dedicated\_node\_string

*Synopsis:* pbs\_dedicated\_node\_string = string

*Description:* (previously `dedicated_node_string`) The string which is used in the PBS node config to distinguish the grid nodes from the rest. Suppose only a subset of nodes are available for grid jobs, and these nodes have a common `node property string`, this case the string should be set to this value and only the nodes with the corresponding `pbs node property` are counted as grid enabled nodes. Setting the `dedicated_node_string` to the value of the `pbs node property` of the grid-enabled nodes will influence how the `totalcpus`, `user freecpus` is calculated. You don't need to set this attribute if your cluster is fully available for the grid and your cluster's PBS config does not use the `node property` method to assign certain nodes to grid queues. You shouldn't use this config option unless you make sure your PBS config makes use of the above described setup.

*Default:* undefined

*Example:*

```
pbs_dedicated_node_string=gridnode
```

**Warning:** CHANGE14: renamed. moved here from [cluster]

### condor\_bin\_path

*Synopsis:* condor\_bin\_path = path

*Description:* Path to Condor binaries. Must be set if Condor is used.

*Default:* /usr/bin

*Example:*

```
condor_bin_path=/opt/condor/bin
```

### condor\_config

*Synopsis:* condor\_config = path

*Description:* Full path to Condor config file. Must be set if Condor is used and the config file is not in its default location (`/etc/condor/condor_config` or `~/condor/condor_config`). The full path to the file should be given.

*Default:* /etc/condor/condor\_config

*Example:*

```
condor_config=/opt/condor/etc/condor_config
```

## condor\_rank

*Synopsis:* condor\_rank = ClassAd\_float\_expression

*Description:* If you are not happy with the way Condor picks nodes when running jobs, you can define your own ranking algorithm by optionally setting the condor\_rank attribute. condor\_rank should be set to a ClassAd float expression that you could use in the Rank attribute in a Condor job description.

*Default:* undefined

*Example:*

```
condor_rank=(1-LoadAvg/2)*(1-LoadAvg/2)*Memory/1000*KFlops/1000000
```

## condor\_requirements

*Synopsis:* condor\_requirements = string

*Description:* It needs to be defined for each Condor queue. Use this option to determine which nodes belong to the current queue. The value of condor\_requirements must be a valid constraints string which is recognized by a condor\_status -constraint ... command. It can reference pre-defined ClassAd attributes (like Memory, Opsy, Arch, HasJava, etc) but also custom ClassAd attributes. To define a custom attribute on a condor node, just add two lines like the ones below in the \$(hostname).local config file on the node:

```
NORDUGRID_RESOURCE=TRUE
STARTD_EXPRS = NORDUGRID_RESOURCE, $(STARTD_EXPRS)
```

A job submitted to this queue is allowed to run on any node which satisfies the condor\_requirements constraint. If condor\_requirements is not set, jobs will be allowed to run on any of the nodes in the pool. When configuring multiple queues, you can differentiate them based on memory size or disk space, for example.

*Default:* undefined

*Example:*

```
condor_requirements=(OpSys == "linux" && NORDUGRID_RESOURCE && Memory >= 1000 &&
↵Memory < 2000)
```

## sge\_bin\_path

*Synopsis:* sge\_bin\_path = path

*Description:* Path to Sun Grid Engine (SGE) binaries, Default is search for qsub command in the shell PATH

*Default:* undefined

*Example:*

```
sge_bin_path=/opt/nlge6/bin/lx24-x86
```

## sge\_root

*Synopsis:* sge\_root = path

*Description:* Path to SGE installation directory. MUST be set if SGE is used.

*Default:* /gridware/sge

*Example:*

```
sge_root=/opt/nlge6
```

### sge\_cell

*Synopsis:* sge\_cell = name

*Description:* The name of the SGE cell to use. This option is only necessary in case SGE is set up with a cell name different from 'default'

*Default:* default

*Example:*

```
sge_cell=default
```

### sge\_qmaster\_port

*Synopsis:* sge\_qmaster\_port = port

*Description:* The SGE port options should be used in case SGE command line clients require SGE\_QMASTER\_PORT and SGE\_EXECD\_PORT environment variables to be set. Usually they are not necessary.

*Default:* undefined

*Example:*

```
sge_qmaster_port=536
```

### sge\_execd\_port

*Synopsis:* sge\_execd\_port = port

*Description:* The SGE port options should be used in case SGE command line clients require SGE\_QMASTER\_PORT and SGE\_EXECD\_PORT environment variables to be set. Usually they are not necessary.

*Default:* undefined

*Example:*

```
sge_execd_port=537
```

### sge\_jobopts

*Synopsis:* sge\_jobopts = string

*Description:* Additional SGE options to be used when submitting jobs to SGE

*Default:* undefined

*Example:*

```
sge_jobopts=-P atlas -r yes
```



### slurm\_bin\_path

*Synopsis:* slurm\_bin\_path = path

*Description:* Path to SLURM binaries, must be set if installed outside of normal PATH

*Default:* /usr/bin

*Example:*

```
slurm_bin_path=/usr/bin
```

### slurm\_wakeupperiod

*Synopsis:* slurm\_wakeupperiod = numsec

*Description:* How long should infosys wait before querying SLURM for new data (seconds)

*Default:* 30

*Example:*

```
slurm_wakeupperiod=15
```

### slurm\_use\_sacct

*Synopsis:* slurm\_use\_sacct = yes/no

*Description:* Indicates whether ARC should use sacct instead of scontrol to obtain information about finished jobs  
Not supported by slurmpy.

*Allowed values:* yes, no

*Default:* yes

*Example:*

```
slurm_use_sacct=yes
```

### lsf\_bin\_path

*Synopsis:* lsf\_bin\_path = path

*Description:* The PATH to LSF bin folder

*Default:* /usr/bin

*Example:*

```
lsf_bin_path=/usr/local/lsf/bin/
```

### lsf\_profile\_path

*Synopsis:* lsf\_profile\_path = path

*Description:* Path to the profile.lsf file. Infoprovider scripts will source profile.lsf to setup LSF utilities environment.

*Default:* /usr/share/lsf/conf/profile.lsf

*Example:*

```
lsf_profile_path=/usr/local/share/lsf/conf/profile.lsf
```

### lsf\_architecture

*Synopsis:* lsf\_architecture = string

*Description:* CPU architecture to request when submitting jobs to LSF. Use only if you know what you are doing.

*Default:* undefined

*Example:*

```
lsf_architecture=PowerPC
```

**Warning:** CHANGE15: moved here from [queue:name] block

### ll\_bin\_path

*Synopsis:* ll\_bin\_path = path

*Description:* The PATH to the LoadLeveler bin folder

*Default:* /usr/bin

*Example:*

```
ll_bin_path=/opt/ibml1/LoadL/full/bin
```

### ll\_consumable\_resources

*Synopsis:* ll\_consumable\_resources = yes/no

*Description:* Indicates whether the LoadLeveler setup is using Consumable Resources.

*Allowed values:* yes, no

*Default:* no

*Example:*

```
ll_consumable_resources=yes
```

### boinc\_db\_host

*Synopsis:* boinc\_db\_host = hostname

*Description:* Connection strings for the boinc database: host

*Default:* localhost

*Example:*

```
boinc_db_host=localhost
```

### boinc\_db\_port

*Synopsis:* boinc\_db\_port = port

*Description:* Connection strings for the boinc database: port

*Default:* 3306

*Example:*

```
boinc_db_port=3306
```

### boinc\_db\_name

*Synopsis:* boinc\_db\_name = db\_name

*Description:* Connection strings for the boinc database: db\_name

*Default:* undefined

*Example:*

```
boinc_db_name=myproject
```

### boinc\_db\_user

*Synopsis:* boinc\_db\_user = user

*Description:* Connection strings for the boinc database: db\_user

*Default:* undefined

*Example:*

```
boinc_db_user=boinc
```

### boinc\_db\_pass

*Synopsis:* boinc\_db\_pass = pwd

*Description:* Connection strings for the boinc database: pwd

*Default:* undefined

*Example:*

```
boinc_db_pass=password
```

### dgbridge\_stage\_dir

*Synopsis:* dgbridge\_stage\_dir = path

*Description:* Desktop Bridge www publish dir.

**Warning:** CHANGE: DELETED

### dgbridge\_stage\_prepend

*Synopsis:* dgbridge\_stage\_prepend = url

*Description:* Desktop Bridge url prefix pointing to dgbridge\_stage\_dir.

**Warning:** CHANGE: DELETED

### [lrms/ssh] block

This sub-block configures the ssh environment for remote batch management. Currently ONLY the python slurm (slurmpy) EXPERIMENTAL batch module is able to utilize the remote ssh feature. Parameters within this block are relevant if the cluster frontend is remotely located wrt. CE frontend (machine running A-REX). If specified with the parameters below, the session, cache and runtime directories will be mounted from the cluster frontend on the CE frontend using sshfs. Job submission and management will done using ssh (Paramiko).

**Warning:** CHANGE16: new sub-block

### remote\_host

*Synopsis:* \*remote\_host = hostname

*Description:* Define the remote cluster frontend machine which contains session, cache and runtime directories and where jobs should be submitted to.

*Default:* undefined

mandatory

*Example:*

```
remote_host=myremotehost.org
```

### private\_key

*Synopsis:* \*private\_key = path

*Description:* Location of the private key which should be used establish connection to the machine specified by the 'remote\_host' attribute.

*Default:* undefined

mandatory

*Example:*

```
private_key=/etc/grid-security/hostkey-priv.pem
```

### remote\_user

*Synopsis:* \*remote\_user = user

*Description:* User on remote cluster which should be used for mounting directories, submitting and managing jobs, and gathering information about cluster.

*Default:* undefined

mandatory

*Example:*

```
remote_user=grid
```

### remote\_sessiondir

*Synopsis:* remote\_sessiondir = path

*Description:* Session directory on cluster frontend to be mounted (sshfs) on CE machine at directory specified by the 'sessiondir' attribute in the [arex] block.

*Default:* undefined

mandatory

*Example:*

```
remote_sessiondir=/scratch/grid
```

### remote\_runtimedir

*Synopsis:* remote\_runtimedir = path

*Description:* Runtime environment directory on cluster frontend to be mounted (sshfs) on CE machine at directory specified by the 'runtimedir' attribute in the [arex] block.

**Warning:** TODO: check if it is still relevant or work with the ARC6 RTE framework

*Default:* undefined

*Example:*

```
remote_runtimedir=/SOFTWARE/runtime
```

### remote\_cachedir

*Synopsis:* \*remote\_cachedir = path

*Description:* Cache directory on cluster frontend to be mounted (sshfs) on CE machine at directory specified by the 'cachedir' attribute in the [arex] block.

**Warning:** TODO: why is it mandatory and why the remote\_sessiondir is not mandatory?

*Default:* undefined

*Example:*

```
remote_cachedir=/scratch/cache
```

### ssh\_timeout

*Synopsis:* ssh\_timeout = time

*Description:* Timeout of establishing ssh connection. Unit in seconds.

*Default:* 10

*Example:*

```
ssh_timeout=10
```

### [arex] block

The [arex] block, together with its various subblocks, configures the A-REX service hosted in arched. A-REX takes care of various middleware tasks on the frontend such as job creation and management, stagein/stageout, LRMS job submission, data caching, etc. . .

**Warning:** CHANGE173: renamed block

### user

*Synopsis:* user = user[:group]

*Description:* Switch to a non root user/group after startup. Use with caution because of limited functionality when arex is not run under root.

*Default:* root

*Example:*

```
user=grid:grid
```

### norootpower

*Synopsis:* norootpower = yes|no

*Description:* If set to yes, all job management processes will switch to mapped user's identity while accessing session directory. This is useful if session directory is on NFS with root squashing turned on.

*Allowed values:* yes, no

*Default:* no

*Example:*

```
norootpower=yes
```

### delegationdb

*Synopsis:* delegationdb = db\_name

*Description:* specify which DB to use to store delegations. Currently supported db\_names are bdb and sqlite

*Default:* sqlite

*Example:*

```
delegationdb=sqlite
```

**Warning:** CHANGE171: modified. new default.

### watchdog

*Synopsis:* watchdog = yes/no

*Description:* Specifies if additional watchdog processes is spawned to restart main process if it is stuck or dies.

*Allowed values:* yes, no

*Default:* no

*Example:*

```
watchdog=no
```

### loglevel

*Synopsis:* loglevel = level

*Description:* (previously debug) Set loglevel of the arched daemon hosting A-REX service between 0 (FATAL) and 5 (DEBUG). Defaults to 3 (INFO).

*Allowed values:* 0, 1, 2, 3, 4, 5

*Default:* 3

*Example:*

```
loglevel=3
```

**Warning:** CHANGE37: renamed

### logfile

*Synopsis:* logfile = path

*Description:* Specify A-REX log file location. If using an external log rotation tool be careful to make sure it matches the path specified here.

*Default:* /var/log/arc/arex.log

*Example:*

```
logfile=/var/log/arc/arex.log
```

**Warning:** CHANGE189: modified default (renamed file)

## joblog

*Synopsis:* `joblog = path`

*Description:* Specifies where to store specialized log about started and finished jobs. If path is empty log is NOT written. Controlled by `logrotate` if default name is kept. This log is not used by any other part of ARC so can be safely disabled if you are not interested in storing jobs log.

*Default:* `/var/log/arc/arex-jobs.log`

*Example:*

```
joblog=""
```

`logsize size [number]` - 'Size' specifies in bytes how big log file is

**Warning:** CHANGE38: DELETED. This functionality is not used.

`logreopen yes|no` - Specifies if log file must be closed after each record is added.

**Warning:** CHANGE39: DELETED. not really used. hide it from the `arc.conf` template. Can stay as a hidden feature.

## fixdirectories

*Synopsis:* `fixdirectories = yes|missing|no`

*Description:* Specifies during startup A-REX should create all directories needed for its operation and set suitable default permissions. If `no` is specified then A-REX does nothing to prepare its operational environment. In case of `missing` A-REX only creates and sets permissions for directories which are not present yet. For `yes` all directories are created and permissions for all used directories are set to default safe values.

*Allowed values:* `yes,missing,no`

*Default:* `yes`

*Example:*

```
fixdirectories=yes
```

## controldir

*Synopsis:* `controldir = path`

*Description:* The directory of the A-REX's internal job metadata files. For a heavy loaded computing elements you can consider to locate `controldir` on a dedicated partition optimized for small random reads and writes. The directory is not needed on the nodes.

*Default:* `/var/spool/arc/jobstatus`

*Example:*

```
controldir=/var/spool/arc/jobstatus
```



## sessiondir

*Synopsis:* sessiondir = path [drain]

*Description:* the directory which holds the sessiondirs of the grid jobs. Multiple session directories may be specified. In this case jobs are spread evenly over the session directories. If sessiondir=\* is set, the session directory will be spread over the \${HOME}/.jobs directories of every locally mapped unix user. It is preferred to use common session directories. The path may be followed by drain, in which case no new jobs will be assigned to that sessiondir, but current jobs will still be processed and accessible.

This option is **multivalued**.

*Default:* /var/spool/arc/sessiondir

*Example:*

```
sessiondir=/scratch/arcsessions drain
sessiondir=*
```

## defaultttl

*Synopsis:* defaultttl = [ttl [ttr]]

*Description:* The ttl parameter sets the time in seconds for how long a job session directory will survive after job execution has finished. If not specified the default is 1 week. The ttr parameter sets how long information about a job will be kept after the session directory is deleted. If not specified, the ttr default is one month.

*Default:* 604800 2592000

*Example:*

```
defaultttl=2592000
```

## shared\_filesystem

*Synopsis:* shared\_filesystem = yes/no

*Description:* Specifies if computing nodes can access folders mounted with protocols like NFS with the same pathnames as the frontend. Note that the default 'yes' assumes that the paths to the session directories are the same on both frontend and nodes. If these paths are not the same, then one should set the scratchdir option. The option changes the RUNTIME\_NODE\_SEES\_FRONTEND variable in the submission scripts.

*Allowed values:* yes, no

*Default:* yes

*Example:*

```
shared_filesystem=yes
```

## scratchdir

*Synopsis:* scratchdir = path

*Description:* The path on computing node to move session directory to before execution. If defined should contain the path to the directory on the computing node which can be used to store a jobs' files during execution. Sets the environment variable RUNTIME\_LOCAL\_SCRATCH\_DIR. If the variable is not set, then the session dir is not moved before execution. Don't set this parameter unless you want to move the sessiondir to scratchdir on the node.

*Default:* undefined

*Example:*

```
scratchdir=/local/scratch/
```

### shared\_scratch

*Synopsis:* shared\_scratch = path

*Description:* The path on frontend where scratchdir can be found. If defined should contain the path corresponding to that set in scratchdir as seen on the frontend machine. Sets the environment variable RUNTIME\_FRONTEND\_SEES\_NODE.

*Default:* undefined

*Example:*

```
shared_scratch=/mnt/scratch
```

### tmpdir

*Synopsis:* tmpdir = path

*Description:* A temporary directory used by A-REX.

*Default:* /tmp

*Example:*

```
tmpdir=/tmp
```

### runtimedir

*Synopsis:* runtimedir = path

*Description:* The directory which holds the additional runtimeenvironment scripts, added by system administrator. Several directories can be specified. To enable RTEs to be advertised in the information system and used during submission the arcctl tool should be used.

This option is **multivalued**.

*Default:* undefined

*Example:*

```
runtimedir=/var/spool/arc/extraruntimes  
runtimedir=/cvmfs/vo/arcruntime
```

### maxjobs

*Synopsis:* maxjobs = number1 number2 number3 number4 number5

*Description:* specifies maximum allowed number of jobs. number1 - jobs which are not in FINISHED state (jobs tracked in RAM) number2 - jobs being run (SUBMITTING, INLRMS states) number3 - jobs being processed per DN number4 - jobs in whole system number5 - LRMS scripts limit (jobs in SUBMITTING and CANCELING) A parameter set to -1 means no limit.

*Default:* -1 -1 -1 -1 -1

*Example:*

```
maxjobs=10000 10 2000 -1 -1
```

**Warning:** CHANGE40: Modified. Explicitly indicate no limit with -1. Missing number should not be allowed.

### maxrerun

*Synopsis:* maxrerun = number

*Description:* Specifies how many times job can be rerun if it failed in LRMS. This is only an upper limit, the actual rerun value is set by the user in his xrsl.

*Default:* 5

*Example:*

```
maxrerun=5
```

### statecallout

*Synopsis:* statecallout = state options plugin\_path

*Description:* (previously authplugin) This parameter enables a callout feature of A-REX: every time job goes to state A-REX will run plugin\_path executable. Options consist of key=value pairs separated by , . Possible keys are:

**timeout** wait for result no longer that 'value' seconds, (timeout= can be omitted).

**onsuccess, onfailure, ontimeout** what to do if plugin exited with exit code 0, not 0, timeout achieved.

Possible actions are:

**pass** continue executing job,

**fail** cancel job,

**log** write to log fail about problem and continue executing job.

This option in **multivalued**.

*Default:* undefined

arc-blahp-logger: a statecallout plugin shipped with ARC: it's purpose is to write accounting log for every finished job in BLAH format.

*Example:*

```
statecallout=FINISHED timeout=10,onfailure=pass /usr/libexec/arc/arc-blahp-logger -
↳I %I -U %u -L %C/job.%I.local -P %C/job.%I.proxy
```

**Warning:** CHANGE41: renamed from authplugin

### wakeupperiod

*Synopsis:* wakeupperiod = time

*Description:* Specifies how often A-REX checks for new jobs arrived, job state change requests, etc. That is responsiveness of A-REX. time is time period in seconds. Default is 3 minutes. Usually no need to change this

parameter because important state changes are also triggering out-of-schedule checks. NOTE: This parameter does not affect responsiveness of backend scripts - especially `scan-<LRMS>-job`. That means that upper estimation of time for detecting job finished executing is sum of responsiveness of backend script + `wakeupperiod`.

*Default:* 180

*Example:*

```
wakeupperiod=180
```

### infoproviders\_timelimit

*Synopsis:* `infoproviders_timelimit = seconds`

*Description:* (previously `infoproviders_timeout`) Sets the execution time limit of the infoprovider scripts started by the A-REX. Infoprovider scripts running longer than the specified `timelimit` are gracefully handled by the A-REX (the behaviour depends on the state of the system) Increase this value if you have many jobs in the controldir and infoproviders need more time to process.

*Default:* 10800

*Example:*

```
infoproviders_timelimit=10800
```

**Warning:** CHANGED42: rename, `infoproviders_timelimit`, move to `[arex]` block

### pidfile

*Synopsis:* `pidfile = path`

*Description:* Specify location of file containing PID of daemon process.

*Default:* `/var/run/arc/arex.pid`

*Example:*

```
pidfile=/var/run/run/arex.pid
```

**Warning:** TODO: modified pid file location and name

### mail

*Synopsis:* `mail = email_address`

*Description:* Specifies the email address from where the notification mails are sent

*Default:* `$VAR{user}@$VAR{[common]hostname}`

*Example:*

```
mail=grid.support@somewhere.org
```

**Warning:** CHANGE43: modified. implement a default! in the format of `root@localhost`.

## gnu\_time

*Synopsis:* gnu\_time = path

*Description:* The gnu time command

**Warning:** CHANGE: moved to [lrms]

## nodename

*Synopsis:* nodename = path

*Description:* The command to obtain hostname of computing node.

**Warning:** CHANGE: moved to [lrms]

## helper

*Synopsis:* helper = user executable arguments

*Description:* By enabling this parameter A-REX will run an external `helper` program under the user `user-account`. The program will be kept running, every time the executable finishes it will be started again. As a limitation, currently only ‘.’ is supported as username, which corresponds to the user running A-REX.

*Default:* undefined

*Example:*

```
helper=. /usr/local/bin/myutility
```

## helperlog

*Synopsis:* helperlog = path

*Description:* Configuration option to specify the location of log for helpers.

**Warning:** TODO: check if not in controldir

*Default:* /var/log/arc/job.helper.errors

*Example:*

```
helperlog=/var/log/arc/job.helper.errors
```

**Warning:** CHANGE44: new parameter

## forcedefaultvoms

*Synopsis:* forcedefaultvoms = VOMS\_FQAN

*Description:* specify VOMS FQAN which user will be assigned if his/her credentials contain no VOMS attributes. To assign different values to different queues put this command into [queue] block.

**Warning:** TODO: was missed in queue, check it actually works with queues

*Default:* undefined

*Example:*

```
forcedefaultvoms=/vo/group/subgroup
```

localcred timeout plugin\_path - Every time an external executable

**Warning:** CHANGE45: DELETED. remove from code, config.

globus\_tcp\_port\_range - Firewall configuration.

**Warning:** CHANGE46: MOVED from this block to [arex/data-staging]

globus\_udp\_port\_range - Firewall configuration.

**Warning:** CHANGE47: MOVED from this block to [arex/data-staging]

x509\_user\_cert - Location of credentials for service.

**Warning:** CHANGE48: DELETED. no need to separately set these for A-REX. The common block is enough.

**Warning:** TODO: for consistency with other blocks maybe we should add the cert-related options back?

x509\_user\_key - Location of credentials for service.

**Warning:** CHANGE49: DELETED. no need to separately set these for A-REX. The common block is enough.

x509\_cert\_dir - Location of trusted CA certificates

**Warning:** CHANGE50: DELETED. no need to separately set these for A-REX. The common block is enough.

### [arex/cache] block

This subblock enables and configures the cache functionality of A-REX. A-REX can cache input files downloaded as part of stagein proces of grid jobs so that subsequent jobs requiring the same file don't have to download it again. The cached file will be symlinked (or copied) into the session directory of the job. To disable to cache functionality simply comment out the [arex/cache] config block. It is a good idea to have the cache on its own separate file system that is shared with the nodes. For more information about the cache functionality of A-REX consult the sysadmin guide.

**Warning:** CHANGE51: new block

## cachedir

*Synopsis:* `*cachedir = cache_path [link_path]`

*Description:* Specifies a directory to store cached data. Multiple cache directories may be specified. Cached data will be distributed evenly over the caches. Optional `link_path` specifies the path at which the `cache_path` is accessible on computing nodes, if it is different from the path on the A-REX host. If `link_path` is set to `.` files are not soft-linked, but copied to session directory. If a cache directory needs to be drained, then `link_path` should specify `drain`, in which case no new files will be added to the cache. Restart of `arex` also needed.

This option is **multivalued**.

*Default:* undefined

*Example:*

```
cachedir=/scratch/cache
cachedir=/shared/cache /frontend/jobcache
cachedir=/fs1/cache drain
```

## [arex/cache/cleaner] block

This subblock enables the cleaning functionality of the cache. If this block is not enabled then the cache will not be cleaned by A-REX. Either `cachesize` or `cachelifetime` should also be set to enable cleaning.

**Warning:** CHANGE: new block

## logfile

*Synopsis:* `logfile = path`

*Description:* (previously `cachelogfile`) sets the filename where output of the cache-clean tool should be logged. Defaults to `/var/log/arc/cache-clean.log`.

*Default:* `/var/log/arc/cache-cleaner.log`

*Example:*

```
logfile=/tmp/cache-clean.log
```

**Warning:** CHANGE52: renamed as logfile

## loglevel

*Synopsis:* `loglevel = level`

*Description:* (previously `cacheloglevel`) specifies the level of logging by the cache-clean tool, between 0 (FATAL) and 5 (DEBUG). Defaults to 3 (INFO).

*Allowed values:* 0, 1, 2, 3, 4, 5

*Default:* 3

*Example:*

```
loglevel=4
```

**Warning:** CHANGE53: renamed as loglevel

remotecachedir cache\_path [link\_path] - specifies caches which are under

**Warning:** CHANGE54: DELETED, this feature is not in use any longer. remove parameter and clean code

## cacheseize

*Synopsis:* cacheseize = max min

*Description:* Specifies high and low watermarks for space used by cache, as a percentage of the space on the file system on which the cache directory is located. When the max is exceeded, files will be deleted to bring the used space down to the min level. It is a good idea to have the cache on its own separate file system.

*Default:* 100 100

*Example:*

```
cacheseize=50 20
```

## calculatesize

*Synopsis:* calculatesize = filesystem/cachedir

*Description:* (previously cacheshared) specifies the way the space occupied by the cache will be calculated. If set to cachedir then cache-clean calculates the size of the cache instead of using filesystem used space.

*Allowed values:* filesystem, cachedir

*Default:* filesystem

*Example:*

```
calculatesize=cachedir
```

**Warning:** CHANGE: renamed parameter

## cachelifetime

*Synopsis:* cachelifetime = time

*Description:* Turns on time-based file cleaning. Files accessed less recently than the given time period will be deleted. Example values of this option are 1800, 90s, 24h, 30d. When no suffix is given the unit is seconds.

*Default:* undefined

*Example:*

```
cachelifetime=30d
```



## cachespacetool

*Synopsis:* `cachespacetool = path [options]`

*Description:* specifies an alternative tool to `df` that `cache-clean` should use to obtain space information on the cache file system. The output of this command must be `total_bytes` `used_bytes`. The cache directory is passed as the last argument to this command.

*Default:* undefined

*Example:*

```
cachespacetool=/etc/getspace.sh
```

## cacheleantimeout

*Synopsis:* `cacheleantimeout = time`

*Description:* the timeout in seconds for running the `cache-clean` tool. If using a large cache or slow file system this value can be increased to allow the cleaning to complete. Defaults to 3600 (1 hour).

*Default:* 3600

*Example:*

```
cacheleantimeout=10000
```

## [arex/data-staging] block

This subblock enables and configures the data staging capabilities of A-REX. A subsystem called DTR (Data Transfer Reloaded) is responsible for collecting input data for a job before submission to the LRMS, and for staging out data after the job has finished. Automagic data staging of A-REX is a very powerful feature, disabling this functionality (by commenting out the subblock) is not recommended.

**Warning:** CHANGE55: RENAMED block

## loglevel

*Synopsis:* `loglevel = number`

*Description:* (previously `debug`) Sets the log level for transfer logging in `job.id.errors` files, between 0 (FATAL) and 5 (DEBUG). Default is to use value set by `loglevel` option in `[arex]` section.

*Allowed values:* 0, 1, 2, 3, 4, 5

*Default:* `$VAR{[arex]loglevel}`

*Example:*

```
loglevel=4
```

**Warning:** CHANGE56: renamed as `loglevel`.

## logfile

*Synopsis:* logfile = path

*Description:* (previously central\_logfile) A central file in which all data staging messages from every job will be collected and logged in addition to their job.id.errors files. If this option is not present or the path is empty the log file is not created. This file is not automatically controlled by logrotate unless you name it as /var/log/arc/datastaging.log.

*Default:* undefined

*Example:*

```
logfile=/var/log/arc/datastaging.log
```

**Warning:** CHANGE: renamed

## statefile

*Synopsis:* statefile = path

*Description:* (previously dtrlog) A file in which data staging state information (for monitoring and recovery purposes) is periodically dumped.

*Default:* \$VAR{[arex]controldir}/dtr.state

*Example:*

```
statefile=/tmp/dtr.state
```

**Warning:** CHANGE57: renamed, modified: new default value

## usehostcert

*Synopsis:* usehostcert = yes/no

*Description:* Whether the A-REX host certificate should be used for communication with remote hosts instead of the users' proxies.

*Allowed values:* yes, no

*Default:* no

*Example:*

```
usehostcert=yes
```

## maxtransfertries

*Synopsis:* maxtransfertries = number

*Description:* the maximum number of times download and upload will be attempted per job (retries are only performed if an error is judged to be temporary)

*Default:* 10

*Example:*

```
maxtransfertries=20
```

### securetransfer

*Synopsis:* securetransfer = yes|no

*Description:* if data connection allows to choose use secure/non-secure data transfer. Currently only works for gridftp.

**Warning:** TODO: DELETE

**Warning:** CHANGE: DELETED

### passivetransfer

*Synopsis:* passivetransfer = yes/no

*Description:* If yes, gridftp transfers are passive. Setting this option to yes can solve transfer problems caused by firewalls.

*Allowed values:* yes, no

*Default:* yes

*Example:*

```
passivetransfer=yes
```

**Warning:** CHANGE58: new default: yes

### globus\_tcp\_port\_range

*Synopsis:* globus\_tcp\_port\_range = port\_range

*Description:* In a firewalled environment the software which uses GSI needs to know what ports are available. This parameter is only needed if passivetransfer=no was set. These variable are similar to the Globus enviroment variables GLOBUS\_TCP\_PORT\_RANGE and GLOBUS\_UDP\_PORT\_RANGE.

*Default:* 9000, 9300

**Warning:** TODO: test if a space such as 9000, 9300 brakes gridftp in ARC?

*Example:*

```
globus_tcp_port_range=9000,12000
```

**Warning:** CHANGE59: moved here

### globus\_udp\_port\_range

*Synopsis:* globus\_udp\_port\_range = port\_range

*Description:* In a firewalled environment the software which uses GSI needs to know what ports are available. This parameter is only needed if passivetransfer=no was set. These variable are similar to the Globus enviroment variables GLOBUS\_TCP\_PORT\_RANGE and GLOBUS\_UDP\_PORT\_RANGE.

*Default:* 9000, 9300

*Example:*

```
globus_udp_port_range=9000,12000
```

**Warning:** CHANGE 60: moved here

### httpgetpartial

*Synopsis:* httpgetpartial = yes/no

*Description:* If yes, HTTP GET transfers may transfer data in chunks/parts. If no - data is always transfered in one piece.

*Allowed values:* yes, no

*Default:* no

*Example:*

```
httpgetpartial=no
```

**Warning:** CHANGE61: new default: no

### speedcontrol

*Synopsis:* speedcontrol = min\_speed min\_time min\_average\_speed max\_inactivity

*Description:* specifies how slow data transfer must be to trigger error. Transfer is cancelled if speed is below min\_speed bytes per second for at least min\_time seconds, or if average rate is below min\_average\_speed bytes per second, or no data was transferred for longer than max\_inactivity seconds. Value of zero turns feature off.

*Default:* 0 300 0 300

*Example:*

```
speedcontrol=0 300 100 300
speedcontrol=""
```

**Warning:** CHANGE62: modified: missing parameter speedcontrol= should also turn the feature off, not just zero value.

### maxdelivery

*Synopsis:* maxdelivery = number

*Description:* Maximum number of concurrent file transfers, i.e. active transfers using network bandwidth. This is the total number for the whole system including any remote staging hosts.

*Default:* 10

*Example:*

```
maxdelivery=40
```

### maxprocessor

*Synopsis:* maxprocessor = number

*Description:* Maximum number of concurrent files in each pre- and post- processing state, eg cache check or replica resolution.

**Warning:** TODO: clarify what are those post and pre states, explain it to sysadmins

*Default:* 10

*Example:*

```
maxprocessor=20
```

### maxemergency

*Synopsis:* maxemergency = number

*Description:* Maximum emergency slots which can be assigned to transfer shares when all slots up to the limits configured by the above two options are used by other shares. This ensures shares cannot be blocked by others.

*Default:* 1

*Example:*

```
maxemergency=5
```

### maxprepared

*Synopsis:* maxprepared = number

*Description:* Maximum number of files in a prepared state, i.e. pinned on a remote storage such as SRM for transfer. A good value is a small multiple of maxdelivery.

*Default:* 200

*Example:*

```
maxprepared=250
```

### sharepolicy

*Synopsis:* sharepolicy = grouping

*Description:* (previously sharetype) Defines the mechanism to be used for the grouping of the job transfers. DTR assigns the transfers to shares, so that those shares can be assigned to different priorities. Possible values for grouping are dn, voms:vo, voms:role and voms:group:

**dn** each job is assigned to a share based on the DN of the user submitting the job.

**voms:vo** each job is assigned to a share based on the VO specified in the proxy.

**voms:role** each job is assigned to a share based on the role specified in the first attribute found in the proxy.

**voms:group** each job is assigned to a share based on the group specified in the first attribute found in the proxy.

In case of the voms schemes, if the proxy is not a VOMS proxy, then a default share is used. If sharepolicy is not set then the client-defined priority is applied.

*Default:* undefined

*Example:*

```
sharepolicy=voms:role
```

**Warning:** CHANGE63: renamed

## sharepriority

*Synopsis:* sharepriority = share priority

*Description:* (previously definedshare) Defines a share with a fixed priority, different from the default (50). Priority is an integer between 1 (lowest) and 100 (highest).

This option in **multivalued**.

*Default:* undefined

*Example:*

```
sharepriority=myvo:students 20
sharepriority=myvo:production 80
```

**Warning:** CHANGE64: renamed

## copyurl

*Synopsis:* copyurl = url\_head local\_path

*Description:* Configures that DTR should use copy instead of download in case of certain stagein files. URLs, starting from 'url\_head' should be accessed in a different way (most probably unix open). The 'url\_head' part of the URL will be replaced with 'local\_path' and file from obtained path will be copied to the session directory. NOTE: 'local\_path' can also be of URL type.

This option in **multivalued**.

*Default:* undefined

*Example:*

```
copyurl=gsiftp://example.org:2811/data/ /data/
copyurl=gsiftp://example2.org:2811/data/ /data/
```

## linkurl

*Synopsis:* linkurl = url\_head local\_path [node\_path]

*Description:* Identical to 'copyurl', configures DTR so that for certain URLs files won't be downloaded or copied (in case of copyurl), but soft-link will be created. The 'local\_path' specifies the way to access the file from the frontend, and is used to check permissions. The 'node\_path' specifies how the file can be accessed from computing nodes, and will be used for soft-link creation. If 'node\_path' is missing - 'local\_path' will be used.

This option is **multivalued**.

*Default:* undefined

*Example:*

```
linkurl=gsiftp://somewhere.org/data /data
linkurl=gsiftp://example.org:2811/data/ /scratch/data/
```

## use\_remote\_acix

*Synopsis:* use\_remote\_acix = URL

*Description:* (previously acix\_endpoint) If configured then the ARC Cache Index, available at the URL, will be queried for every input file specified in a job description and any replicas found in sites with accessible caches will be added to the replica list of the input file. The replicas will be tried in the order specified by preferredpattern variable.

*Default:* undefined

*Example:*

```
use_remote_acix=https://cacheindex.ndgf.org:6443/data/index
```

**Warning:** CHANGE65: rename

## preferredpattern

*Synopsis:* preferredpattern = pattern

*Description:* specifies a preferred pattern on which to sort multiple replicas of an input file. It consists of one or more patterns separated by a pipe character (|) listed in order of preference. Replicas will be ordered by the earliest match. If the dollar character (\$) is used at the end of a pattern, the pattern will be matched to the end of the hostname of the replica. If an exclamation mark (!) is used at the beginning of a pattern, any replicas matching the pattern will be excluded from the sorted replicas.

*Default:* undefined

*Example:*

```
preferredpattern=srm://myhost.ac.uk|.uk$|ndgf.org$|badhost.org$
```

The following options are used to configure multi-host data staging deployment scenario. In that setup a couple of additional data staging boxes are enabled to off-load data transfers.

deliveryservice = URL - The URL to a remote data delivery service which can perform remote data staging.

*Default:* undefined

*Example:*

```
deliveryservice=https://myhost.org:443/datadeliveryservice
```

### localdelivery

*Synopsis:* localdelivery = yes/no

*Description:* If any deliveryservice is defined, this option determines whether local data transfer is also performed.

*Allowed values:* yes, no

*Default:* no

*Example:*

```
localdelivery=yes
```

### remotesizelimit

*Synopsis:* remotesizelimit = size

*Description:* Lower limit on file size (in bytes) of files that remote hosts should transfer. Can be used to increase performance by transferring small files using local processes.

*Default:* undefined

*Example:*

```
remotesizelimit=100000
```

### [arex/ws] block

A-REX exposes a set of Web Service interfaces that can be used to create and manage jobs, obtain information about the CE and the jobs, handle delegations, access cache information, so on. Comment out this block if you don't want to provide WS-interfaces for various A-REX functionalities.

**Warning:** CHANGE66: new block. most of the parameters originates from the old [grid-manager] block

### wsurl

*Synopsis:* wsurl = url

*Description:* (previously arex\_mount\_point) Specifies the base URL under which the web service interfaces will be available. The URL argument must be a full URL consisting of protocol+host+port+path: e.g. `https://<hostname>:<port>/<path>` Make sure the chosen port is not blocked by firewall or other security rules.

*Default:* `https://$VAR{ [common]hostname } :443/arex`

*Example:*

```
wsurl=https://piff.hep.lu.se:443/arex
```

**Warning:** CHANGE67: renamed. modified: set default as `https://hostname:443/arex`



enable\_arc\_interface yes/no - turns on or off the ARC own WS interface

**Warning:** CHANGE68: DELETED. All non-EMIES interfaces (bes and co.) should be removed from the code since EMI-ES will be the only WS interface we support.

enable\_emies\_interface - enable the EMI Execution Service interface.

**Warning:** CHANGE69: DELETED. The interfaces are enabled/disabled by the block, no need for such parameter.

## logfile

*Synopsis:* logfile = path

*Description:* (previously wslogfile) Specify log file location for WS-interface operations.

*Default:* /var/log/arc/ws-interface.log

*Example:*

```
logfile=/var/log/arc/ws-interface.log
```

**Warning:** CHANGE70: renamed.

## max\_job\_control\_requests

*Synopsis:* max\_job\_control\_requests = number

*Description:* The max number of simultaneously processed job management requests over WS interface - like job submission, cancel, status check etc.

*Default:* 100

*Example:*

```
max_job_control_requests=100
```

## max\_infosys\_requests

*Synopsis:* max\_infosys\_requests = number

*Description:* The max number of simultaneously processed info requests over WS interface.

*Default:* 1

*Example:*

```
max_infosys_requests=1
```

## max\_data\_transfer\_requests

*Synopsis:* max\_data\_transfer\_requests = number

*Description:* The max number of simultaneously processed data transfer requests over WS interface - like data staging.

*Default:* 100

*Example:*

```
max_data_transfer_requests=100
```

### [arex/ws/emies] block

EMIES is The Web Service interface offered by A-REX to create and manage computational jobs, obtain resource and activity information, also handle delegation. EMIES is a complete set of Computing Element interface, it encapsulates all the functionality a CE should expose via its interface. See interface description at [http://www.nordugrid.org/documents/EMI-ES-Specification\\_v1.16.odt](http://www.nordugrid.org/documents/EMI-ES-Specification_v1.16.odt) If you don't want to offer EMIES interface, remove or comment out this block in our `arc.conf`.

**Warning:** CHANGE71: new subblock

### require\_gridmapfile

*Synopsis:* `require_gridmapfile = yes/no`

*Description:* (previously `allowunknown`) Specifies whether to check user subject against grid-mapfile and reject users not listed in gridmap-file.

*Allowed values:* `yes, no`

*Default:* `no`

*Example:*

```
require_gridmapfile=no
```

**Warning:** CHANGE: renamed and reverted default behaviour not to require gridmap listing

### allownew

*Synopsis:* `allownew = yes/no`

*Description:* The 'allownew' config parameter sets if the Computing Element accepts submission of new jobs via the WS-interface. This parameter can be used to close down the CE.

*Allowed values:* `yes, no`

*Default:* `yes`

*Example:*

```
allownew=yes
```

**Warning:** CHANGE72: new parameter in this block. Implement support in the code for WS-interface.

## allownew\_override

*Synopsis:* allownew\_override = [authgroup ...]

*Description:* (previously allowsubmit) Defines which authorization groups are allowed to submit new jobs via the WS-interfaces when the CE is closed with allownew=no. Note that it requires the allownew=no to be set.

This option is **multivalued**.

*Default:* undefined

*Example:*

```
allownew_override=biouers atlasusers
allownew_override=yourauthgroup
```

**Warning:** CHANGE73: rename. modify behaviour: possible to specify several groups on a single line!

## allowaccess

*Synopsis:* allowaccess = authgroup [authgroup ...]

*Description:* (previously groupcfg) Specifies authorization groups that are allowed to access the WS EMIES - based jobsubmission interface. If this parameter is not set, then no additional restrictions are applied for authorizing user access. The matching group is also used for evaluating VOMS attributes representing the user.

*Default:* undefined

**Warning:** TODO: check who is authorized in case of no gridmapfile and empty allowaccess

*Example:*

```
allowaccess=biouers atlasusers
```

**Warning:** CHANGE74: new parameter in the EMIES block

## maxjobdesc

*Synopsis:* maxjobdesc = size

*Description:* specifies maximal allowed size of job description in bytes. Default value is 5MB. Use 0 to set unlimited size.

*Default:* 5242880

*Example:*

```
maxjobdesc=0
```

**Warning:** CHANGE75: new parameter in this block. Implement support in the code for WS-interface.

### [arex/ws/cache] block

The content of the A-REX cache can be accessed via a WS-interface. Configuring this block will allow reading cache files through a special URL. For example, if the remote file `gsiftp://remotehost/file1` is stored in the cache and the WS interfaces (configured above) are available via `wsurl` of `https://hostname:443/arex/`, then the cached copy of the file can be access via the following special URL: `https://hostname:443/arex/cache/gsiftp://remotehost/file1`  
Comment out this block if you don't want to expose the cache content via WS-interface.

**Warning:** CHANGE76: new block

### cacheaccess

*Synopsis:* `cacheaccess = rule`

*Description:* This parameter defines the access control rules for the cache wsinterface, the rules for allowing access to files in the cache remotely through the A-REX web interface. If not set, then noone can access anything. The default is not set that means complete denial. A rule has three parts:

1. Regular expression defining a URL pattern
2. Credential attribute to match against a client's credential
3. Regular expression defining a credential value to match against a client's credential

A client is allowed to access the cached file if a URL pattern matches the cached file URL and the client's credential has the attribute and matches the value required for that pattern. Possible values for credential attribute are `dn`, `voms:vo`, `voms:role` and `voms:group`.

This option in **multivalued**.

*Default:* undefined

*Example:*

```
cacheaccess=gsiftp://host.org/private/data/. * voms:vo myvo:production
cacheaccess=gsiftp://host.org/private/data/bob/. * dn /O=Grid/O=NorduGrid/. *
```

### [arex/ws/candypond] block

The CandyPond (Cache and deliver your pilot on-demand data) A-REX Web Service (previously called Cache Service) exposes various useful data-staging related operations for the pilot job model where input data for jobs is not known until the job is running on the worker node. This service is intended to be used by A-REX managed jobs. This service requires the `[arex/data-staging]` functionality.

The CandyPond service is available via the `wsurl/candypond` URL (e.g. `https://hostname:443/arex/candypond`)

**Warning:** TODO: revise packaging now that it is NOT a separate, independent service.

**Warning:** CHANGE77: NEW block, renamed service

`enable_cache_service yes/no` - Turn on or off the cache service interface.

**Warning:** CHANGE78: DELETED. Service interface, functionality is enabled/disabled by subblocks.

### [arex/ws/argus] block

The Web Service components of A-REX may directly use the Argus service (<https://twiki.cern.ch/twiki/bin/view/EGEE/AuthorizationFramework>) for requesting authorization decisions and performing client mapping to a local user account. This block turns on and configures the A-REX WS - Argus integration. When this block is enabled, A-REX will communicate to Argus PEP or PDP service for every WS interface operation! Comment out this block if you don't intend to use any external Argus service with A-REX WS interfaces. Using Argus with gridftp interface is possible only via LCMAPS callout.

**Warning:** CHANGE79: new block for the Argus functionality. Change code so that Argus is turned on/off based on this block

### arguspep\_endpoint

*Synopsis:* `*arguspep_endpoint = url`

*Description:* Specifies URL of Argus PEPD service to use for authorization and user mapping. It is worth to mention that `requireClientCertAuthentication` (default is false) item of `pepd.ini` (configuration of Argus PEPD service) is set to be 'true', then https should be used, otherwise http is proper.

---

**Note:** Argus will be contacted for every WS interface operation requested!

---

*Default:* undefined

*Example:*

```
arguspep_endpoint=https://somehost.somedomain:8154/authz
```

### arguspep\_profile

*Synopsis:* `arguspep_profile = profile_name`

*Description:* Defines which communication profile to use while communicating with Argus PEPD service. Possible values for `profile_name` are:

direct - pass all authorization attributes (only for debugging) subject - pass only subject name of client  
emi - ARC native profile developed in EMI project. This is default option.

*Allowed values:* direct, subject, emi

*Default:* emi

*Example:*

```
arguspep_profile=emi
```

**Warning:** CHANGE80: modified parameter values: remove cream (or at least hide from the `arc.conf.reference`)

### arguspep\_usermap

*Synopsis:* `arguspep_usermap = yes/no`

*Description:* Specifies whether response from Argus service may define mapping of client to local account. Default is 'no'. Note that Argus is contacted after all the other user mapping is performed. Hence it can overwrite all other decisions.

*Allowed values:* yes, no

*Default:* no

*Example:*

```
arguspep_usermap=no
```

### argusdpd\_endpoint

*Synopsis:* argusdpd\_endpoint = url

*Description:* Specifies URL of Argus PDP service to use for authorization and user mapping. It is worth to mention that `requireClientCertAuthentication` (default is false) item of `pdp.ini` (configuration of Argus PDP service) is set to be 'true', then https should be used, otherwise http is proper.

---

**Note:** Argus will be contacted for every WS interface operation requested!

---

*Default:* undefined

*Example:*

```
argusdpd_endpoint=https://somehost.somedomain:8152/authz
```

### argusdpd\_profile

*Synopsis:* argusdpd\_profile = profile\_name

*Description:* Defines which communication profile to use while communicating with Argus PDP service. Possible values for profile\_name are:

subject - pass only subject name of client emi - ARC native profile developed in EMI project. This is default option.

*Allowed values:* subject, emi

*Default:* emi

*Example:*

```
argusdpd_profile=emi
```

**Warning:** CHANGE81: modified parameter values: remove cream (or at least hide from the `arc.conf.reference`)

### argusdpd\_accpetnotapplicable

*Synopsis:* argusdpd\_accpetnotapplicable = yes/no

*Description:* Specify if the `NotApplicable` decision returned by Argus PDP service is treated as reason to deny request. Default is 'no', which treats `NotApplicable` as reason to deny request.

*Allowed values:* yes, no

*Default:* no

*Example:*

```
argusdpd_acceptnotapplicable=no
```

### [arex/jura] block

JURA is the accounting record generating and reporting module of A-REX. A-REX periodically executes JURA to process the job log files, and based on the accounting target destinations specified in them, JURA creates usage records in the appropriate format and sends the generated records to one or more accounting service destinations. Optionally, the generated accounting records can be archived on the CE. Enable and configure this block if you want to send accounting records to accounting services. Note that a dedicated `accounting target` subblock is needed for every accounting destination. The target subblocks are either of a type `apel` or `sgas`: `[arex/jura/apel:targetname]` or `[arex/jura/sgas:targetname]`

**Warning:** CHANGE85: new dedicated block for JURA

**Warning:** CHANGE86: A-REX should NOT provide the possibility of client-side target selection (in xrsl) any longer!

### logfile

*Synopsis:* `logfile = path`

*Description:* (previously `jobreport_logfile`) The name of the jura logfile.

*Default:* `/var/log/arc/jura.log`

*Example:*

```
logfile=/var/log/arc/jura.log
```

**Warning:** CHANGE87: renamed.

### loglevel

*Synopsis:* `loglevel = number`

*Description:* Log level for the JURA accounting module.

*Allowed values:* 0, 1, 2, 3, 4, 5

*Default:* 3

*Example:*

```
loglevel=3
```

**Warning:** CHANGE88: new parameter! implement it in the code

## urbatchsize

*Synopsis:* urbatchsize = number

*Description:* JURA sends usage records not one-by-one, but in batches.

**Warning:** CHANGE: DELETED from this block because it should be set either in APEL or SGAS subblocks

## vomsless\_vo

*Synopsis:* vomsless\_vo = voname[#voissuer]

*Description:* This parameter allows the sysadmin to manually assign VOs to jobs that were submitted with VOMS-less grid proxies. voname is the VO name to be used in the generated records (the same as expected in voms-proxy) optional voissuer (relevant to SGAS only) value is the VOMS server identity (certificate DN).

*Default:* undefined

*Example:*

```
vomsless_vo=atlas
vomsless_vo=atlas#/DC=ch/DC=cern/OU=computers/CN=lcg-voms.cern.ch
```

**Warning:** CHANGE92: new parameter.

## vo\_group

*Synopsis:* vo\_group = group

*Description:* Adds an additional VO group attribute to the usage records.

*Default:* undefined

*Example:*

```
vo_group=/atlas/production
```

**Warning:** CHANGE93: new parameter

## urdelivery\_keepfailed

*Synopsis:* urdelivery\_keepfailed = days

*Description:* Specifies for how many days JURA will try to send a record to the destination accounting service before it gives up. Records not successfully sent by after the number of days expired will be deleted from the controldir/logs directory. The deleted records are nevertheless archived if archiving was turned on. This parameter can be overwritten with a target specific value.

*Default:* 30

*Example:*

```
urdelivery_keepfailed=30
```



**Warning:** CHANGE94: new parameter.

### urdelivery\_frequency

*Synopsis:* urdelivery\_frequency = seconds

*Description:* (previously jobreport\_period) Specifies the frequency of message publishing: how often the JURA process is started by A-REX.

*Default:* 3600

*Example:*

```
urdelivery_frequency=3600
```

**Warning:** CHANGE95: renamed

jobreport publisher - name of the accounting records publisher.

**Warning:** CHANGE96: DELETED. hardcode JURA as THE publisher in A-REX.

### x509\_host\_key

*Synopsis:* x509\_host\_key = path

*Description:* Optional parameter to overwrite [common] block values.

*Default:* \$VAR{ [common]x509\_host\_key }

*Example:*

```
x509_host_key=/etc/grid-security/hostkey.pem
```

**Warning:** CHANGE97: new parameter in this block

### x509\_host\_cert

*Synopsis:* x509\_host\_cert = path

*Description:* Optional parameter to overwrite [common] block values.

*Default:* \$VAR{ [common]x509\_host\_cert }

*Example:*

```
x509_host_cert=/etc/grid-security/hostcert.pem
```

**Warning:** CHANGE98: new parameter in this block

### x509\_cert\_dir

*Synopsis:* x509\_cert\_dir = path

*Description:* Optional parameter to overwrite [common] block values.

*Default:* \$VAR{[common]x509\_cert\_dir}

*Example:*

```
x509_cert_dir=/etc/grid-security/certificates
```

**Warning:** CHANGE99: new parameter in this block

jobreport credentials path [key\_file ``[cert\_file ``[ca\_dir]]““

**Warning:** CHANGE100: DELETED. make sure jura can use the standard credentials from common block or the optional x509 parameters above

jobreport options [name:value, ...]- specifies additional parameters for the jobreporter.

**Warning:** CHANGE101: DELETED. a new block structure was created to configure those jura options.

jobreport=URL ... number specifies that A-REX has to report information about jobs being

**Warning:** CHANGE102: DELETED. new blocks are introduced as a replacement.

### [arex/jura/archiving] block

This block enables the archiving functionality of JURA. If archiving is enabled then the generated usage records are stored in a the specified archivedir directory on the disk with the following file naming convention used: usagerecord<Type>.<jobid>.<random>. Here Type is one of the supported record formats such as OGFUR or CAR. By default the archiving is turned off.

**Warning:** CHANGE89: NEW block

### archivettl

*Synopsis:* archivettl = days

*Description:* The archive time to live (ttl) parameter sets the number of days to keep the archive records in the archivedir. If not specified the records are kept forever.

*Default:* undefined

*Example:*

```
archivettl=365
```

**Warning:** CHANGE90: new parameter. new feature to implement!

## archivedir

*Synopsis:* archivedir = path

*Description:* Sets the directory path for the archived usage records. If archiving is turned on, then the generated usage records are stored in this directory with the following file naming convention used:: usagerecord<Type>.<jobid>.<random>. Here Type is one of the supported record formats such as OGFUR or CAR.

*Default:* /var/spool/arc/jura/archive

**Warning:** TODO: create the directory

*Example:*

```
archivedir=/var/spool/arc/jura/archive
```

**Warning:** CHANGE172: NEW

## [arex/jura/sgas:targetname] block

An SGAS sub-block of [arex/jura] enables and configures an SGAS accounting server as a target destination to which JURA will send properly formatted usage records. The [arex/jura] config values are applicable for every target sub-block but also can be overwritten. You need to define a separate block with a unique targetname for every SGAS target server.

**Warning:** CHANGE103: new dedicated sub-blocks for SGAS JURA targets

## targeturl

*Synopsis:* \*targeturl = url

*Description:* The service endpoint URL of SGAS server.

*Default:* undefined

*Example:*

```
targeturl=https://grid.uio.no:8001/logger
```

**Warning:** CHANGE104: new parameter

## localid\_prefix

*Synopsis:* localid\_prefix = prefix\_string

*Description:* Sets a prefix value for the LocalJobID ur parameter for the SGAS usage records.

*Default:* undefined

*Example:*

```
localid_prefix=some_text_for_SGAS
```

**Warning:** CHANGE105: new parameter

## vofilter

*Synopsis:* vofilter = vo

*Description:* Configures a job record filtering mechanism based on the VO attribute of the jobs. Only the matching job records, which was one of VO that you set here, will be sent to the target accounting service.

This option is **multivalued**.

*Default:* undefined

*Example:*

```
vofilter=atlas  
vofilter=fgi.csc.fi
```

**Warning:** CHANGE106: new parameter

## urbatchsize

*Synopsis:* urbatchsize = number

*Description:* JURA sends usage records not one-by-one, but in batches. This options sets the size of a batch. Zero value means unlimited batch size. This option can also be set in the sgas/apel target blocks to overwrite common value.

*Default:* 50

*Example:*

```
urbatchsize=80
```

**Warning:** CHANGE107: new parameter.

## [arex/jura/apel:targetname] block

An APEL sub-block of [arex/jura] enables and configures an APEL accounting server as a target destination to which JURA will send properly formatted usage records. The [arex/jura] config values are applicable for every target sub-block but also can be overwritten. You need to define a separate block with a unique targetname for every APEL target server.

**Warning:** CHANGE108: new dedicated sub-blocks for APEL JURA targets

## targeturl

*Synopsis:* \*targeturl = url

*Description:* The service endpoint URL of the APEL accounting server.

*Default:* undefined

*Example:*

```
targeturl=https://mq.cro-ngi.hr:6162
```

**Warning:** CHANGE109: new parameter

### topic

*Synopsis:* topic = topic\_name

*Description:* Sets the name of the APEL topic to which JURA will publish the accounting records.

*Default:* /queue/global.accounting.cpu.central

*Example:*

```
topic=/queue/global.accounting.test.cpu.central
```

**Warning:** CHANGE110: new parameter.

### gocdb\_name

*Synopsis:* \*gocdb\_name = name

*Description:* Can be used to specify the GOCDB name of the resource. This value would be seen as Site attribute in the generated APEL record.

*Default:* undefined

*Example:*

```
gocdb_name=GRID_UIO_NO
```

**Warning:** CHANGE111: new parameter.

### benchmark\_type

*Synopsis:* benchmark\_type = type

*Description:* Type of benchmark (Si2k, Sf2k, HEPSPEC) to be reported in every UR.

*Default:* undefined

*Example:*

```
benchmark_type=HEPSPEC
```

**Warning:** CHANGE112: new parameter

### benchmark\_value

*Synopsis:* benchmark\_value = number

*Description:* The value of the of benchmark to be reported in every UR.

*Default:* undefined

*Example:*

```
benchmark_value=2.4
```

**Warning:** CHANGE113: new parameter

### benchmark\_description

*Synopsis:* benchmark\_description = string

*Description:* Additional description for the benchmark to be reported in every UR.

*Default:* undefined

*Example:*

```
benchmark_description=some description for benchmark
```

**Warning:** CHANGE114: new parameter

### use\_ssl

*Synopsis:* use\_ssl = yes/no

*Description:* Turns on/off ssl for the SSM communication with APEL server.

*Allowed values:* yes, no

*Default:* no

*Example:*

```
use_ssl=yes
```

**Warning:** CHANGE115: new parameter

### vofilter

*Synopsis:* vofilter = vo

*Description:* Configures a job record filtering mechanism based on the VO attribute of the jobs. Only the matching job records, which was one of VO that you set here, will be sent to the target accounting service.

This option in **multivalued**.

*Default:* undefined

*Example:*

```
vofilter=atlas
vofilter=fgi.csc.fi
```

**Warning:** CHANGE116: new parameter

### urbatchsize

*Synopsis:* urbatchsize = number

*Description:* JURA sends usage records not one-by-one, but in batches. This options sets the size of a batch. Zero value means unlimited batch size. This option can also be set in the sgas/apel target blocks to overwrite common value.

*Default:* 1000

*Example:*

```
urbatchsize=1000
```

**Warning:** CHANGE117: new parameter. new default.

### [arex/ganglia] block

This block enables the monitoring of ARC-specific metrics through an existing ganglia installation. Earlier versions (ARC < 6.0) relied only on the standalone tool gangliarc. Ganglia is now integrated into ARC to extract jobmetrics. Ganglia for system metrics is run as separate gangliarc instance.

**Warning:** CHANGE185: RENAMED block

### gmetric\_bin\_path

*Synopsis:* gmetric\_bin\_path = path

*Description:* (previously gmetric\_exec) The path to gmetric executable.

*Default:* /usr/bin/gmetric

*Example:*

```
gmetric_bin_path=/usr/local/bin/gmetric
```

**Warning:** CHANGE190: Moved from deleted [gangliarc] block and renamed. Path used in arc-ganglia implementation JobsMetrics.cpp

### logfile

*Synopsis:* logfile = path

*Description:* log file of the daemon.

*Default:* /var/log/arc/gangliarc.log

*Example:*

```
logfile=/tmp/gangliarc.log
```

**Warning:** TODO: is this needed for the internal ganglia?

## pidfile

*Synopsis:* pidfile = pid

*Description:* pid file of the daemon.

*Default:* /var/run/gangliarc.pid

*Example:*

```
pidfile=/tmp/gangliarc.pid
```

**Warning:** TODO: it is not a separate service, therefore pid is not there.

## python\_bin\_path

*Synopsis:* python\_bin\_path = path

*Description:* The path to python executable.

*Default:* /usr/bin/python

*Example:*

```
python_bin_path=/usr/local/bin/python
```

**Warning:** TODO (Maiken) clarify if this is still needed.

ganglialocation path - path to ganglia gmetric executable

**Warning:** CHANGE188: DELETED. documented first in 5.3.0. Use gmetric\_bin instead.

## metrics

*Synopsis:* metrics = name\_of\_the\_metrics

*Description:* the metrics to be monitored. metrics takes a comma-separated list of one or more of the following metrics: - staging – number of tasks in different data staging states - cache – free cache space - session – free session directory space - heartbeat – last modification time of A-REX heartbeat - processingjobs – the number of jobs currently being processed by ARC (jobs

between PREPARING and FINISHING states)

- failedjobs – the number of failed jobs per last 100 finished
- jobstates – number of jobs in different A-REX internal stages
- all – all of the above metrics



*Default:* all

*Allowed values:* staging, cache, session, heartbeat, processingjobs, failedjobs, jobstates, all

*Example:*

```
metrics=all
```

**Warning:** TODO which metrics is used by the internal ganglia?

## frequency

*Synopsis:* frequency = seconds

*Description:* The period between each information gathering cycle, in seconds.

*Default:* 60

*Example:*

```
frequency=300
```

**Warning:** CHANGE: default increased from 20s to one minute

## [gridftpd] block

This block enables and configures the gridftp server. The usage of the gridftp is twofold in connection with ARC: 1) The server together with its custom jobplugin can be used as a job submission and management interface for an ARC CE. 2) The server with the fileplugin can be used as a very simplistic storage element. This block configures the common server capabilities. To make the gridftp service functional, you need to enable at least one of the plugin subblocks as well.

## require\_gridmapfile

*Synopsis:* require\_gridmapfile = yes/no

*Description:* (previously allowunknown) Specifies whether to check user subject against grid-mapfile and reject users not listed in gridmap-file.

*Allowed values:* yes, no

*Default:* no

*Example:*

```
require_gridmapfile=no
```

**Warning:** CHANGE: renamed and reverted default behaviour not to require gridmap listing

## user

*Synopsis:* `user = user[:group]`

*Description:* Switch to a non root user/group after startup **WARNING:** Make sure that the certificate files are owned by the user/group specified by this option.

*Default:* `root:root`

*Example:*

```
user=grid
```

## loglevel

*Synopsis:* `loglevel = level`

*Description:* (previously debug) Set log level of the gridftpd daemon, between 0 (FATAL) and 5 (DEBUG). Default is 3 (INFO).

*Allowed values:* 0, 1, 2, 3, 4, 5

*Default:* 3

*Example:*

```
loglevel=2
```

**Warning:** CHANGE118: renamed

`daemon yes/no` - Whether the is run in daemon mode. Default is yes.

**Warning:** CHANGE119: DELETED. remove this config parameter from `arc.conf`. the functionality will remain via cli option

## logfile

*Synopsis:* `logfile = path`

*Description:* Set logfile location of the gridftp server.

*Default:* `/var/log/arc/gridftpd.log`

*Example:*

```
logfile=/var/log/arc/gridftpd.log
```

`logsize size [number]` - 'Size' specifies in bytes how big log file is

**Warning:** CHANGE120: DELETED. remove for simplification purposes.

## pidfile

*Synopsis:* `pidfile = path`

*Description:* Specify location of file containing PID of daemon process.

*Default:* /var/run/gridftpd.pid

*Example:*

```
pidfile=/var/run/gridftpd.pid
```

## port

*Synopsis:* port = bindport

*Description:* Port to listen on. For gridftp-based job submission strongly advised to use the default 2811 port because 3rd party clients assume ARC CE using that port.

*Default:* 2811

*Example:*

```
port=2811
```

## allowencryption

*Synopsis:* allowencryption = yes/no

*Description:* (previously encryption) should data encryption be allowed on client request. Encryption is very heavy, therefore the default is no.

*Allowed values:* yes, no

*Default:* no

*Example:*

```
allowencryption=no
```

**Warning:** CHANGE : RENAMED as allowencryption

include - Include contents of another config file.

**Warning:** CHANGE121: DELETED. no need for this feature any longer

## allowactivedata

*Synopsis:* allowactivedata = yes/no

*Description:* if no, only passive data transfer is allowed. By default both passive and active data transfers are allowed.

*Default:* yes

*Example:*

```
allowactivedata=yes
```

### maxconnections

*Synopsis:* maxconnections = number

*Description:* The maximum number of connections accepted by a gridftpd server.

*Default:* 100

*Example:*

```
maxconnections=200
```

### defaultbuffer

*Synopsis:* defaultbuffer = size

*Description:* Defines size of every buffer for data reading/writing. The actual value may decrease if the cumulative size of all buffers exceeds value specified by maxbuffer.

*Default:* 65536

*Example:*

```
defaultbuffer=65536
```

### maxbuffer

*Synopsis:* maxbuffer = size

*Description:* Defines maximal amount of memory in bytes to be allocated for all data reading/writing buffers. Default is 640kB. The number of buffers is (max {3, min {41, 2P + 1}}), where P is the parallelism level requested by the client. Hence, even without parallel streams enabled number of buffers will be 3.

*Default:* 655360

*Example:*

```
maxbuffer=655360
```

### globus\_tcp\_port\_range

*Synopsis:* globus\_tcp\_port\_range = port\_range

*Description:* In a firewalled environment the software which uses GSI needs to know what ports are available. If not set a random port is selected. These variable are similar to the Globus environment variables: GLOBUS\_TCP\_PORT\_RANGE and GLOBUS\_UDP\_PORT\_RANGE.

*Default:* 9000, 9300

*Example:*

```
globus_tcp_port_range=9000,12000
```

### globus\_udp\_port\_range

*Synopsis:* globus\_udp\_port\_range = port\_range

*Description:* In a firewalled environment the software which uses GSI needs to know what ports are available. If not set a random port is selected. These variable are similar to the Globus environment variables: GLOBUS\_TCP\_PORT\_RANGE and GLOBUS\_UDP\_PORT\_RANGE.

*Default:* 9000, 9300

*Example:*

```
globus_udp_port_range=9000,12000
```

## firewall

*Synopsis:* firewall = hostname

*Description:* The hostname or IP address to use in response to PASV command instead of the IP address of a network interface of computer.

*Default:* undefined

*Example:*

```
firewall=my.host.org
```

## x509\_host\_key

*Synopsis:* x509\_host\_key = path

*Description:* Optional parameter to overwrite [common] block values.

*Default:* \$VAR{ [common]x509\_host\_key }

*Example:*

```
x509_host_key=/etc/grid-security/hostkey.pem
```

**Warning:** CHANGE122: renamed

## x509\_host\_cert

*Synopsis:* x509\_host\_cert = path

*Description:* Optional parameter to overwrite [common] block values.

*Default:* \$VAR{ [common]x509\_host\_cert }

*Example:*

```
x509_host_cert=/etc/grid-security/hostcert.pem
```

**Warning:** CHANGE123: renamed

## x509\_cert\_dir

*Synopsis:* x509\_cert\_dir = path

*Description:* Optional parameter to overwrite [common] block values.

*Default:* \$VAR{ [common]x509\_cert\_dir }

*Example:*

```
x509_cert_dir=/etc/grid-security/certificates
```

pluginpath - directory where the plugin libraries are installed, default is

**Warning:** CHANGE124: DELETED. hide this parameter from `arc.conf`. gridftp service should work out-of-the-box on a standard linux installation.

### [gridftpd/jobs] block

The jobplugin of the gridftp server implements a custom job management and submission interface of ARC CE. This subblock enables and configures that interface. Consult the Technical Reference within the ARC sysadmin guide for the interface specification. Comment out this subblock if you don't want a gridftp-based jobinterface.

path vrttdir - The path to the virtual gridftpd directory which is used during the

**Warning:** CHANGE125: DELETED. remove this flexibility. the `/jobs` string must be hardcoded in the server.

plugin name - specifies name of shared library to be loaded relative to `pluginpath`.

**Warning:** CHANGE126: DELETED. hide it from sysadmin. make the plugin loading/configuration automatic. if a gridftpd/jobs block is enabled, load the right plugin automatically.

### allownew

*Synopsis:* allownew = yes/no

*Description:* This parameter sets if the ARC CE accepts submission of new jobs via the gridftp interface. This parameter can be used to close down the ARC CE.

*Allowed values:* yes, no

*Default:* yes

*Example:*

```
allownew=yes
```

### allownew\_override

*Synopsis:* allownew\_override = [authgroup ...]

*Description:* (previously allowsubmit) Defines which authorization groups are allowed to submit new jobs via the gridftp interface when the CE is closed with allownew=no. Note that it requires the allownew=no to be set.

This option in **multivalued**.

*Default:* undefined

*Example:*

```
allownew_override=biouusers atlasusers  
allownew_override=yourauthgroup
```

**Warning:** CHANGE127: rename. modify behaviour: possible to specify several groups on a single line!

### allowaccess

*Synopsis:* allowaccess = authgroup [authgroup ...]

*Description:* (previously groupcfg) Specifies authorization groups that are allowed to access the gridftp-based jobsubmission interface. If this parameter is not set, then no additional restrictions are applied for authorizing user access. The matching group is also used for evaluating VOMS attributes representing the user.

**Warning:** TODO: check what happens if no gridmapfile used with empty value

*Default:* undefined

*Example:*

```
allowaccess=biouusers atlasusers
```

**Warning:** CHANGE128: renamed

remotegmdirs controldir sessiondir - Specifies control and session directories to which jobs can be submitted but which are under the control of another A-REX.

**Warning:** CHANGE129: DELETED. remove feature and config parameter.

### maxjobdesc

*Synopsis:* maxjobdesc = size

*Description:* specifies maximal allowed size of job description in bytes. Default value is 5MB. Use 0 to set unlimited size.

*Default:* 5242880

*Example:*

```
maxjobdesc=0
```

configfile service\_configuration\_path - If [gridftpd] and [arex] configuration parts are located in separate files this configuration

**Warning:** CHANGE130: DELETED.

### [gridftpd/filedir] block

The fileplugin module of the gridftp server can be used to set up a simplistic grid storage element (SE). This subblock enables and configures such an SE by exporting a directory using the gridftpd's fileplugin. Comment out this block if you don't need a SE.

plugin name - specifies name of shared library to be loaded relative to

**Warning:** CHANGE131: DELETED. hide it from sysadmin. make the plugin loading/configuration automatic. if a gridftp/filedir block is enabled, load the right plugin automatically.

## path

*Synopsis:* \*path = virtmdir

*Description:* The name of the virtual directory served by the gridftp server. The exported storage area is accessible as `gsiftp://my_server/virtmdir`. Even `/` is a valid choice.

*Default:* undefined

*Example:*

```
path=/topdir
```

## mount

*Synopsis:* \*mount = path

*Description:* The physical directory corresponding to the virtual one: `gsiftp://my_server/virtmdir` will give access to this location.

*Default:* undefined

*Example:*

```
mount=/scratch/grid
```

## allowaccess

*Synopsis:* allowaccess = authgroup [authgroup ...]

*Description:* (previously `groupcfg`) Specifies authorization groups that are allowed to access the gridftp-based jobsubmission interface. If this parameter is not set, then no additional restrictions are applied for authorizing user access.

*Default:* undefined

*Example:*

```
allowaccess=biouers atlasusers
```

**Warning:** CHANGE132: renamed

## dir

*Synopsis:* dir = path options

*Description:* Specifies access rules for accessing files in `path` (relative to virtual and real path) and all the files and directories below. Available permissions check options are:

**nouser** do not use local file system rights, only use those specifies in this line

**owner** check only file owner access rights

**group** check only group access rights



**other** check only `others` access rights

If none of the above specified usual unix access rights are applied. Available permissions enforcement options are:

**read** allow reading files

**delete** allow deleting files

**append** allow appending files (does not allow creation)

**overwrite** allow overwriting already existing files (does not allow creation, file attributes are not changed)

**dirlist** allow obtaining list of the files

**cd** allow to make this directory current

**create owner:group permissions\_or:permissions\_and** allow creating new files. File will be owned by `owner` and owning group will be `group`. If `*` is used, the user/group to which connected user is mapped will be used. The permissions will be set to `permissions_or` & `permissions_and`. (second number is reserved for the future usage).

**mkdir owner:group permissions\_or:permissions\_and** allow creating new directories.

Example shows setting permissions on mounted `/` directory and adjusting permissions on `/section1` and `/section2` subdirectories.

This option in **multivalued**.

*Default:* undefined

**Warning:** TODO: check behaviour when this option is missing and user is mapped to root. (what is the default user mapping for file-plugin?)

*Example:*

```
dir=/ nouser read cd dirlist delete create *:* 664:664 mkdir *:* 775:775
dir=/section1 nouser read mkdir *:* 700:700 cd dirlist
dir=/section2 nouser read mkdir *:* 700:700 cd dirlist
```

## [infosys] block

This block enables and configures the core part of the information system. Enables the information collection to be used by other ARC components, including interfaces. Parameters in this block applies to all the infosys subsystems.

### logfile

*Synopsis:* logfile = path

*Description:* (previously providerlog) Specifies log file location for the information provider scripts.

*Default:* `/var/log/arc/infoprovider.log`

*Example:*

```
logfile=/var/log/arc/infoprovider.log
```

**Warning:** CHANGE133: renamed

## loglevel

*Synopsis:* `loglevel = number`

*Description:* (previously `provider_loglevel`) The loglevel for the infoprovider scripts (0-5). The infosys default is 1 (critical errors are logged) Each value corresponds to the following verbosity levels: FATAL => 0, ERROR => 1, WARNING => 2, INFO => 3, VERBOSE => 4, DEBUG => 5

*Allowed values:* 0, 1, 2, 3, 4, 5

*Default:* 1

*Example:*

```
loglevel=3
```

**Warning:** CHANGE134: renamed

`overwrite_config yesln` - determines if the infosys startup scripts should generate new low-level slapd configuration files.

**Warning:** CHANGE135: DELETED. This functionality got lost during the years. startup scripts don't support this feature any longer

`oldconfsuffix .suffix` - sets the suffix of the backup files of the low-level slapd config files.

**Warning:** CHANGE136: DELETED. This functionality got lost during the years. startup scripts don't support this feature any longer

`debug` - sets the debug level/verbosity of the startup script {0 or 1}.

**Warning:** CHANGE137: DELETED.

`infosys_compat` - Setting this variable will cause ARC to use the old infoproviders. NOTE: this only applies to ARC < 13.11. Old infoproviders

**Warning:** CHANGE138: DELETED. not relevant for a very long time by now.

`cachetime` affects old infoproviders, and forces the validity time of the record.

**Warning:** CHANGE139: DELETED. not parsed any longer.

`giis_fifo` - path to fifo used by EGIIS. default is `/var/run/arc/giis-fifo`

**Warning:** CHANGE140: DELETED. hardcode it in startupscript!

## user

*Synopsis:* `user = unix_user`

*Description:* the unix user running the infosys processes such as

**Warning:** CHANGE: DELETED, moved to the [infosys/ldap] sub-block

### validity\_ttl

*Synopsis:* validity\_ttl = seconds

*Description:* The published infosys records advertise their validity e.g. how long the info should be considered up-to-date by the clients. Use this parameter to set the published validity value.

**Note:** different schemas may render this information differently.

*Default:* 10800

*Example:*

```
validity_ttl=10800
```

### [infosys/ldap] block

This infosys subblock enables and configures the ldap hosting service for the infosys functionality. Using an LDAP server with some schema is one way to publish information about your Computing Element. Comment out this block if you don't want to run an LDAP-based information system.

**Warning:** CHANGE141: new block

### hostname

*Synopsis:* hostname = FQDN

*Description:* the hostname of the machine running the slapd service will be the bind for slapd. If not present, will be taken from the [common]

*Default:* \$VAR{ [common]hostname }

*Example:*

```
hostname=my.testbox
```

### slapd\_hostnamebind

*Synopsis:* slapd\_hostnamebind = string

*Description:* May be used to set the hostname part of the network interface to which the slapd process will bind. Most of the cases no need to set since the hostname parameter is already sufficient. The example below will bind the slapd process to all the network interfaces available on the server.

*Default:* undefined

*Example:*

```
slapd_hostnamebind=*
```

## port

*Synopsis:* port = port\_number

*Description:* The port on which the slapd service runs. The default infosys port is assumed to be 2135 by many clients, therefore think twice before you change it because 3rd party clients assume 2135 to be the ldap infosys port.

*Default:* 2135

*Example:*

```
port=2135
```

## user

*Synopsis:* user = unix\_user

*Description:* overwrites the unix user running the slapd. By default the startup scripts search for well-known ldap-users like ldap or openldap than fall-back to root if not found.

*Default:* undefined

*Example:*

```
user=slapd
```

**Warning:** CHANGE: moved here from [infosys]

## slapd

*Synopsis:* slapd = path

*Description:* explicitly define the path to slapd command. By default the startup scripts search for slapd binary in the system PATH.

*Default:* undefined

*Example:*

```
slapd=/usr/sbin/slapd
```

## slapadd

*Synopsis:* slapadd = path

*Description:* Configure where the slapadd command is located.

**Warning:** CHANGE: DELETED (not used in the code)

## slapd\_loglevel

*Synopsis:* slapd\_loglevel = number

*Description:* Sets the native slapd loglevel (see man slapd). Slapd logs via syslog. The default is set to no-logging (0) and it is RECOMMENDED not to be changed in a production environment. Non-zero slap\_loglevel value causes serious performance decrease.

*Default:* 0

*Example:*

```
slapd_loglevel=0
```

## threads

*Synopsis:* threads = number

*Description:* The native slapd threads parameter, default is 32.

*Default:* 32

*Example:*

```
threads=128
```

## timelimit

*Synopsis:* timelimit = seconds

*Description:* The native slapd timelimit parameter. Maximum number of seconds the slapd server will spend answering a search request. Default is 3600. You probably want a much lower value.

*Default:* 3600

*Example:*

```
timelimit=1800
```

## idletimeout

*Synopsis:* idletimeout = seconds

*Description:* The native slapd idletimeout parameter. Maximum number of seconds the slapd server will wait before forcibly closing idle client connections. It's value must be larger than the value of timelimit option. If not set, it defaults to timelimit + 1.

*Default:* \$EVAL{\$VAR{timelimit} + 1}

*Example:*

```
idletimeout=1801
```

giis\_location - If giis\_location is not set, ARC\_LOCATION will be used instead.

**Warning:** CHANGE142: DELETED.

slapd\_cron\_checkpoint - LDAP checkpoint enable/disable This option was introduced to solve bug #2032, to reduce the number of log files produced by BDII.

**Warning:** CHANGE143: DELETED. not relevant any longer.

db\_archive - path to slapd\_db\_archive binary Only used by the above LDAP checkpoint option

**Warning:** CHANGE144: DELETED. not relevant any longer.

db\_checkpoint - path to slapd\_db\_checkpoint binary Only used by the above LDAP checkpoint option

**Warning:** CHANGE145: DELETED. not relevant any longer.

### infosys\_ldap\_run\_dir

*Synopsis:* infosys\_ldap\_run\_dir = path

*Description:* The location where NorduGrid/GLUE2 LDAP ldif file will be generated, and where the fifo to sync between infoproviders and BDII will be generated.

*Default:* /var/run/arc/infosys/

*Example:*

```
infosys_ldap_run_dir=/var/run/arc/infosys/
```

### ldap\_schema\_dir

*Synopsis:* ldap\_schema\_dir = path

*Description:* Allows to explicitly specify an additional path to the schema files. Note that this doesn't override standard location, but adds the specified path to the standard locations /etc/ldap and /etc/openldap. Normally it is sufficient to use only standard schema file locations, therefore not to set this parameter.

*Default:* undefined

*Example:*

```
ldap_schema_dir=/nfs/ldap/schema/
```

infosys\_nordugrid enable - These three variables decide which schema should be used for

**Warning:** CHANGE146: DELETED. subblocks are used to enable/disable schema-specific publishing.

The following options configure the third-party bdii ldap parameters. In 99% of cases no need to change anything and use the defaults. These variables are usually automatically set by ARC, and are here mostly for debug purposes and to tweak exotic BDII installations.

### bdii\_debug\_level

*Synopsis:* bdii\_debug\_level = level

*Description:* Set this parameter to DEBUG to check bdii errors in bdii-update.log At the same time don't enable slapd logs this way reducing performance issues.

*Default:* ERROR

*Example:*

```
bdii_debug_level=ERROR
```

## bdii\_provider\_timeout

*Synopsis:* bdii\_provider\_timeout = seconds

*Description:* (previously provider\_timeout in bdii block) This variable allows a system administrator to modify the behaviour of bdii-update. This is the time BDII waits for the bdii provider scripts generated by A-REX infosys to produce their output.

*Default:* 10800

*Example:*

```
bdii_provider_timeout=10800
```

**Warning:** CHANGE: renamed

BDII5 uses these variables. These might change depending on BDII version. ARC sets them by inspecting distributed bdii configuration files. DO NOT

**Warning:** CHANGE UNLESS YOU KNOW WHAT YOU'RE DOING

## bdii\_location

*Synopsis:* bdii\_location = path

*Description:* The installation directory for the BDII.

*Default:* /usr

*Example:*

```
bdii_location=/usr
```

## bdii\_run\_dir

*Synopsis:* bdii\_run\_dir = path

*Description:* Contains BDII pid files and slapd pid files

*Default:* /var/run/arc/bdii

*Example:*

```
bdii_run_dir=/var/run/arc/bdii
```

## bdii\_log\_dir

*Synopsis:* bdii\_log\_dir = path

*Description:* Contains infosys logs

*Default:* /var/log/arc/bdii

*Example:*

```
bdii_log_dir=/var/log/arc/bdii
```

### **bdi\_i\_tmp\_dir**

*Synopsis:* bdi\_i\_tmp\_dir = path

*Description:* Contains provider scripts

*Default:* /var/tmp/arc/bdi\_i

*Example:*

```
bdi_i_tmp_dir=/var/tmp/arc/bdi_i
```

### **bdi\_i\_var\_dir**

*Synopsis:* bdi\_i\_var\_dir = path

*Description:* Contains slapd databases

*Default:* /var/lib/arc/bdi\_i

*Example:*

```
bdi_i_var_dir=/var/lib/arc/bdi_i
```

### **bdi\_i\_update\_pid\_file**

*Synopsis:* bdi\_i\_update\_pid\_file = path

*Description:* Allows to change bdi\_i-update pidfiles filename and location

*Default:* \$VAR{bdi\_i\_run\_dir}/bdi\_i-update.pid

*Example:*

```
bdi_i_update_pid_file=/var/run/arc/bdi_i/bdi_i-update.pid
```

slapd\_pid\_file path - Allows to change slapd pidfiles filename and location

**Warning:** CHANGE148: DELETED

### **bdi\_i\_database**

*Synopsis:* bdi\_i\_database = backend\_type

*Description:* Configure what ldap database backend should be used.

*Default:* hdb

*Example:*

```
bdi_i_database=hdb
```

### **bdi\_i\_conf**

*Synopsis:* bdi\_i\_conf = path

*Description:* Location of the bdi\_i config file generated by ARC.

*Default:* \$VAR{[infosys/ldap]infosys\_ldap\_run\_dir}/bdi\_i.conf



*Example:*

```
bdi_conf=/var/run/arc/infosys/bdi.conf
```

### **bdi\_update\_cmd**

*Synopsis:* bdi\_update\_cmd = path

*Description:* path to bdi-update script

*Default:* \$VAR{bdi\_location}/sbin/bdi-update

*Example:*

```
bdi_update_cmd=/usr/sbin/bdi-update
```

### **bdi\_db\_config**

*Synopsis:* bdi\_db\_config = path

*Description:* path to slapd database configuration file

*Default:* /etc/bdi/DB\_CONFIG

*Example:*

```
bdi_db_config=/etc/bdi/DB_CONFIG
```

### **bdi\_archive\_size**

*Synopsis:* bdi\_archive\_size = number

*Description:* Sets BDII\_ARCHIVE\_SIZE in bdi configuration file

*Default:* 0

*Example:*

```
bdi_archive_size=0
```

### **bdi\_breathe\_time**

*Synopsis:* bdi\_breathe\_time = number

*Description:* Sets BDII\_BREATHE\_TIME in bdi configuration file

*Default:* 10

*Example:*

```
bdi_breathe_time=10
```

### **bdi\_delete\_delay**

*Synopsis:* bdi\_delete\_delay = number

*Description:* Sets BDII\_DELETE\_DELAY in bdi configuration file

*Default:* 0

*Example:*

```
bdi_delete_delay=0
```

### **bdi\_read\_timeout**

*Synopsis:* bdi\_read\_timeout = number

*Description:* Sets BDII\_READ\_TIMEOUT in bdi configuration file

*Default:* \$EVAL{\$VAR{bdi\_provider\_timeout} + \$VAR{[arex]infoproviders\_timelimit} + \$VAR{[arex]wakeupperiod}}

*Example:*

```
bdi_read_timeout=300
```

### **cachettl**

*Synopsis:* cachettl = number

*Description:* cachettl affects registration to egiis The value is reported back by the index server as Mds-Service-Ldap-cachettl Default is zero, i.e. no limits are set.

**Warning:** CHANGE: DELETED

Infosys Schema sub-blocks: The following infosys sub-blocks enable information publishing according to various information schema. In order to publish information in a certain schema, the corresponding sub-block must be defined in addition to the schema-neutral [infosys/cluster] and [queue:name] blocks! Comment out a specific schema block if you don't want to publish a specific information schema representation. Currently available information model (schema) sub-blocks:

- [infosys/nordugrid] - The native ARC info representation of a cluster and its queues
- [infosys/glue2] - The GLUE2 information model, both LDAP and XML (the latter is for WS-interface)
- [infosys/glue2/ldap] - The LDAP rendering of the GLUE2 model
- [infosys/glue1] - The legacy GLUE1 model (only LDAP)
- [infosys/glue1/site-bdi] - The site BDII element of the GLUE1 legacy model

### **[infosys/nordugrid] block**

Enables the publication of the NorduGrid information model in the LDAP-based infosys. See the NORDUGRID-TECH-4 for schema definition. The configuration block does not contain any parameter. The information tree is populated based on the contents of the schema-neutral [infosys/cluster] and [queue:name] blocks.

**Warning:** CHANGE155: new schema block. should be used to turn on/off nordugrid-ldap publication.

### **[infosys/glue2] block**

Enables the publication of the GLUE2 information model both in the LDAP and XML rendering. The information tree is populated based on the contents of the schema-neutral [infosys/cluster] and [queue:name] blocks and the GLUE2 specific schema sub-blocks.

**Warning:** CHANGE156: new schema block. should be used to turn on/off GLUE2 publication (both ldap & xml)

AdminDomain entity parameters: `admindomain_name` = string - The Name attribute for the `admindomain`. This will show in top-BDII to group the resources belonging to this cluster. To group a bunch of clusters under the same AdminDomain, just use the same name. If not specified, will default to UNDEFINEDVALUE.

*Default:* UNDEFINEDVALUE

*Example:*

```
admindomain_name=ARC-TESTDOMAIN
```

### **admindomain\_description**

*Synopsis:* `admindomain_description` = text

*Description:* The free-form description of this domain.

*Default:* undefined

*Example:*

```
admindomain_description=ARC test Domain
```

### **admindomain\_www**

*Synopsis:* `admindomain_www` = url

*Description:* The URL pointing at a site holding information about the AdminDomain.

*Default:* undefined

*Example:*

```
admindomain_www=http://www.nordugrid.org/
```

### **admindomain\_distributed**

*Synopsis:* `admindomain_distributed` = yes/no

*Description:* Set this to yes if the domain is distributed that means, if the resources belonging to the domain are considered geographically distributed.

*Allowed values:* yes, no

*Default:* no

*Example:*

```
admindomain_distributed=yes
```

### **admindomain\_owner**

*Synopsis:* `admindomain_owner` = email

*Description:* The contact email of a responsible person for the domain

*Default:* undefined

*Example:*

```
admindomain_owner=admin@nordugrid.org
```

### **admindomain\_otherinfo**

*Synopsis:* `admindomain_otherinfo = text`

*Description:* Free-form text that fills the OtherInfo GLUE2 field. no need to set, used only for future development.

*Default:* undefined

*Example:*

```
admindomain_otherinfo=Test Other info
```

ComputingService entity parameters: `computingservice_qualitylevel = qllevel` - (previously `infosys_glue2_service_qualitylevel`) Allows a sysadmin to define different GLUE2 QualityLevel values for A-REX. Refer to GLUE2 documentation for the qualitylevel definitions.

*Allowed values:* production, pre-production, testing, development

*Default:* production

*Example:*

```
computingservice_qualitylevel=production
```

**Warning:** CHANGE157: renamed

end of the `[infosys/glue2]` schema block #####

### **[infosys/glue2/ldap] block**

Enables the publication of the LDAP-rendering of the GLUE2 infomodel. The information tree is populated based on the contents of the schema-neutral `[infosys/cluster]` and `[queue:name]` blocks and the GLUE2 specific schema sub-blocks.

**Warning:** CHANGE158: new schema block. should be used to turn on/off GLUE2 LDAP tree publication.

`infosys_glue2_ldap` - Enables GLUE2 schema

**Warning:** CHANGE: DELETED. use block instead.

### **showactivities**

*Synopsis:* `showactivities = yes/no`

*Description:* (previously `infosys_glue2_ldap_showactivities`) Enables GLUE2 ComputingActivities to appear in the LDAP rendering

*Allowed values:* yes, no

*Default:* no

*Example:*

```
showactivities=no
```

**Warning:** CHANGE160: renamed.

### [infosys/glue1] block

This block enables the publication of GLUE1 LDAP representation of a CE. The information tree is populated based on the contents of the schema-neutral [infosys/cluster] and [queue:name] blocks and the GLUE1 specific schema sub-blocks. This block holds information that is needed by the glue1, in addition to the schema-neutral blocks.

**Warning:** CHANGE161: renamed. should be used to turn on/off GLUE1 publication.

infosys\_glue12 - Enables glue1.2/1.3 schema

**Warning:** CHANGE162: DELETED. use block instead

### resource\_location

*Synopsis:* resource\_location = string

*Description:* GLUE1 location attribute. IMPORTANT: no slashes or backslashes here!

*Default:* undefined

*Example:*

```
resource_location=Kastrup, Denmark
```

### resource\_latitude

*Synopsis:* resource\_latitude = latitude

*Description:* GLUE1 latitude.

*Default:* undefined

*Example:*

```
resource_latitude=55.75000
```

### resource\_longitude

*Synopsis:* resource\_longitude = longitude

*Description:* GLUE1 longitude.

*Default:* undefined

*Example:*

```
resource_longitude=12.41670
```

### cpu\_scaling\_reference\_si00

*Synopsis:* cpu\_scaling\_reference\_si00 = number

*Description:* GLUE1 CPU\_scaling

*Default:* undefined

*Example:*

```
cpu_scaling_reference_si00=2400
```

### processor\_other\_description

*Synopsis:* processor\_other\_description = string

*Description:* GLUE1 proc description

*Default:* undefined

*Example:*

```
processor_other_description=Cores 3, Benchmark 9.8-HEP-SPEC06
```

### glue\_site\_web

*Synopsis:* glue\_site\_web = url

*Description:* GLUE1 site web url

*Default:* undefined

*Example:*

```
glue_site_web=http://www.ndgf.org
```

### glue\_site\_unique\_id

*Synopsis:* glue\_site\_unique\_id = siteid

*Description:* GLUE1 site id

*Default:* undefined

*Example:*

```
glue_site_unique_id=NDGF-T1
```

provide\_glue\_site\_info yes/no - This variable decides if the GlueSite should be published.

**Warning:** CHANGE163: DELETED. enabled if [infosys/glue1/site-bdii] block exists, otherwise disabled.

### [infosys/glue1/site-bdii] block

Enable this block ONLY if you want to publish a semi-fake GLUE1 site-bdii as part of the LDAP server. This block is used to configure ARC to generate a semi-fake site-bdii that can be registered in GOCDB.

**Warning:** CHANGE164: renamed. turn on/off the fake site-bdii based on this block.

### unique\_id

*Synopsis:* \*unique\_id = id

*Description:* The unique id used to identify this site, eg NDGF-T1

*Default:* undefined

*Example:*

```
unique_id=NDGF-T1
```

### url

*Synopsis:* \*url = url

*Description:* The url of the resource BDII underlying the fake site bdii. It is on the format: `ldap://host.domain:2170/mds-vo-name=something,o=grid`.

*Default:* undefined

*Example:*

```
url=ldap://host.domain:2170/mds-vo-name=something,o=grid
```

Information schema-neutral blocks `[infosys/cluster]` and `[queue/NAME]` contain attributes that describe the computing cluster together with its queues. The parameters are available for every information model/schema representation.

### [infosys/cluster] block

This block describes the cluster characteristics of a Computing Element. The information specified here is mostly used by the Infosys ARC component.

### alias

*Synopsis:* alias = text

*Description:* An arbitrary alias name of the cluster, optional.

*Default:* undefined

*Example:*

```
cluster_alias=Big Blue Cluster in Nowhere
```

### hostname

*Synopsis:* hostname = fqdn

*Description:* Set the FQDN of the frontend.

*Default:* \$VAR{ [common]hostname }

*Example:*

```
hostname=myhost.org
```

### interactive\_contactstring

*Synopsis:* interactive\_contactstring = url

*Description:* the contact URL for interactive logins, set this if the cluster supports some sort of grid-enabled interactive login (gsi-ssh),

This option in **multivalued**.

*Default:* undefined

*Example:*

```
interactive_contactstring=gsissh://frontend.cluster:2200
```

### comment

*Synopsis:* comment = text

*Description:* Free text field for additional comments on the cluster in a single line, no newline character is allowed!

*Default:* undefined

*Example:*

```
comment=This cluster is specially designed for XYZ applications: www.xyz.org
```

### cluster\_location

*Synopsis:* cluster\_location = formatted\_string

*Description:* The geographical location of the cluster, preferably specified as a postal code with a two letter country prefix

*Default:* undefined

*Example:*

```
cluster_location=DK-2100
```

### cluster\_owner

*Synopsis:* cluster\_owner = text

*Description:* It can be used to indicate the owner of a resource, multiple entries can be used

This option in **multivalued**.

*Default:* undefined

*Example:*

```
cluster_owner=World Grid Project  
cluster_owner=University of NeverLand
```



## advertisedvo

*Synopsis:* advertisedvo = vo\_name

*Description:* (previously authorizedvo) This attribute is used to advertise which VOs are authorized on the cluster. Add only one VO for each advertisedvo entry. Multiple VOs in the same line will cause errors. These entries will be shown in all GLUE2 AccessPolicy and MappingPolicy objects, that is, they will apply for all End-points(Interfaces) and all Shares(currently queues). You can also add additional advertisedvos to queues. The information is also published in the Nordugrid schema.

---

**Note:**

**it is IMPORTANT to understand that this parameter is NOT enforcing any access control, it is just for information publishing!**

---

This option is **multivalued**.

*Default:* undefined

*Example:*

```
advertisedvo=atlas
advertisedvo=community.nordugrid.org
```

**Warning:** CHANGE: renamed it as advertisedvo

## clustersupport

*Synopsis:* clustersupport = email

*Description:* This is the support email address of the resource.

This option is **multivalued**.

*Default:* undefined

*Example:*

```
clustersupport=arc.support@mysite.org
clustersupport=arc.support@myproject.org
```

## homogeneity

*Synopsis:* homogeneity = True/False

*Description:* Determines whether the cluster consists of identical NODES with respect to cputype, memory, installed software (opsys). The frontend is NOT needed to be homogeneous with the nodes. In case of inhomogeneous nodes, try to arrange the nodes into homogeneous groups assigned to a queue and use queue-level attributes. False may trigger multiple GLUE2 ExecutionEnvironments to be published if applicable.

*Allowed values:* True, False

*Default:* True

*Example:*

```
homogeneity=True
```

## architecture

*Synopsis:* architecture = string

*Description:* Sets the hardware architecture of the NODES. The architecture is defined as the output of the `uname -m` (e.g. i686). Use this cluster attribute if only the NODES are homogeneous with respect to the architecture. Otherwise the queue-level attribute may be used for inhomogeneous nodes. If the frontend's architecture agrees to the nodes, the `adotf` (Automatically Determine On The Frontend) can be used to request automatic determination.

*Default:* adotf

*Example:*

```
architecture=adotf
```

## opsys

*Synopsis:* opsys = formatted\_string

*Description:* This multivalued attribute is meant to describe the operating system of the computing NODES. Set it to the opsys distribution of the NODES and not the frontend! opsys can also be used to describe the kernel or libc version in case those differ from the originally shipped ones. The distribution name should be given as `distrname-version.number`, where spaces are not allowed. Kernel version should come in the form `kernelname-version.number`. If the NODES are inhomogeneous with respect to this attribute do NOT set it on cluster level, arrange your nodes into homogeneous groups assigned to a queue and use queue-level attributes. opsys will be used to fill GLUE2 OSName, OSVersion and OSFamily unless these values are explicitly defined. See below for their usage.

This option is **multivalued**.

*Default:* undefined

**Warning:** TODO: check if adotf is supported for this parameter. If so, use that as default.

*Example:*

```
opsys=Linux-2.6.18
opsys=glibc-2.5.58
opsys=CentOS-5.6
```

## nodecpu

*Synopsis:* nodecpu = formatted\_string

*Description:* This is the cputype of the homogeneous nodes. The string is constructed from the `/proc/cpuinfo` as the value of `model name` and `@` and value of `cpu MHz`. Do NOT set this attribute on cluster level if the NODES are inhomogeneous with respect to cputype, instead arrange the nodes into homogeneous groups assigned to a queue and use queue-level attributes. Setting the `nodecpu=adotf` will result in Automatic Determination On The Frontend, which should only be used if the frontend has the same cputype as the homogeneous nodes.

*Default:* adotf

*Example:*

```
nodecpu=AMD Duron(tm) Processor @ 700 MHz
```

## nodememory

*Synopsis:* `nodememory = number`

*Description:* This is the amount of memory (specified in MB) on the node which can be guaranteed to be available for the application. Please note in most cases it is less than the physical memory installed in the nodes. Do NOT set this attribute on cluster level if the NODES are inhomogeneous with respect to their memories, instead arrange the nodes into homogeneous groups assigned to a queue and use queue-level attributes.

*Default:* undefined

*Example:*

```
nodememory=64000
```

## benchmark

*Synopsis:* `benchmark = name value`

*Description:* This optional multivalued attribute can be used to specify benchmark results on the cluster level. Use this cluster attribute if only the NODES are homogeneous with respect to the benchmark performance. Otherwise the similar queue-level attribute should be used. Please try to use standard benchmark names, if possible.

This option in **multivalued**.

*Default:* undefined

*Example:*

```
benchmark=SPECINT2000 222
benchmark=SPECFP2000 333
```

## middleware

*Synopsis:* `middleware = string`

*Description:* The multivalued attribute shows the installed grid software on the cluster. Nordugrid-ARC is automatically set, no need to specify

This option in **multivalued**.

*Default:* undefined

*Example:*

```
middleware=my software
```

## nodeaccess

*Synopsis:* `nodeaccess = inbound/outbound`

*Description:* Determines how the nodes can connect to the internet. Not setting anything means the nodes are sitting on a private isolated network. `outbound` access means the nodes can connect to the outside world while `inbound` access means the nodes can be connected from outside. `inbound & outbound` access together means the nodes are sitting on a fully open network.

This option in **multivalued**.

*Default:* undefined

*Allowed values:* `inbound, outbound`

*Example:*

```
nodeaccess=inbound
nodeaccess=outbound
```

## localse

*Synopsis:* localse = url

*Description:* This multivalued parameter tells the BROKER that certain URLs (and locations below that) should be considered locally available to the cluster.

This option is **multivalued**.

*Default:* undefined

*Example:*

```
localse=gsiftp://my.storage/data1/
localse=gsiftp://my.storage/data2/
```

gm\_mount\_point - this is the same as the path from the [gridftpd/jobs]

**Warning:** CHANGE150: DELETED. hardcode jobs in the infoproviders.

gm\_port - this is the same as the port from the [gridftpd] block. The

**Warning:** CHANGE151: DELETED. hardcode 2811 in the infoproviders.

## cpudistribution

*Synopsis:* cpudistribution = formatted\_string

*Description:* This is the CPU distribution over nodes given in the form ncpu:m where:

n is the number of CPUs per machine m is the number of such machines

*Example:* 1cpu:3,2cpu:4,4cpu:1 represents a cluster with 3 single CPU machines, 4 dual CPU machines and one machine with 4 CPUs.

*Default:* undefined

*Example:*

```
cpudistribution=1cpu:3,2cpu:4,4cpu:1
```

## [queue:name] block

Each grid-enabled queue on the cluster should be represented and described by a separate queue block. The queue\_name should be used as a label in the block name. In case of fork, or other LRMSes with no queue names, just use any unique string. A queue can represent a PBS/LSF/SGE/SLURM/LL queue, a SGE pool, a Condor pool or a single machine in case 'fork' type of LRMS. This block describes the queue characteristics.

name string - Sets the name of the grid-enabled queue. It MUST match the name label from the queue block header.

**Warning:** CHANGE187: DELETED. use the name in the block header instead!

### homogeneity

*Synopsis:* `homogeneity = True/False`

*Description:* determines whether the queue consists of identical NODES with respect to cputype, memory, installed software (opsys). In case of inhomogeneous nodes, try to arrange the nodes into homogeneous groups and assigned them to a queue. Possible values: True,False, the default is True.

*Allowed values:* True, False

*Default:* `$VAR{[infosys/cluster]homogeneity}`

*Example:*

```
homogeneity=True
```

### scheduling\_policy

*Synopsis:* `scheduling_policy = MAUI/FIFO`

*Description:* This optional parameter tells the scheduling policy of

**Warning:** CHANGE DELETED

### comment

*Synopsis:* `comment = text`

*Description:* A free-form text field for additional comments on the queue in a single line, no newline character is allowed!

*Default:* undefined

*Example:*

```
comment=This queue is nothing more than a condor pool
```

### pbs\_queue\_node

*Synopsis:* `pbs_queue_node = string`

*Description:* (previously `queue_node_string`) In PBS you can assign nodes to a queue (or a queue to nodes) by using the `node` property mark in PBS config.

Essentially, `pbs_queue_node` value is used to construct `nodes= string` in PBS script, such as `nodes=count:pbs_queue_node` where `count` is taken from the job description (1 if not specified).

This corresponds to setting the following parameter in PBS for this queue:

```
resources_default.neednodes = cpu_topology[:pbs_queue_node]
```

Setting the `pbs_queue_node` changes how the `queue-totalcpus`, `user freecpus` are determined for this queue.

You shouldn't use this option unless you are sure that your PBS configuration makes use of the above configuration. Read NorduGrid PBS instructions for more information: <http://www.nordugrid.org/documents/pbs-config.html>

*Default:* undefined

*Example:*

```
pbs_queue_node=gridlong_nodes
pbs_queue_node=ppn=4:ib
```

**Warning:** CHANGE152: renamed

### sgе\_jobopts

*Synopsis:* sge\_jobopts = string

*Description:* Per-queue override of additional SGE options to be used when submitting jobs to SGE to this queue

*Default:* undefined

*Example:*

```
sge_jobopts=-P atlas -r yes
```

### condor\_requirements

*Synopsis:* condor\_requirements = string

*Description:* It needs to be defined for each Condor queue. Use this option to determine which nodes belong to the current queue. The value of `condor_requirements` must be a valid constraints string which is recognized by a `condor_status -constraint ...` command. It can reference pre-defined ClassAd attributes (like Memory, Opsy, Arch, HasJava, etc) but also custom ClassAd attributes. To define a custom attribute on a condor node, just add two lines like the ones below in the `$(hostname).local` config file on the node:

```
NORDUGRID_RESOURCE=TRUE
STARTD_EXPRS = NORDUGRID_RESOURCE, $(STARTD_EXPRS)
```

A job submitted to this queue is allowed to run on any node which satisfies the `condor_requirements` constraint. If `condor_requirements` is not set, jobs will be allowed to run on any of the nodes in the pool. When configuring multiple queues, you can differentiate them based on memory size or disk space, for example.

*Default:* `$VAR{[lrms]condor_requirements}`

*Example:*

```
condor_requirements=(OpSys == "linux" && NORDUGRID_RESOURCE && Memory >= 1000 &&
↪Memory < 2000)
```

### totalcpus

*Synopsis:* totalcpus = number

*Description:* Manually sets the number of cpus assigned to the queue. No need to specify the parameter in case the `queue_node_string` method was used to assign nodes to the queue (this case it is dynamically calculated and the static value is overwritten) or when the queue have access to the entire cluster (this case the cluster level `totalcpus` is the relevant parameter).

*Default:* undefined

*Example:*

```
totalcpus=32
```

queue-level configuration parameters: `nodecpu`, `nodememory`, `architecture`, `opsys` and `benchmark` should be set if they are homogeneous over the nodes assigned to the queue AND they are different from the cluster-level value. Their meanings are described in the `[infosys/cluster]` block. Usage: this queue collects nodes with `nodememory=512` while another queue has nodes with `nodememory=256` -> don't set the cluster attributes but use the queue-level attributes. When the frontend's architecture or `cputype` agrees with the queue nodes, the `adotf` (Automatically Determine On The Frontend) can be used to request automatic determination of architecture or `nodecpu`. For GLUE2, fine tune configuration of `OSName`, `OSVersion`, `OSFamily` is allowed in a dedicated subblock.

## nodecpu

*Synopsis:* `nodecpu = formatted_string`

*Description:* see description at `[infosys/cluster]` block

*Default:* `$VAR{[infosys/cluster]nodecpu}`

*Example:*

```
nodecpu=AMD Duron(tm) Processor @ 700 MHz
```

## nodememory

*Synopsis:* `nodememory = number`

*Description:* see description at `[infosys/cluster]` block

*Default:* `$VAR{[infosys/cluster]nodememory}`

*Example:*

```
nodememory=512
```

## defaultmemory

*Synopsis:* `defaultmemory = number`

*Description:* The LRMS memory request of job to be set by the LRMS backend scripts, if a user submits a job without specifying how much memory should be used. The order of precedence is: job description -> `[lrms-defaultmemory]` -> `[queue-defaultmemory]`. This is the amount of memory (specified in MB) that a job will request.

*Default:* undefined

*Example:*

```
defaultmemory=512
```

**Warning:** CHANGE: new parameter in this block

## architecture

*Synopsis:* architecture = string

*Description:* see description at [infosys/cluster] block

*Default:* \$VAR{[infosys/cluster]architecture}

*Example:*

```
architecture=adotf
```

## opsys

*Synopsis:* opsys = formatted\_string

*Description:* see description at [infosys/cluster] block

This option in **multivalued**.

*Default:* \$VAR{[infosys/cluster]opsys}

*Example:*

```
opsys=Linux-2.6.18  
opsys=glibc-2.5.58
```

## osname

*Synopsis:* osname = string

*Description:* Only for GLUE2

This option in **multivalued**.

*Default:* undefined

*Example:*

```
osname=Ubuntu
```

## osversion

*Synopsis:* osversion = string

*Description:* Only for GLUE2

*Default:* undefined

*Example:*

```
osversion=12.04
```

## osfamily

*Synopsis:* osfamily = string

*Description:* Only for GLUE2

*Default:* undefined

*Example:*



```
osfamily=linux
```

## benchmark

*Synopsis:* benchmark = name value

*Description:* see description at [infosys/cluster] block

This option is **multivalued**.

*Default:* \$VAR{[infosys/cluster]benchmark}

*Example:*

```
benchmark=SPECINT2000 222
benchmark=SPECFP2000 333
```

## allowaccess

*Synopsis:* allowaccess = authgroup [authgroup ...]

*Description:* Specifies authorization groups that are allowed to submit jobs to this queue. It does not override similar parameter specified in [gridftpd/jobs] and [arex/ws/emies] blocks. If this parameter is not set, then no additional restrictions are applied for authorizing user ability to submit jobs to this queue. The matching group from this parameter does not affect which VOMS attributes are selected as representing the user.

*Default:* undefined

*Example:*

```
allowaccess=biouers atlasusers
```

**Warning:** CHANGE214: NEW parameter in the [queue:name] block

## advertisedvo

*Synopsis:* advertisedvo = vo\_name

*Description:* (previously authorizedvo) This attribute is used to advertise which VOs are authorized on the [queue:name] of the cluster. Add only one VO for each advertisedvo entry. Multiple VOs in the same line will cause errors. These entries will be shown in the MappingPolicy objects, that is, they will apply for the Shares that corresponds to the queue. The information is also published in the Nordugrid schema.

---

**Note:** if you have also configured advertisedvo in the [infosys/cluster] block, the result advertised VOs per queue is the union set of what is contained in the [infosys/cluster] and in this [queue:name] block!

---



---

**Note:** it is IMPORTANT to understand that this parameter is NOT enforcing any access control, it is just for information publishing!

---

This option is **multivalued**.

*Default:* undefined

*Example:*

```
advertisedvo=atlas
advertisedvo=community.nordugrid.org
```

**Warning:** CHANGE: renamed it as advertisedvo

### maxslotsperjob

*Synopsis:* maxslotsperjob = number

*Description:* This GLUE2 specific parameter configures the MaxSlotsPerJob value on a particular queue. This value is usually generated by LRMS infocollectors, but there are cases in which a system administrator might like to tweak it. Default is to publish what is returned by the LRMS, and if nothing is returned, NOT to publish the MaxSlotsPerJob attribute. If a system administrator sets the value here, that value will be published instead, regardless of what the LRMS returns. Each LRMS might have a different meaning for this value.

*Default:* undefined

*Example:*

```
maxslotsperjob=5
```

### forcedefaultvoms

*Synopsis:* forcedefaultvoms = VOMS\_FQAN

*Description:* specify VOMS FQAN which user will be assigned if his/her credentials contain no VOMS attributes.

*Default:* \$VAR{ [arex] forcedefaultvoms }

*Example:*

```
forcedefaultvoms=/vo/group/subgroup
```

**Warning:** CHANGE: documented in [queue] as described in [arex]

cachetime - LDAP parameters of the queue+jobs+users.pl (old) infoprovider, use the defaults,

**Warning:** CHANGE153: DELETED.

sizelimit - affects registration to EGIIS

**Warning:** CHANGE154: DELETED

**Warning:** TODO: add Florido's new variables both to the queue and cluster block (maxcputime and maxwall-time)

**[datadelivery-service] block**

This block configures and enables the data delivery service. This service is intended to off-load data-staging from A-REX and usually deployed on one or more separate machines. See details at [https://wiki.nordugrid.org/wiki/Data\\_Staging/Multi-host](https://wiki.nordugrid.org/wiki/Data_Staging/Multi-host)

This service can also act as an independent data transfers service that case it would require a intelligent data manager that could replace A-REX's intelligence. (see draft vision document)

**Warning:** TODO: point to the documentation that describes protocol, operations.

**Warning:** CHANGE174: NEW block

**transfer\_dir**

*Synopsis:* \*transfer\_dir = path

*Description:* (previously allowed\_dir) The directori(es) on the DDS host in which the service is allowed to read and write. When DDS is used as a remote transfer service assisting A-REX then this is usually one or more cache and/or session directories shared as a common mount with A-REX.

This option in **multivalued**.

*Default:* undefined

*Example:*

```
transfer_dir=/shared/arc/cache
transfer_dir=/shared/arc/session
```

**Warning:** CHANGE184: NEW, also renamed from allowed\_dir to work\_dir

**Warning:** TODO renamed

**hostname**

*Synopsis:* hostname = FQDN

*Description:* The hostname of the machine on which DDS service runs.

*Default:* \$EXEC{hostname -f}

*Example:*

```
hostname=localhost
```

**Warning:** CHANGE175: NEW

**Warning:** TODO: got remaned from interface, fix DDS startup script and profile.

## port

*Synopsis:* port = port

*Description:* Port on which service listens

*Default:* 443

*Example:*

```
port=8443
```

**Warning:** CHANGE176: NEW

## ipversion

*Synopsis:* ipversion = number

*Description:* IP version to use

*Default:* 4

*Example:*

```
ipversion=6
```

**Warning:** TODO: Ask Aleksandr if can we have both 4 and 6 together (multivalued)?

**Warning:** CHANGE177: NEW

## pidfile

*Synopsis:* pidfile = path

*Description:* pid file of the daemon

*Default:* /var/run/arc/datadelivery-service.pid

**Warning:** TODO: changed the pid file name and location!

*Example:*

```
pidfile=/tmp/delivery.pid
```

**Warning:** CHANGE178: NEW

## logfile

*Synopsis:* logfile = path

*Description:* log file of the daemon

*Default:* /var/log/arc/datadelivery-service.log

*Example:*

```
logfile=/tmp/delivery.log
```

**Warning:** CHANGE179: NEW

## loglevel

*Synopsis:* loglevel = level

*Description:* set loglevel of the data delivery service between 0 (FATAL) and 5 (DEBUG). Defaults to 3 (INFO).

*Allowed values:* 0, 1, 2, 3, 4, 5

*Default:* 3

*Example:*

```
loglevel=4
```

**Warning:** CHANGE179b: NEW

## user

*Synopsis:* user = username

*Description:* Overwrites the user under which the service runs. The default is the user starting the service. DDS is very limited if not run as root.

*Default:* undefined

*Example:*

```
user=ddsuser
```

**Warning:** CHANGE180: NEW

## secure

*Synopsis:* secure = yes/no

*Description:* Set to no if the service should run without a host certificate. In this case the corresponding deliveryservice option in the [arex/data-staging] A-REX configuration block should use http rather than https URLs.

*Allowed values:* yes, no

*Default:* yes

*Example:*

```
secure=no
```

**Warning:** CHANGE181: NEW

### allowed\_ip

*Synopsis:* \*allowed\_ip = ip

*Description:* IP address authorized to access service. Normally this is the A-REX host IP.

This option in **multivalued**.

*Default:* undefined

*Example:*

```
allowed_ip=192.0.2.1
```

**Warning:** CHANGE182: NEW

### allowed\_dn

*Synopsis:* allowed\_dn = DN

*Description:* DN authorized to access service. This option restricts access to specified DNs (of the users who submit jobs to A-REX). It is only effective if secure=yes.

This option in **multivalued**.

*Default:* undefined

*Example:*

```
allowed_dn=/O=Grid/O=Big VO/CN=Main Boss
```

**Warning:** CHANGE183: NEW

### x509\_host\_key

*Synopsis:* x509\_host\_key = path

*Description:* Optional parameter to overwrite [common] block values.

*Default:* \$VAR{ [common]x509\_host\_key }

*Example:*

```
x509_host_key=/etc/grid-security/hostkey.pem
```

**Warning:** CHANGE: new in this block

**x509\_host\_cert**

*Synopsis:* x509\_host\_cert = path

*Description:* Optional parameter to overwrite [common] block values.

*Default:* \$VAR{ [common]x509\_host\_cert }

*Example:*

```
x509_host_cert=/etc/grid-security/hostcert.pem
```

**Warning:** CHANGE: new in this block

**x509\_cert\_dir**

*Synopsis:* x509\_cert\_dir = path

*Description:* Optional parameter to overwrite [common] block values.

*Default:* \$VAR{ [common]x509\_cert\_dir }

*Example:*

```
x509_cert_dir=/etc/grid-security/certificates
```

**Warning:** CHANGE: new in this block

**[acix-scanner] block**

The ARC Cache Index (ACIX) is a distributed system that maintains a catalog of locations of cached files stored in various A-REX caches. ACIX consists of two components, the Cache Scanner (on the CEs), and the Index Server.

This config block enables and configures the cache scanner component of ACIX. The scanning component of ACIX is a separate service that runs alongside A-REX/DDS and all it needs from A-REX/DDS service is the location of the cache.

The acix-scanner periodically scans the cache directories and composes a Bloom filter of A-REX cache content which can be pulled via its public interface. In the current deployment the ACIX index services are the main consumers of the collected information.

**Warning:** CHANGE82: renamed block, renamed component (cache scanner, instead of cacheserver)

**cachedir**

*Synopsis:* cachedir = cache\_path

*Description:* Specifies the cache directory to be scanned in case not set in [arex/cache] block (e.g. the scanner is running on a different machine than A-REX)

This option is **multivalued**.

*Default:* \$VAR{ [arex/cache]cachedir }

*Example:*

```
cachedir=/scratch/cache  
cachedir=/fs1/cache drain
```

## logfile

*Synopsis:* logfile = path

*Description:* Log file location for the acix-scanner.

*Default:* /var/log/arc/arc-acix-scanner.log

*Example:*

```
logfile=/tmp/arc-acix-scanner.log
```

**Warning:** CHANGE83: modified default for the logfile

**Warning:** TODO: changed logfile name

## hostname

*Synopsis:* hostname = string

*Description:* Hostname on which the acix-scanner listens

*Default:* \$EXEC{hostname -f}

*Example:*

```
hostname=myhost.org
```

## port

*Synopsis:* port = port

*Description:* Port on which the acix-scanner service listens

*Default:* 5443

*Example:*

```
port=6000
```

## cachedump

*Synopsis:* cachedump = yes/no

*Description:* Whether to make a dump of the list of files on the cache at \$TMP/ARC-ACIX/timestamp each time the acix-scanner runs.

*Default:* no

*Allowed values:* yes, no

*Example:*



```
cachedump=yes
```

### x509\_host\_key

*Synopsis:* x509\_host\_key = path

*Description:* Optional parameter to overwrite [common] block values.

*Default:* \$VAR{ [common]x509\_host\_key }

*Example:*

```
x509_host_key=/etc/grid-security/hostkey.pem
```

**Warning:** CHANGE: new in this block

### x509\_host\_cert

*Synopsis:* x509\_host\_cert = path

*Description:* Optional parameter to overwrite [common] block values.

*Default:* \$VAR{ [common]x509\_host\_cert }

*Example:*

```
x509_host_cert=/etc/grid-security/hostcert.pem
```

**Warning:** CHANGE: new in this block

### x509\_cert\_dir

*Synopsis:* x509\_cert\_dir = path

*Description:* Optional parameter to overwrite [common] block values.

*Default:* \$VAR{ [common]x509\_cert\_dir }

*Example:*

```
x509_cert_dir=/etc/grid-security/certificates
```

**Warning:** CHANGE: new in this block

### [acix-index] block

The ARC Cache Index (ACIX) is a distributed system that maintains a catalog of locations of cached files stored in various A-REX caches. ACIX consists of two components, the Cache Scanner (on the CEs), and the Index Server. This config block enables and configures the index server component of ACIX.

The index server component of ACIX collects cache content filters generated by a set of acix-scanners and maintains an aggregated view of distributed cache contents.

The acix-index server is deployed separately and can be queried for the location of cached files. The service endpoint is <https://hostname:6443/data/index> and a query is performed via giving the URLs to check as comma-separated values to the `url` option, e.g.: `index_service_endpoint?url=http://www.nordugrid.org:80/data/echo.sh,http://my.host/data1`

### cachescanner

*Synopsis:* `*cachescanner = url`

*Description:* (previously cacheserver) ACIX cache scanners from which to pull information

This option is **multivalued**.

*Default:* undefined

*Example:*

```
cachescanner=https://some.host:5443/data/cache
cachescanner=https://another.host:5443/data/cache
```

**Warning:** CHANGE84: renamed

**Warning:** TODO: should be mandatory

### x509\_host\_key

*Synopsis:* `x509_host_key = path`

*Description:* Optional parameter to overwrite [common] block values.

*Default:* `$VAR{ [common]x509_host_key }`

*Example:*

```
x509_host_key=/etc/grid-security/hostkey.pem
```

**Warning:** CHANGE: new in this block

### x509\_host\_cert

*Synopsis:* `x509_host_cert = path`

*Description:* Optional parameter to overwrite [common] block values.

*Default:* `$VAR{ [common]x509_host_cert }`

*Example:*

```
x509_host_cert=/etc/grid-security/hostcert.pem
```

**Warning:** CHANGE: new in this block

**x509\_cert\_dir**

*Synopsis:* x509\_cert\_dir = path

*Description:* Optional parameter to overwrite [common] block values.

*Default:* \$VAR{ [common]x509\_cert\_dir }

*Example:*

```
x509_cert_dir=/etc/grid-security/certificates
```

**Warning:** CHANGE: new in this block

**[nordugridmap] block**

This optional block is used to fine-tune the behavior of the nordugridmap utility - an ARC tool used to generate grid-mapfiles. Normal setups don't need to configure this block. Please refer to [userlist/name] block above to find information how to specify sources for userlist generation. This section setup general source-independent parameters.

**x509\_host\_key**

*Synopsis:* x509\_host\_key = path

*Description:* Optional parameter to overwrite [common] block values.

*Default:* \$VAR{ [common]x509\_host\_key }

*Example:*

```
x509_host_key=/etc/grid-security/hostkey.pem
```

**Warning:** CHANGE24: renamed

**x509\_host\_cert**

*Synopsis:* x509\_host\_cert = path

*Description:* Optional parameter to overwrite [common] block values.

*Default:* \$VAR{ [common]x509\_host\_cert }

*Example:*

```
x509_host_cert=/etc/grid-security/hostcert.pem
```

**Warning:** CHANGE25: renamed

**x509\_cert\_dir**

*Synopsis:* x509\_cert\_dir = path

*Description:* Optional parameter to overwrite [common] block values.

*Default:* \$VAR{ [common]x509\_cert\_dir }

*Example:*

```
x509_cert_dir=/etc/grid-security/certificates
```

### gridmap\_owner

*Synopsis:* gridmap\_owner = username

*Description:* The owner of the generated mapfiles.

*Default:* root

*Example:*

```
gridmap_owner=root
```

### gridmap\_group

*Synopsis:* gridmap\_group = groupname

*Description:* The group of generated gridmapfiles.

*Default:* root

*Example:*

```
gridmap_group=root
```

### gridmap\_permissions

*Synopsis:* gridmap\_permissions = filemode

*Description:* The permissions of generated gridmapfiles.

*Default:* 0600

*Example:*

```
gridmap_permissions=0600
```

generate\_vomapfile - control is nordugridmap will generate vo-mapfile used by arc-ur-logger. Default is 'yes'.

**Warning:** CHANGE26: DELETED, we don't support arc-ur-logger any longer. don't generate that file any longer either.

vomapfile - path to vo-mapfile location.

**Warning:** CHANGE27: DELETED, we don't support arc-ur-logger any longer. don't generate that file any longer either.

## log\_to\_file

*Synopsis:* log\_to\_file = yes/no

*Description:* control whether logging output of nordugridmap will be saved to file. If the value is 'no' nordugridmap will write all information to STDERR.

*Allowed values:* yes, no

*Default:* yes

*Example:*

```
log_to_file=no
```

## logfile

*Synopsis:* logfile = path

*Description:* specify the nordugridmap log file location when log\_to\_file is set yes.

*Default:* /var/log/arc/nordugridmap.log

*Example:*

```
logfile=/var/log/arc/nordugridmap.log
```

## cache\_enable

*Synopsis:* cache\_enable = yes/no

*Description:* Controls whether caching of external sources will be used.

*Allowed values:* yes, no

*Default:* yes

*Example:*

```
cache_enable=yes
```

## cachedir

*Synopsis:* cachedir = path

*Description:* Specifies the path where cached sources will be stored.

*Default:* /var/spool/arc/gridmapcache/

*Example:*

```
cachedir=/var/spool/arc/gridmapcache/
```

**Warning:** TODO changed file location

### cachetime

*Synopsis:* cachetime = seconds

*Description:* Controls how long (in seconds) the cached information remains valid. Default is 259200 (3 days).

*Default:* 259200

*Example:*

```
cachetime=259200
```

issuer\_processing - control the behavior of [userlist/name] block's require\_issuerdn

**Warning:** CHANGE28: DELETED

### mapuser\_processing

*Synopsis:* mapuser\_processing = overwrite/keep

*Description:* Controls the behavior of [userlist:name] block's mapped\_unixid parameter usage. Please see 'mapped\_unixid' description in [userlist:name] block for details.

*Allowed values:* keep, overwrite

*Default:* keep

*Example:*

```
mapuser_processing=keep
```

### allow\_empty\_unixid

*Synopsis:* allow\_empty\_unixid = yes/no

*Description:* Controls whether empty (or unspecified) 'mapped\_unixid' [userlist:name] block option is allowed to be used. Please see 'mapped\_unixid' description for details.

*Allowed values:* yes, no

*Default:* yes

*Example:*

```
allow_empty_unixid=no
```

**Warning:** CHANGE29: MODIFIED. set default to yes.

### voms\_method

*Synopsis:* voms\_method = soap/get

*Description:* Controls how to get information from VOMS(S) sources. Valid values are:

soap - call SOAP method directly using SOAP::Lite  
get - use old implementation that manually parses XML response

*Allowed values:* soap, get

*Default:* soap

*Example:*

```
voms_method=soap
```

## loglevel

*Synopsis:* loglevel = level

*Description:* (previously debug) Controls the verbosity of nordugridmap output. Valid values are:

- 0 - FATAL - only critical fatal error shown
- 1 - ERROR - errors, including non-critical are shown
- 2 - WARNING (default) - configuration errors that can be ignored
- 3 - INFO - processing information
- 4 - VERBOSE - a bit more processing information
- 5 - DEBUG - lot of processing information

When test run is requested (`-test` command line option of the nordugridmap) loglevel is automatically set to 5 (DEBUG).

*Allowed values:* 0, 1, 2, 3, 4, 5

*Default:* 2

*Example:*

```
loglevel=4
```

**Warning:** CHANGE30: renamed as loglevel

**Warning:** TODO set default to 2

## fetch\_timeout

*Synopsis:* fetch\_timeout = seconds

*Description:* Controls how many time (in seconds) nordugridmap will wait for external sources retrieval.

*Default:* 15

*Example:*

```
fetch_timeout=15
```

Services registration into EMIR block configures and enables the registration process of a services enabled in this configuration file into EMI indexing service (EMIR). Currently only implemented for A-REX.

**Warning:** CHANGE165: DELETED. entire emir registration block is removed from `arc.conf`. check for code cleanup as well.

emiurls url - List of URL separated by comma of EMIR services which are to accept

**Warning:** CHANGE166: DELETED

validity sec - Time in seconds for which registration records should stay valid.

**Warning:** CHANGE167: DELETED

period sec - Time in seconds how often registration record should be sent to the

**Warning:** CHANGE168: DELETED

disablereg\_xbes yes/no - disablereg\_xbes may be used to selectively disable registration of

**Warning:** CHANGE169: DELETED

### **[infosys/index/indexname] block**

WARNING: the ldap-based Info Index Service, together with the registration is OBSOLETE technology, therefore the new config format is not supporting GIIS. Use earlier ARC releases if you want to setup an EGIS server.

**Warning:** CHANGE191: DELETED block, the entire block is deleted.

name - The unique (within the hosting machine) name of the

**Warning:** CHANGE192: DELETED

allowregistration - Implements registration filtering within an Index Service

**Warning:** CHANGE193: DELETED

### **[infosys/index/indexname/registration/registrationname] block**

WARNING: the ldap-based registration is OBSOLETE technology therefore the new config is not supporting setting up GIIS services. Use previous ARC releases for that.

**Warning:** CHANGE194: DELETED. entire config block is deleted.

targethostname - the hostname of the machine running the registration target

**Warning:** CHANGE195: DELETED

targetport - the port on which the target Index Service is running.

**Warning:** CHANGE196: DELETED

targetsuffix - the LDAP suffix of the target Index Service



**Warning:** CHANGE197: DELETED

regperiod - The registration period in seconds, the registration messages are

**Warning:** CHANGE198: DELETED

registranthostname - the hostname of the machine sending the registrations.

**Warning:** CHANGE199: DELETED

registrantport - the port of the slapd service hosting the

**Warning:** CHANGE200: DELETED

registrantsuffix - the LDAP suffix of the registrant Index Service.

**Warning:** CHANGE201: DELETED

timeout - The suggested timeout to be included in the registration.

**Warning:** CHANGE202: DELETED

ttl - The suggested TTL to be included in the registration. The default

**Warning:** CHANGE203: DELETED

### [infosys/cluster/registration/NAME] block

WARNING: the ldap-based registration is OBSOLETE technology

**Warning:** CHANGE213: DELETED block, all the config options are OBSOLETE and DELETED in this block

### registrationlog

*Synopsis:* registrationlog = path

*Description:* specifies the logfile for the registration processes initiated by your machine.

**Warning:** CHANGE212: DELETED

### sizelimit

*Synopsis:* sizelimit = number

*Description:* sizelimit affects registration to egiis The value is reported back by the index server as Mds-Service-Ldap-sizelimit

**Warning:** CHANGE212: DELETED

### targetgiis

*Synopsis:* \*targetgiis = name

*Description:* the name of the index service to which the registration to be sent.

**Warning:** CHANGE212: DELETED

### targethostname

*Synopsis:* \*targethostname = hostname

*Description:* the hostname of the machine running the registration target

**Warning:** CHANGE204: DELETED

### targetport

*Synopsis:* targetport = number

*Description:* the port on which the target Index Service is running.

**Warning:** CHANGE205: DELETED

### targetsuffix

*Synopsis:* targetsuffix = string

*Description:* the LDAP suffix of the target Index Service

**Warning:** CHANGE206: DELETED

### regperiod

*Synopsis:* regperiod = seconds

*Description:* The registration period in seconds, the registration messages are

**Warning:** CHANGE207: DELETED

### registranthostname

*Synopsis:* registranthostname = hostname

*Description:* the hostname of the machine sending the registrations.

**Warning:** CHANGE208: DELETED

### registrantport

*Synopsis:* registrantport = port

*Description:* the port of the slapd service hosting the registrant Index Service. The attribute inherits its value from the

**Warning:** CHANGE209: DELETED

### registrantsuffix

*Synopsis:* registrantsuffix = string

*Description:* the LDAP suffix of the registrant cluster resource It is automatically determined from the [infosys] block and the

**Warning:** CHANGE210: DELETED

### ttl

*Synopsis:* ttl = seconds

*Description:* The suggested TTL to be included in the registration. The default

**Warning:** CHANGE211: DELETED

## 1.4 Working with RunTime Environments in ARC6

### 1.4.1 Understanding RunTime Environments

ARC Computing Element is a front-end to the various heterogeneous resource providers. To run jobs on the particular resource provider there are always set of software or workflow-specific paths, tools, libraries, environmental variables or even dynamic content that should be recreated in the job content.

To provide the flexible way of job runtime environment tuning ARC enforces the concept of the RunTime Environment (RTE).

ARC RunTime Environments (RTEs) provides the two functions:

**Advertising** indicate the available environment to be requested by end-users

**Modifying job environment** flexibly contextualize job execution environment

### Advertising RTEs

Advertising RTEs provide user interfaces to application software and other resources in a way that is independent of the details of the local installation of the application and computing platform (OS, hardware, etc.).

It addresses setups typically required by large research groups or user bases, dealing with a common set of software. The actual implementation of particular RTE may differ from site to site as necessary.

However, it should be designed so that resource providers with different accounting, licence or other site-specific implementation details can advertise the same application interface (RTE) for all users.

Despite possibly different parameters or implementation, the same software addressed by the same RTE name known by community. It is also supported to add RTE versioning at the end of RTE name (after dash). RTE version will be used for resource matchmaking along with RTE name.

For example to request ENV/PROXY and APPS/HEP/ATLAS with version 20.1.0.1 or greater in *xRSL* job description:

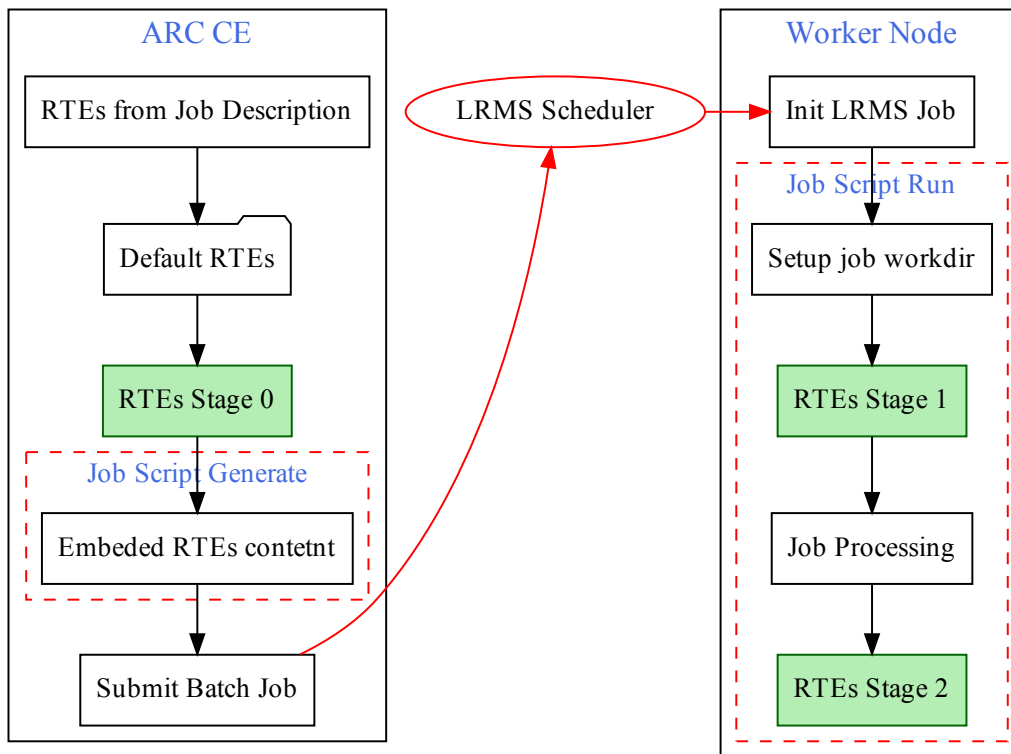
```
(runTimeEnvironment="ENV/PROXY")  
(runTimeEnvironment>="APPS/HEP/ATLAS-20.1.0.1")
```

It is always up to the local system administrators to take a decision whether to *install* and *enable* a particular RTE or not.

### Modifying job environment

The RTE content itself is a BASH script that aimed to run any arbitrary code during the job life cycle.

The first argument of RTE script indicate the so-called RTE stage. If the job description specifies additional arguments for corresponding RTE those are appended starting at second position.



There are 3 stages of RTE execution:

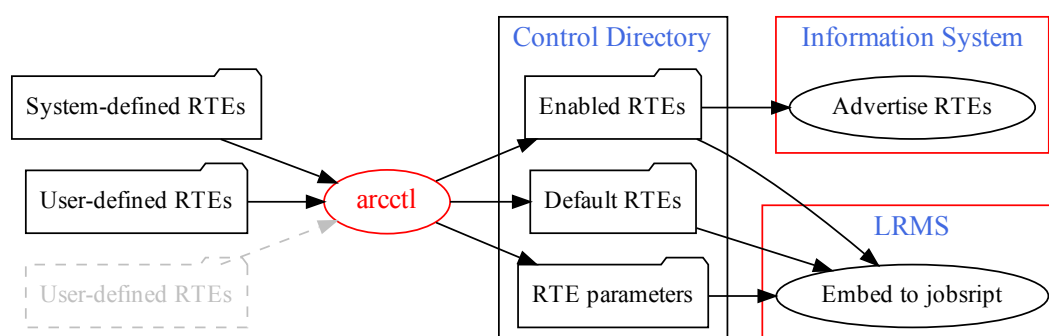
**Stage 0** RTE script sourced before the creation of the job's LRMS submission script. In this case the scripts are run by A-REX on the frontend (ARC CE), before the job is sent to the LRMS. Some environment variables are defined in this case, and can be changed to influence the job's execution later. *TODO: list of grami attributes as a dedicated technical note*

**Stage 1** Embedded RTE function runs before the main job processing on the Worker Node under the LRMS. Such stage can prepare the environment for some third-party software package. The current directory in this case is the one which would be used for execution of the job. Variable `$HOME` also points to this directory.

**Stage 2** Embedded RTE function runs after the main job processing on the Worker Node under the LRMS. The main purpose is to clean possible changes done by *Stage 1* (like removing temporary files).

## 1.4.2 Operating RunTime Environments

In ARC6 release operating RunTime Environments is changed significantly and rely on *ARC Control Tool*:



## Installing RTE scripts

There are set of *System-defined RTEs* pre-installed with the ARC CE packages that aimed to fulfill common workflows.

ARC CE administrator can add additional RTE directories (so-called *User-defined RTEs*). This additional places should be specified in `arc.conf` using the `runtime_dir` configuration option in `[arex]` block.

---

**Note:** In ARC6 directories with RTE script are local to ARC CE and **SHOULD NOT** be shared with worker nodes

---

The RTE names used for *advertising* are implied by directories structure, e.g. in the ENV/PROXY the ENV is a directory inside *System RTEs* location and PROXY is a name of file.

## Enabling RTEs

*Installed RTEs* should be enabled to be *advertised* and used during the job submission.

### By name

To enable particular RTE by name run the following command:

```
arcctl rte enable ENV/PROXY
```

### By path

Especially if you have several RTEs with the same name installed, you can choose the exact one by specifying filesystem path:

```
arcctl rte enable /usr/share/arc/rte/ENV/PROXY
```

## Using wildcards

To enable several RTEs you can pass as many names as you want to the `arcctl` command. Additionally you can use glob (`man 7 glob`) wildcards in RTE names.

The following command will enable all APPS/HEP/ATLAS RTEs for SLC6 builds:

```
arcctl rte enable APPS/HEP/ATLAS-*--X86_64-SLC6-*
```

**Note:** It is advised to use wildcards to enable all user-defined RTEs during ARC5 to ARC6 upgrade

## Dummy RTEs

In case you need RTEs only for *advertising* but no need for script content, you can create *Dummy RTE* for specified name. The following command enables APPS/MYAPP RTE with empty content:

```
arcctl rte enable APPS/MYAPP --dummy
```

## Default RTEs

Default RTEs aimed to address the workflows when *advertising* and implicit request in job description is not needed, however *modification* of every submitted job (adjusting memory, setting LRMS scratch, etc) is required on the resource provider.

*Installed RTEs* can be selected for default inclusion to job lifecycle with the following *ARC Control Tool* command:

```
arcctl rte default ENV/LRMS-SCRATCH
```

This will made ENV/LRMS-SCRATCH transparently added to each job and executed the same way as *Enabled RTEs*.

**Note:** You can use the same *by-name*, *by-path* and *wildcard* techniques as for *enabling*

## RTE Parameters

To achieve various cases due to heterogeneity of resource providers, some RTEs can be parametrized.

For example, the system-defined ENV/PROXY RTE that transfer the delegated proxy-certificate to the worker node can optionally transfer CA certificate directories. This optional part is controlled by COPY\_CACERT\_DIR parameter.

To check either RTE contains parameters and their default values, run the:

```
[root ~]# arcctl rte params-get ENV/PROXY
COPY_CACERT_DIR=Yes
```

You can also see the description and allowed values adding `--long` keyword.

To set RTE parameter value, the following command should be used:

```
arcctl rte params-set ENV/PROXY COPY_CACERT_DIR No
```

## List available RTEs and their status

To view the summary of all *installed*, *enabled* and *default* RTEs run:

```
[root ~]# arcctl rte list
<output omitted>
APPS/HEP/ATLAS-20.8.0-X86_64-SLC6-GCC48-OPT (user, enabled)
```

```

APPS/HEP/ATLAS-20.8.1-X86_64-SLC6-GCC48-OPT (user, enabled)
APPS/HEP/ATLAS-20.8.2-X86_64-SLC6-GCC49-OPT (user, enabled)
<output omitted>
ENV/LRMS-SCRATCH (system, default)
ENV/PROXY (system, masked, disabled)
ENV/PROXY (user, enabled)
ENV/RTE (system, disabled)
ENV/RUNTIME/ALIEN-2.17 (user, enabled)
VO-biomed-CVMFS (dummy, enabled)

```

The first tag describe RTE origin (system, user or *dummy*). The following tags shows the status.

The special masked keyword indicates that RTE name used more that once and *by-name* operations will apply to other RTE script. In example ENV/PROXY will be enabled from user-defined location and system-defined will be masked. However it is possible to enable masked RTE *by path*.

Listing the particular kind of RTEs (e.g. *enabled*) is possible with appropriate argument (see *ARC Control Tool* for all available options):

```

[root ~]# arcctl rte list --enabled
<output omitted>
APPS/HEP/ATLAS-20.8.2-X86_64-SLC6-GCC49-OPT
ENV/PROXY
ENV/RUNTIME/ALIEN-2.17
VO-biomed-CVMFS

```

The *long* listing allows to get the detailed pointers to RTEs locations and descriptions:

```

[root ~]# arcctl rte list --long
System pre-defined RTEs in /usr/share/arc/rte:
  ENV/PROXY # copy proxy certificate to the job session_
  ↪directory
  ENV/RTE # copy RunTimeEnvironment scripts to the job_
  ↪session directory
  ENV/LRMS-SCRATCH # enables the usage of local to WN scratch_
  ↪directory defined by LRMS
User-defined RTEs in /etc/arc/rte:
  ENV/RUNTIME/ALIEN-2.17 # RTE Description is Not Available
  ENV/PROXY # RTE Description is Not Available
Enabled RTEs:
  ENV/RUNTIME/ALIEN-2.17 -> /etc/arc/rte/ENV/RUNTIME/ALIEN-2.17
  ENV/PROXY -> /etc/arc/rte/ENV/PROXY
Default RTEs:
  ENV/LRMS-SCRATCH -> /usr/share/arc/rte/ENV/LRMS-SCRATCH

```

## View RTE content

Dumping the content of RTE that will be embedded to job script is possible with *cat* action:

```

[root ~]# arcctl rte cat ENV/LRMS-SCRATCH
SCRATCH_VAR="LOCALTMP"
# description: enables the usage of local to WN scratch directory defined by LRMS
# param:SCRATCH_VAR:string:WORKDIR:Variable name that holds the path to job-
  ↪specific WN scratch directory

SCRATCH_VAR="${SCRATCH_VAR:-WORKDIR}"

if [ "x$1" = "x0" ]; then
  RUNTIME_LOCAL_SCRATCH_DIR="\${${SCRATCH_VAR}}"
fi

```



## Disable and Undefault RTEs

*Enabled RTEs* can be disabled running:

```
arcctl rte disable ENV/PROXY
```

The similar operation for *default RTEs* is called undefault:

```
arcctl rte undefault ENV/LRMS-SCRATCH
```

---

**Note:** You can use the same *by-name*, *by-path* and *wildcard* techniques as for *enabling*

---

## 1.4.3 System-defined RunTime Environments shipped with ARC

### ENV/PROXY

Export delegated credentials (proxy certificate) to the job's session directory. Optionally copies CA certificates directory from ARC CE to session directory.

Sets the X509\_USER\_PROXY, X509\_USER\_CERT and X509\_CERT\_DIR to make files instantly available to client tools.

*Parameters:*

- COPY\_CACERT\_DIR = Yes/No - If set to Yes, CA certificate directory will be copied to the session directory along with proxy certificate.

### ENV/RTE

Copy RunTime Environment scripts to the job session directory for some workloads that require files itself instead of embedding the RTE to jobscript.

Designed to be used as *default* RTE.

Has no parameters.

### ENV/LRMS-SCRATCH

Many resource providers uses *scratchdir* to move files to local to worker node disk before running the job.

There is a useful case when local scratch is created dynamically by LRMS (e.g. in the job prologue) and than cleaned up automatically after the job completion. The scratch place should be indicated by some environmental variable that holds a path to such LRMS-defined scratch directory.

This RTE designed to be used as *default* RTE to enable this optional functionality.

*Parameters:*

- SCRATCH\_VAR = name - Variable name that holds the path to job-specific WN scratch directory (default is WORKDIR)

## 1.5 Deploying ARCHERY for Project/VO

This document shows the example configuration of ARCHERY to hold Computing Elements endpoints for some project (experiment, virtual organization, etc).

Examples assumes that you have BIND as a DNS service.

## 1.5.1 Entry point

An entry point to ARCHERY-based registry is a domain name. In this guide `example.org` used as an entry point. To submit job using ARCHERY run:

```
arcsub -g example.org myjob.xrsl
```

## 1.5.2 Transaction signature key to manage endpoints

It is possible to manage records manually, however to provide up-to-date information and keep is simple to manage endpoints it is advised to use Dynamic DNS updates with `archery-manage`.

### Generate key

Use the following command to generate the key:

```
dnssec-keygen -a HMAC-MD5 -b 256 -n USER archery
```

From the generated files you need a *secret* part to be included in both BIND and `archery-manage` configuration.

### Define key in BIND

Create the `/etc/named/archery.key` and put the secret key generated on the previous step here:

```
key archery_key {
    algorithm hmac-md5;
    secret "S0Me+SecRet+keYgener@tedwithdnssec==";
};
```

Include key definition into `named.conf` using the config line:

```
include "/etc/named/archery.key";
```

### Keyfile for archery-manage

Create a file (let it be named `archery-manage.key`) with the generated key in the following format:

```
archery_key:S0Me+SecRet+keYgener@tedwithdnssec==
```

## 1.5.3 Configure DNS zone to host endpoints

It is possible to put records directly into the `example.org` zone, however for manageability and security<sup>1</sup> reasons it is advised to have dedicated zone configured for ARCHERY records.

You can use any name for dedicated sub-zone, this example uses `index.example.org`. Addresses used in the following examples:

- Master DNS: `ns1.example.org` (192.0.2.100)
- Slave DNS: `ns2.example.org` (192.0.2.200)

---

<sup>1</sup> The content will be updated dynamically and it is better to eliminate the possibility of changing non-archery related records. Dedicated subzone is an easiest way.

## Define zone in DNS

Add zone definition to master `named.conf`:

```
zone "index.example.org." IN {
    type master;
    file "master/index.example.org.db";
    notify yes;
    also-notify {
        192.0.2.200; /* slave DNS IP address */
    };
    allow-transfer {
        192.0.2.200; /* slave DNS IP address */
    };
    allow-update {
        key archery_key;
    };
};
```

Please note `allow-update` directive that authorize dynamic DNS update queries signed by `archery-manage` key.

Slave DNS should be configured without any special options:

```
zone "dk.archery.nordugrid.org." IN {
    type slave;
    file "slave/index.example.org.db";
    masters {
        192.0.2.100;
    };
    allow-transfer {
        192.0.2.100;
    };
};
```

## Create zonefile with basic structure

Zonefile requires only basic SOA record and will be filled with data by `archery-manage`. The zonefile template (timers are subject to arrange depending on planned update frequency):

```
$ORIGIN example.org.
$TTL 3600
index IN SOA ns1.example.org. hostmaster.example.org. (
    2018082401 ; serial
    1200      ; refresh (20 minutes)
    180      ; retry (3 minutes)
    604800   ; expire (1 week)
    60      ; minimum (1 minute)
)
NS ns1.example.org.
NS ns2.example.org.
```

## Define records in parent zone

Define NS records<sup>2</sup> to refer defined subzone:

```
$ORIGIN example.org.
# dedicated ARCHERY zone
```

<sup>2</sup> If you plan to use different out-of-scope domain names in NS don't forget to add glue A records.

```
index NS ns1.example.org.
index NS ns2.example.org.
```

Define ARCHERY entry point TXT resource record:

```
$ORIGIN example.org.
# ARCHERY entry point
_archery TXT "u=index.example.org t=org.nordugrid.archery"
```

## 1.5.4 Prepare static list of Computing Elements

The initial list of CEs used by `archery-manage` to fetch the endpoints from LDAP GLUE2 and push it to DNS zone with incremental DDNS updates. CEs are specified line-by-line in the file:

```
ce01.example.org
ce02.example.org
arc6.example.org
```

To migrate from EGIIS `archery-manage` allows you to dump list of CEs automatically:

```
archery-manage -s egiis:ldap://old.egiis.example.org:2135/Mds-Vo-Name=ExampleVO,
→o=grid \
-o CEs > ces.list
```

## 1.5.5 Populate DNS with information

To populate DNS zone with endpoints information you should run `archery-manage` in the following way:

```
archery-manage -s file:ce.list --ddns-update \
--domain index.example.org --ddns-master-ip 192.0.2.100 \
--ddns-tsig-keyfile archery-manage.key
```

---

**Note:** Updates are performed over the network, so you can run `archery-manage` on any host. It SHOULD NOT be DNS server itself.

---

Consider to add filters to `archery-manage`:

### Filter: Port connectivity

Check network connectivity to endpoint TCP port and filter endpoints that do not pass this test:

```
archery-manage -s file:ce.list -f portscan ...
```

### Filter: Endpoint type

By default any endpoints available on CE are published. If you want to filter endpoints based on type there is a dedicated filter.

For example, if you want only EMI-ES ResourceInfo endpoints for EMI-ES only operation:

```
archery-manage -s file:ce.list -f type:org.ogf.glue.emies.resourceinfo ...
```

EMI-ES ResourceInfo and LDAP GLUE2 endpoints:

```
archery-manage -s file:ce.list -f type:org.ogf.glue.emies.resourceinfo,org.
↳ nordugrid.ldapglue2 ...
```

There is a special endpoint filter `resourceinfo` that is equivalent to EMI-ES ResourceInfo, LDAP GLUE2 and LDAP Nordugrid.

### Filter: VO

For project-based ARCHERY deployment it is also useful to filter endpoints based on VO access policy. Only endpoints that advertise specified VO support will be added<sup>3</sup>:

```
archery-manage -s file:ce.list -f vo:exampleVO ...
```

## 1.5.6 Checking the operation

### General DNS queries

Any DNS client tool can be used, for example:

```
[user ~]$ host -t TXT _archery.example.org
_archery.example.org descriptive text "u=index.example.org t=org.nordugrid.archery"

[root ~]$ host -t TXT _archery.index.example.org
_archery.index.example.org descriptive text "u=dns://ce01.example.org._archery.
↳ index.example.org t=org.nordugrid.archery"
_archery.index.example.org descriptive text "u=dns://ce02.example.org._archery.
↳ index.example.org t=org.nordugrid.archery"
<output omitted>

[root ~]$ host -t TXT ce01.example.org._archery.index.example.org
ce01.example.org._archery.index.example.org descriptive text "u=ldap://ce01.
↳ example.org:2135/o=glue t=org.nordugrid.ldapglue2"
ce01.example.org._archery.index.example.org descriptive text "u=https://ce01.
↳ example.org:60000/arex t=org.ogf.glue.emies.resourceinfo"
<output omitted>
```

### Submit job

Use `arcsb -g example.org -d DEBUG` to check the process of endpoints discovery from ARCHERY.

## 1.5.7 Setup regular updates to ARCHERY

To keep information about endpoints up to date setup a CRON job or Systemd Timer to run `archery-manage` regularly. In combination with at least *port filtering* this allows to eliminate stale endpoints and actualize information in registry.

## 1.5.8 Configure LDAP-monitor to use ARCHERY

Nordugrid LDAP monitor support fetching endpoints<sup>4</sup> from ARCHERY with the following configuration in `settings.inc`:

<sup>3</sup> Resource information (GLUE2PolicyRule in GLUE2 and `nordugrid-cluster-acl` in Nordugrid LDAP) will be used as a source of supported VOs

<sup>4</sup> Nordugrid monitor only works with `org.nordugrid.ldapng` data (and LDAP GLUE2 is experimental with known issues), so Nordugrid LDAP schema publishing is mandatory to use monitor.

```
$archery_list = array (  
  array (  
    "endpoint" => "example.org",  
  )  
);
```

**See also:**

**ARCHERY Architecture Overview** ARCHERY Architecture Overview, including concept and DNS records specification.

**ARCHERY NorduGrid Top-Level Deployment** ARCHERY Deployment for Top-Level Nordugrid Registry, including example configuration.

## DOCUMENTATION FOR DEVELOPERS

### 2.1 Packages and Services Naming changes in ARC6

Summary of the packaging and services naming:

Block name	ARC6 pakage name (ARC5 name)	ARC6 Service Name (ARC5 name)
[arex]	nordugrid-arc-arex	arc-arex (a-rex)
[arex/ws/candypond]	part of arex (nordugrid-arc-candypond)	part of arex (arc-candypond) <sup>2</sup>
[gridftpd]	nordugrid-arc-gridftpd	arc-gridftpd (gridftpd)
[infosys/ldap]	nordugrid-arc-infosys-ldap (nordugrid-arc-aris) <sup>3</sup>	arc-infosys-ldap (nordugrid-arc-aris)
[datadelivery-service]	nordugrid-arc-datadelivery-service	arc-datadelivery-service
[acix-scanner]	nordugrid-arc-acix-scanner (nordugrid-arc-acix-cache) <sup>4</sup>	arc-acix-scanner (acix-cache)
[acix-index]	nordugrid-arc-acix-index	arc-acix-index (acix-index)
[nordugridmap]	nordugrid-arc-nordugridmap (nordugrid-arc-gridmap-utils)	N/A
N/A	nordugrid-arc-test-utils (nordugrid-arc-misc-utils) <sup>5</sup>	N/A

### 2.2 General arc.conf python configuration parser

#### 2.2.1 Parsing configuration

##### Initial configuration parsing

The latest version of `arcconfig-parser` designed to operate with defaults file that holds default values for all possible parameters.

At the time of initial parsing the following chain is executed:

<sup>2</sup> Libraries from dedicated package are moved to A-REX package

<sup>3</sup> Package define LDAP/BDII/Glue-Schema dependencies and contains wrappers to start all this LDAP world. Infoproviders are in the A-REX package.

<sup>1</sup> Stay with an old name until the new agreed naming happened. Some suggestions:

- candyjet - Cache and Deliver You Job Extensive Transfers
- arc-jet - ARC Justified Exceptional Tranfer service ('Arc Jet' is also some NASA complex)
- hexaton - Helper for EXtensive ARC Transfers OffloadiNg

<sup>4</sup> Python packages should be renamed also: `acix.scanned` instead of `acix.cacheserver`.

<sup>5</sup> No longer relevan `saml_assertion_init` tool had been removed

- All blocks and options are parsed from `arc.conf`
- For all blocks defined in `arc.conf` missing options are added from defaults file
- Special constructs in values are substituted (see *Special constructs can be used as values*)

Optional by design parameters that does not have default value (specified with `not set value`) are not included to the parsed configuration.

### Runtime configuration

Configuration that includes both `arc.conf` and defaults config called *runtime configuration*.

In some cases it is useful to save and load runtime configuration:

- To supply C++ services (`a-rex`, `gridftpd`) with configuration that includes defaults from common place
- For repetitive operations on config to eliminate full-chain processing of each invocation

To save runtime configuration to the default location (`/var/run/arc/`):

```
arcconfig-parser --save
```

To save runtime configuration to specified location:

```
arcconfig-parser --save -r /var/run/arc/arex.arc.conf
```

To load runtime configuration instead of full-chain processing and e.g. get the value of `x509_host_key` in `[common]` block:

```
arcconfig-parser --load -b common -o x509_host_key
```

### Special constructs can be used as values

Defaults includes references to another config parts to be consistent with the implied `arc.conf` structure.

The most obvious example is if `x509_host_key` not found in e.g. `[arex/jura]` block it should be taken from `[common]`.

Config parser is following this logic (especially in respect to defaults) and use special constructs to accomplish this behaviour.

### Command substitutions

Configuration option values can contain the construct `$EXEC{<command>}` that substituted to the stdout of `<command>`.

For example:

```
hostname=$EXEC{hostname -f}
```

### Option values substitutions

The construct `$VAR{[block]option}` can be used to substitute the values of another option value.

If option is in the same block as referencing option block name can be omitted - `$VAR{option}`.

For example:

```
x509_host_key=$VAR{[common]x509_host_key}
bdii_update_cmd=$VAR{bdii_location}/sbin/bdii-update
```



## Evaluation of simple code

For limited number of cases `arc.conf` default values relies on arithmetic operations. For this purpose the `$EVAL{string}` special construct had been introduced.

For example:

```
bdi_read_timeout=$EVAL{$VAR{bdi_provider_timeout} + $VAR{[arex]infoproviders_
→timelimit} + $VAR{[arex]wakeupperiod}}
```

## Getting the configuration values

If `--option` argument is passed to `arconfig-parser` parser returns the value of the specified option to stdout.

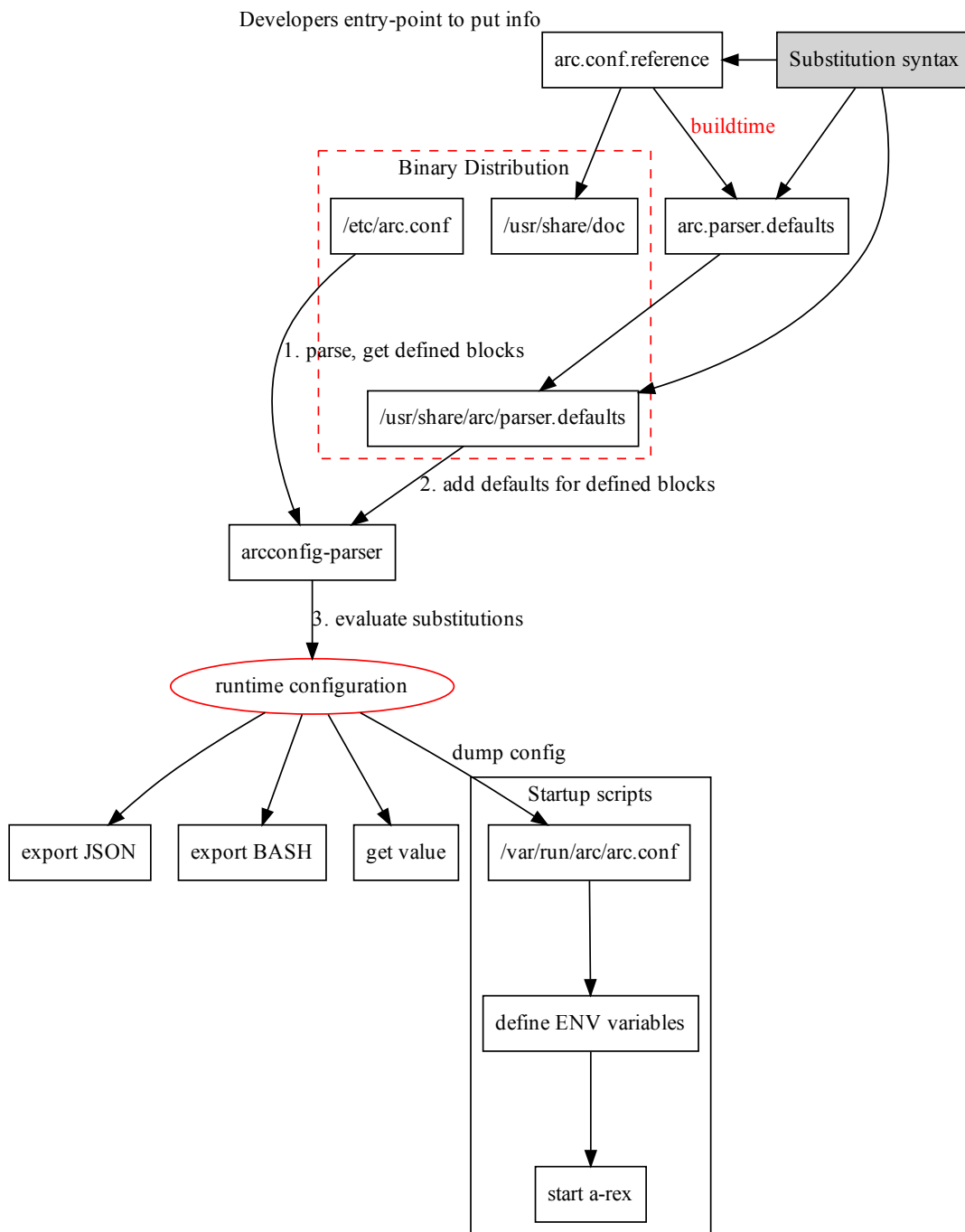
**Without `--option` `arconfig-parser` can be used to operate with configuration blocks:**

- check blocks existence (exit code used to indicate the status of the check)
- return the list of subblocks

**With the `--export` option `arconfig-parser` allows to export config in the following formats:**

- `json` - returns entire configuration or subset of blocks as-is in JSON to stdout
- `bash` - for `[common]` block or specified configuration subset returns `CONFIG_option_name=value` pairs to stdout. Block names ARE NOT included in the exports and option values precedence will be used in the order of passed blocks. If automatic subblocks expansion used with bash export, for every block in sequence - it's subblocks are processed first (in `arc.conf` defined order). It is possible to filter the options that will be exported with additional `--filter` option that can be specified several times.

### Common configuration parsing sequence



### 2.2.2 Examples

Get value of option in block:

```
# arconfig-parser --block infosys --option providerlog
/var/log/arc/infoprovider.log
```

Get value of option in blocks in order they are specified (e.g. if not found in [gridftpd] look in the [common] block<sup>1</sup>):

```
# arconfig-parser --block gridftpd --block common --option x509_user_key
/etc/grid-security/hostkey.pem
```

Export entire configuration to JSON<sup>2</sup>:

```
# arconfig-parser --export json
```

Export [infosys] block options to JSON (for Perl):

```
# arconfig-parser --block infosys --export json
{"infosys": {"loglevel": "5"}, ...
```

Export [infosys] block and all their subblocks options to JSON:

```
# arconfig-parser --block infosys --subblocks --export json
{"infosys/glue2/ldap": {"showactivities": "no"}, ...
```

Export for BASH (compatible with config representation in shell-based LRMS backends):

```
# arconfig-parser --block infosys --block arex --block common --export bash
CONFIG_controlldir="/var/spool/arc/jobstatus"
CONFIG_defaultttl="1210000"
CONFIG_delegationdb="sqlite"
CONFIG_hostname="ce01.example.org"
CONFIG_maaxrerun="5"
CONFIG_maxjobs="10000 -1"
CONFIG_runtimedir="/home/grid/arc/runtime"
CONFIG_sessiondir="__array__" # <= NEW define for multivalued values that indicate
↳indexed vars
CONFIG_sessiondir_0="/mnt/scratch/grid/arc/session"
CONFIG_sessiondir_1="/home/grid/arc/session drain"
...
```

Export for BASH with exported options filtering:

```
# arconfig-parser -b common -f hostname -f x509_cert_dir -e bash
CONFIG_hostname="ce01.example.org"
CONFIG_x509_cert_dir="/etc/grid-security/certificates"
```

Using BASH export:

```
# eval "$( arconfig-parser --block infosys --block arex --block common --export
↳bash )"
# echo "$CONFIG_gridmap"
```

Check block(s) exists ([common/perflog] is not exists in the example):

```
# arconfig-parser --block common/perflog --block arex
# echo $?
1
```

List block subblocks:

```
# arconfig-parser --block infosys --subblocks
infosys
infosys/ldap
infosys/nordugrid
```

<sup>1</sup> Block dependencies are now implied by defaults file, so for most cases it is enough to specify only block in question

<sup>2</sup> *HINT*: use `arconfig-parser --export json | jq .` to view highlighted JSON structure in shell

```
infosys/glue2
infosys/glue2/ldap
infosys/glue1
```

Using parser as Python module:

```
from arc.utils import config

# initial parsing with defaults
config.parse_arc_conf('/tmp/arc.conf', '/tmp/defaults.conf')

# get parsed dictionary and list of blocks in the arc.conf order
>>> confdict = config.get_config_dict()
>>> confblocks = config.get_config_blocks()

# get list of all [queue] subblocks sorted by name
>>> sb = config.get_subblocks(['queue'], is_sorted=True)
>>> sb
['queue:grid', 'queue:grid_rt']

# get value of 'x509_host_key' from [arex] block and than from [common] if not_
↳found in [arex]
>>> a = config.get_value('x509_host_key', ['arex', 'common'])
>>> a
'/etc/grid-security/hostkey.pem'

# get value of 'allowunknown' option from [gridftpd] block
>>> b = config.get_value('allowunknown', 'gridftpd')
>>> b
'yes'

# get value of 'allowunknown' option from [gridftpd] block (always return list)
>>> c = config.get_value('allowunknown', 'gridftpd', force_list=True)
>>> c
['yes']

# get value of 'allowunknown' option from [gridftpd] block (return boolean value)
>>> d = config.get_value('allowunknown', 'gridftpd', bool_yesno=True)
>>> d
True
```

## 2.3 LRMS shell-backends overview for developers

### 2.3.1 CONFIG variables used in LRMS shell-backend:

lrms\_common.sh:

```
$CONFIG_runtimedir      [arex]
$CONFIG_shared_scratch  [arex]
$CONFIG_shared_filesystem [arex]
$CONFIG_scratchdir      [arex]
$CONFIG_gnu_time        [lrms]
$CONFIG_nodename        [lrms]
$CONFIG_enable_perflog_reporting [common]
$CONFIG_perflogdir      [common]
```

submit\_common.sh:

```
$CONFIG_defaultmemory      [queue] [lrms]
$CONFIG_hostname           [common]
$CONFIG_controldir         [arex]
```

**lrms=boinc:**

```
$CONFIG_boinc_app_id      # not in reference
$CONFIG_boinc_db_host     [lrms]
$CONFIG_boinc_db_port     [lrms]
$CONFIG_boinc_db_user     [lrms]
$CONFIG_boinc_db_pass     [lrms]
$CONFIG_boinc_db_name     [lrms]
```

**lrms=condor<sup>3</sup>:**

```
# $CONFIG_enable_perflog_reporting [common] not in reference
# $CONFIG_perflogdir               [common] not in reference
# $CONFIG_controldir               [arex] (for perflog)

$CONFIG_condor_requirements [queue] [lrms]
$CONFIG_condor_rank         [lrms]
# $CONFIG_shared_filesystem [arex]
$CONFIG_condor_bin_path     [lrms]
$CONFIG_condor_config       [lrms]
```

**lrms=fork:**

```
no variables
```

**lrms=ll:**

```
# $CONFIG_enable_perflog_reporting [common] not in reference
# $CONFIG_perflogdir               [common] not in reference
# $CONFIG_controldir               [arex] (for perflog)

$CONFIG_ll_bin_path         [lrms]
$CONFIG_ll_consumable_resources [lrms]
$CONFIG_ll_parallel_single_jobs *not in reference
# $CONFIG_scratchdir         [arex]
```

**lrms=lsf:**

```
# $CONFIG_enable_perflog_reporting [common] not in reference
# $CONFIG_perflogdir               [common] not in reference
# $CONFIG_controldir               [arex] (for perflog)

$CONFIG_lsf_architecture    [lrms]
$CONFIG_lsf_bin_path       [lrms]
```

**lrms=pbs:**

```
# $CONFIG_enable_perflog_reporting [common] not in reference
# $CONFIG_perflogdir               [common] not in reference
# $CONFIG_controldir               [arex] (for perflog)

$CONFIG_pbs_queue_node      [queue]
$CONFIG_pbs_bin_path        [lrms]
$CONFIG_nodememory          [queue] ([infosys/cluster] parser_
↳substitution fallback only)
$CONFIG_pbs_log_path        [lrms]
# $CONFIG_shared_filesystem [arex]
```

<sup>3</sup> Here and following # prefix is for options and are used in \*\_common scripts and not unique to particular backend

lrms=sge:

```
# $CONFIG_enable_perflog_reporting    [common] not in reference
# $CONFIG_perflogdir                  [common] not in reference
# $CONFIG_controldir                   [arex] (for perflog)

$CONFIG_sge_root                       [lrms]
$CONFIG_sge_cell                       [lrms]
$CONFIG_sge_qmaster_port                [lrms]
$CONFIG_sge_execd_port                  [lrms]
$CONFIG_sge_bin_path                    [lrms]
$CONFIG_sge_jobopts                     [queue] [lrms]
# $CONFIG_scratchdir                    [arex]
```

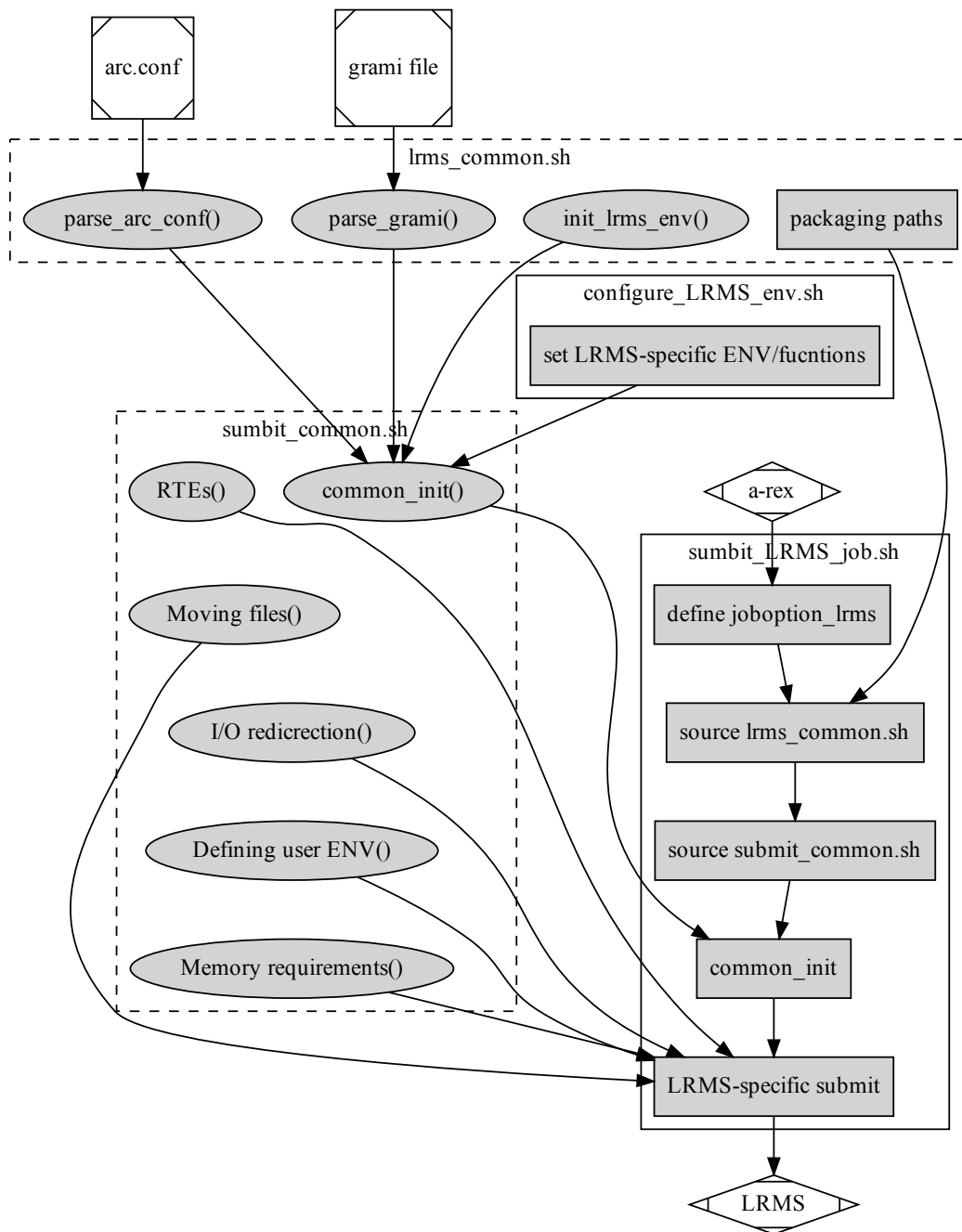
lrms=slurm:

```
# $CONFIG_enable_perflog_reporting    [common] not in reference
# $CONFIG_perflogdir                  [common] not in reference
# $CONFIG_controldir                   [arex] (for perflog)

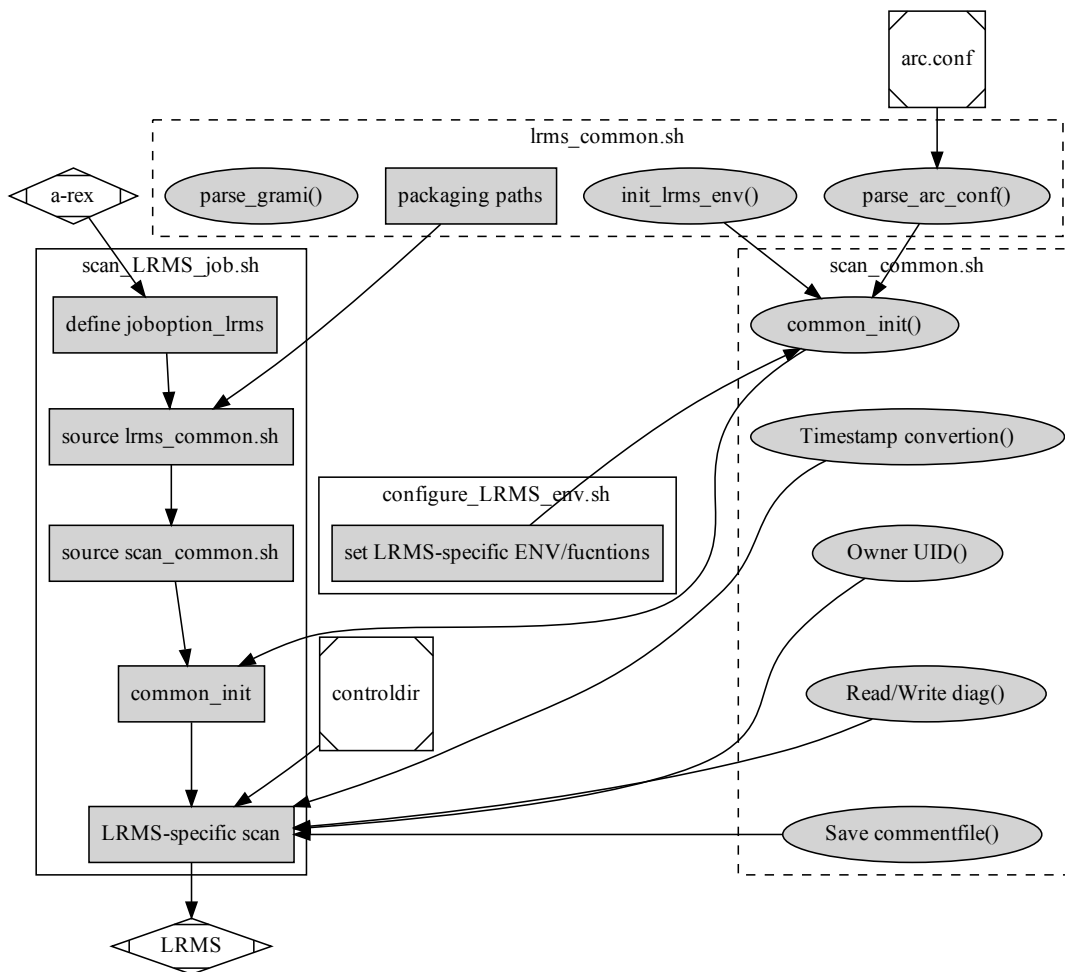
$CONFIG_slurm_wakeupperiod              [lrms]
$CONFIG_slurm_use_sacct                 [lrms]
$CONFIG_slurm_bin_path                  [lrms]
# $CONFIG_shared_filesystem             [arex]
```

## 2.3.2 Call graph

Submitting jobs

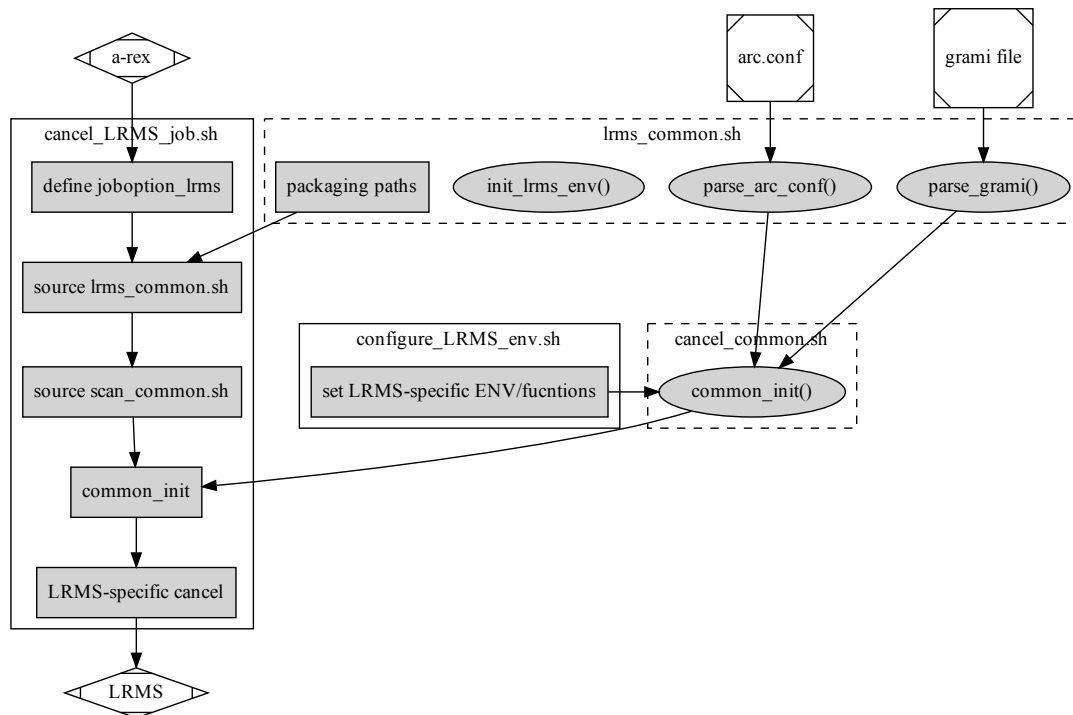


## Scanning jobs





## Canceling jobs



### 2.3.3 Changes in ARC6 memory limits processing:

Current logic of memory limits processing:

- `nodememory` - advertise memory for matchmaking: max memory on the nodes (in `[infosys/cluster]` block or per-queue)
- `defaultmemory` - enforce during submission if no memory limit specified in the job description (in `[lrms]` block or per-queue)

The ARC6 logic is *no enforcement = no limit*<sup>1</sup>

#### Backends behaviour with no memory enforcement limit:

- boinc - set to hardcoded 2GB
- condor - no enforcement
- form - no memory handling at all
- ll - no enforcement
- lsf - no enforcement
- pbs - no enforcement<sup>2</sup>
- sge - no enforcement
- slurm - no enforcement

<sup>1</sup> ARC5 logic was *no enforcement = max node memory* or 1GB if `nodememory` is not published (and not used for matchmaking)

<sup>2</sup> `exclusivenode` is memory-based and `nodememory` value is used in this case



## ARC UTILITIES COMMAND LINE REFERENCE

### 3.1 ARC Control Tool

Nordugrid ARC Computing Element Control Tool

```
usage: arcctl [-h] [-c CONFIG] [-d {CRITICAL,ERROR,WARNING,INFO,DEBUG}]  
           COMPONENT ...
```

#### 3.1.1 Named Arguments

- |                     |   |
|---------------------|---|
| <b>-c, --config</b> | config file location (default is “/etc/arc.conf”)<br>Default: “/etc/arc.conf”   |
| <b>-d, --debug</b>  | Possible choices: CRITICAL, ERROR, WARNING, INFO, DEBUG<br>verbosity level (default is “WARNING”)<br>Default: “WARNING” |

#### 3.1.2 ARC CE Components

- |                  |   |
|------------------|---|
| <b>COMPONENT</b> | Possible choices: rte, job, service, deploy, accounting, config, test-ca<br>DESCRIPTION |
|------------------|---|

#### 3.1.3 Sub-commands:

##### rte

RunTime Environments

```
arcctl rte [-h] ACTION ...
```

##### RunTime Environments Actions

- |               |  |
|---------------|--|
| <b>ACTION</b> | Possible choices: enable, disable, list, default, undefault, cat, params-get,<br>params-set<br>DESCRIPTION |
|---------------|--|

### Sub-commands:

#### enable

Enable RTE to be used by A-REX

```
arcctl rte enable [-h] [-f] [-d] rte [rte ...]
```

#### Positional Arguments

<b>rte</b>	RTE name
------------	----------

#### Named Arguments

<b>-f, --force</b>	Force RTE enabling
--------------------	--------------------

Default: False

<b>-d, --dummy</b>	Enable dummy RTE that do nothing but published in the infosys
--------------------	---

Default: False

#### disable

Disable RTE to be used by A-REX

```
arcctl rte disable [-h] rte [rte ...]
```

#### Positional Arguments

<b>rte</b>	RTE name
------------	----------

#### list

List RunTime Environments

```
arcctl rte list [-h] [-l] [-e | -d | -a | -s | -u | -n]
```

#### Named Arguments

<b>-l, --long</b>	Detailed listing of RTEs
-------------------	--------------------------

Default: False

<b>-e, --enabled</b>	List enabled RTEs
----------------------	-------------------

Default: False

<b>-d, --default</b>	List default RTEs
----------------------	-------------------

Default: False

<b>-a, --available</b>	List available RTEs
------------------------	---------------------

Default: False

<b>-s, --system</b>	List available system RTEs Default: False
<b>-u, --user</b>	List available user-defined RTEs Default: False
<b>-n, --dummy</b>	List dummy enabled RTEs Default: False

### default

Transparently use RTE for every A-REX job

```
arcctl rte default [-h] [-f] rte [rte ...]
```

### Positional Arguments

<b>rte</b>	RTE name
------------	----------

### Named Arguments

<b>-f, --force</b>	Force RTE enabling Default: False
--------------------	--------------------------------------

### undefault

Remove RTE from transparent A-REX usage

```
arcctl rte undefault [-h] rte [rte ...]
```

### Positional Arguments

<b>rte</b>	RTE name
------------	----------

### cat

Print the content of RTE file

```
arcctl rte cat [-h] rte
```

### Positional Arguments

<b>rte</b>	RTE name
------------	----------

### params-get

List configurable RTE parameters

```
arcctl rte params-get [-h] [-l] rte
```

#### Positional Arguments

<b>rte</b>	RTE name
------------	----------

#### Named Arguments

<b>-l, --long</b>	Detailed listing of parameters Default: False
-------------------	--

### params-set

Set configurable RTE parameters

```
arcctl rte params-set [-h] rte parameter value
```

#### Positional Arguments

<b>rte</b>	RTE name
<b>parameter</b>	RTE parameter to configure
<b>value</b>	RTE parameter value to set

### job

A-REX Jobs

```
arcctl job [-h] [-t CACHETTL] ACTION ...
```

#### Named Arguments

<b>-t, --cachettl</b>	GM-Jobs output caching validity in seconds (default is 30) Default: 30
-----------------------	---

#### Jobs Control Actions

<b>ACTION</b>	Possible choices: list, log, info, attr, kill, killall, clean, cleanall, stats DESCRIPTION
---------------	---

## Sub-commands:

### list

List available A-REX jobs

```
arcctl job list [-h] [-l]
                [-s {ACCEPTED,PREPARING,SUBMIT,INLRMS,FINISHING,FINISHED,DELETED,
↵CANCELING}]
                [-o OWNER]
```

## Named Arguments

- l, --long**           Detailed listing of jobs  
Default: False
- s, --state**       Possible choices: ACCEPTED, PREPARING, SUBMIT, INLRMS, FINISH-  
ING, FINISHED, DELETED, CANCELING  
Filter jobs by state
- o, --owner**       Filter jobs by owner

### log

Display job log

```
arcctl job log [-h] [-l] [-s] jobid
```

## Positional Arguments

- jobid**               Job ID

## Named Arguments

- l, --lrms**       Include LRMS job submission script into the output  
Default: False
- s, --service**   Show ARC CE logs containing the jobID instead of job log  
Default: False

### info

Show job main info

```
arcctl job info [-h] jobid
```

## Positional Arguments

- jobid**               Job ID

## attr

Get

```
arcctl job attr [-h] jobid [attr]
```

### Positional Arguments

<b>jobid</b>	Job ID
<b>attr</b>	Attribute name

## kill

Cancel job

```
arcctl job kill [-h] jobid [jobid ...]
```

### Positional Arguments

<b>jobid</b>	Job ID
--------------	--------

## killall

Cancel all jobs

```
arcctl job killall [-h]
                    [-s {ACCEPTED, PREPARING, SUBMIT, INLRMS, FINISHING, FINISHED,
↪DELETED, CANCELING}]
                    [-o OWNER]
```

### Named Arguments

<b>-s, --state</b>	Possible choices: ACCEPTED, PREPARING, SUBMIT, INLRMS, FINISHING, FINISHED, DELETED, CANCELING
	Filter jobs by state
<b>-o, --owner</b>	Filter jobs by owner

## clean

Clean job

```
arcctl job clean [-h] jobid [jobid ...]
```

### Positional Arguments

<b>jobid</b>	Job ID
--------------	--------



## cleanall

Clean all jobs

```
arcctl job cleanall [-h]
                    [-s {ACCEPTED,PREPARING,SUBMIT,INLRMS,FINISHING,FINISHED,
->DELETED,CANCELING}]
                    [-o OWNER]
```

### Named Arguments

- s, --state** Possible choices: ACCEPTED, PREPARING, SUBMIT, INLRMS, FINISHING, FINISHED, DELETED, CANCELING  
Filter jobs by state
- o, --owner** Filter jobs by owner

## stats

Show jobs statistics

```
arcctl job stats [-h] [-S] [-t] [-d] [-l]
```

### Named Arguments

- S, --no-states** Do not show per-state job stats  
Default: False
- t, --total** Show server total stats  
Default: False
- d, --data-staging** Show server datastaging stats  
Default: False
- l, --long** Detailed output of stats  
Default: False

## service

ARC CE services control

```
arcctl service [-h] ACTION ...
```

### Services Actions

- ACTION** Possible choices: enable, disable, start, stop, list  
DESCRIPTION

### Sub-commands:

#### enable

Enable ARC CE services

```
arcctl service enable [-h] [--now] (-a | -s SERVICE)
```

#### Named Arguments

- |                            |  |
|----------------------------|--|
| <b>--now</b>               | Start the services just after enable<br>Default: False               |
| <b>-a, --as-configured</b> | Use information from arc.conf to get services list<br>Default: False |
| <b>-s, --service</b>       | Service name   |

#### disable

Disable ARC CE services

```
arcctl service disable [-h] [--now] (-a | -s SERVICE)
```

#### Named Arguments

- |                            |  |
|----------------------------|--|
| <b>--now</b>               | Stop the services just after disable<br>Default: False               |
| <b>-a, --as-configured</b> | Use information from arc.conf to get services list<br>Default: False |
| <b>-s, --service</b>       | Service name   |

#### start

Start ARC CE services

```
arcctl service start [-h] (-a | -s SERVICE)
```

#### Named Arguments

- |                            |  |
|----------------------------|--|
| <b>-a, --as-configured</b> | Use information from arc.conf to get services list<br>Default: False |
| <b>-s, --service</b>       | Service name   |

## stop

Start ARC CE services

```
arcctl service stop [-h] (-a | -s SERVICE)
```

### Named Arguments

- a, --as-configured** Use information from arc.conf to get services list  
Default: False
- s, --service** Service name

## list

List ARC CE services and their states

```
arcctl service list [-h] [-i | -e | -a]
```

### Named Arguments

- i, --installed** Show only installed services  
Default: False
- e, --enabled** Show only enabled services  
Default: False
- a, --active** Show only running services  
Default: False

## deploy

Third party components deployment

```
arcctl deploy [-h] ACTION ...
```

### Deployment Actions

- |               |  |
|---------------|--|
| <b>ACTION</b> | Possible choices: voms-lsc, igtf-ca, iptables-config |
|               | DESCRIPTION  |

### Sub-commands:

#### voms-lsc

Deploy VOMS list-of-certificates files

```
arcctl deploy voms-lsc [-h] (-v VOMS | -e) [-o] vo
```

### Positional Arguments

**vo** VO Name

### Named Arguments

**-v, --voms** VOMS-Admin URL

**-e, --egi-vo** Fetch information from EGI VOs database  
Default: False

**-o, --openssl** Use external OpenSSL command instead of python SSL  
Default: False

### igtf-ca

Deploy IGTF CA certificates

```
arcctl deploy igtf-ca [-h] [-i {igtf,egi-trustanchors,nordugrid}]  
                        {classic,iota,mics,slcs} [{classic,iota,mics,slcs} ...]
```

### Positional Arguments

**bundle** Possible choices: classic, iota, mics, slcs  
IGTF CA bundle name

### Named Arguments

**-i, --installrepo** Possible choices: igtf, egi-trustanchors, nordugrid  
Add specified repository that contains IGTF CA certificates

### iptables-config

Generate iptables config to allow ARC CE configured services

```
arcctl deploy iptables-config [-h] [--any-state] [--multiport]
```

### Named Arguments

**--any-state** Do not add '-state NEW' to filter configuration  
Default: False

**--multiport** Use one-line multiport filter instead of per-service entries  
Default: False

## accounting

Accounting records management

```
arcctl accounting [-h] ACTION ...
```

### Accounting Actions

ACTION	Possible choices: republish, logs, stats, apel-brokers
	DESCRIPTION

### Sub-commands:

#### republish

Republish archived usage records

```
arcctl accounting republish [-h] -b START_FROM -e END_TILL
                             (-a APEL_URL | -s SGAS_URL)
                             [-t {/queue/global.accounting.cpu.central, /queue/
↪global.accounting.test.cpu.central}]
```

### Named Arguments

<b>-b, --start-from</b>	Limit the start time of the records (YYYY-MM-DD [HH:mm[:ss]])
<b>-e, --end-till</b>	Limit the end time of the records (YYYY-MM-DD [HH:mm[:ss]])
<b>-a, --apel-url</b>	Specify APEL server URL (e.g. <a href="https://mq.cro-ngi.hr:6163">https://mq.cro-ngi.hr:6163</a> )
<b>-s, --sgas-url</b>	Specify APEL server URL (e.g. <a href="https://grid.uio.no:8001/logger">https://grid.uio.no:8001/logger</a> )
<b>-t, --apel-topic</b>	Possible choices: /queue/global.accounting.cpu.central, /queue/global.accounting.test.cpu.central Redefine APEL topic (default is “/queue/global.accounting.cpu.central”) Default: “/queue/global.accounting.cpu.central”

#### logs

Show accounting logs

```
arcctl accounting logs [-h] [-s]
```

### Named Arguments

<b>-s, --ssm</b>	Show SSM logs instead of Jura logs Default: False
------------------	--

## stats

Show archived records statistics

```
arcctl accounting stats [-h] -t {apel,sgas} [-b START_FROM] [-e END_TILL]
                        [--filter-vo FILTER_VO] [--filter-user FILTER_USER]
                        [-j | -w | -c | -v | -u]
```

### Named Arguments

<b>-t, --type</b>	Possible choices: apel, sgas Accounting system type
<b>-b, --start-from</b>	Limit the start time of the records (YYYY-MM-DD [HH:mm[:ss]])
<b>-e, --end-till</b>	Limit the end time of the records (YYYY-MM-DD [HH:mm[:ss]])
<b>--filter-vo</b>	Count only the jobs owned by this VO(s)
<b>--filter-user</b>	Count only the jobs owned by this user(s)
<b>-j, --jobs</b>	Show number of jobs Default: False
<b>-w, --walltime</b>	Show total WallTime Default: False
<b>-c, --cputime</b>	Show total CPUTime Default: False
<b>-v, --vos</b>	Show VO that owns jobs Default: False
<b>-u, --users</b>	Show users that owns jobs Default: False

## apel-brokers

Fetch available APEL brokers from GLUE2 Top-BDII

```
arcctl accounting apel-brokers [-h] [-t TOP_BDII] [-s]
```

### Named Arguments

<b>-t, --top-bdii</b>	Top-BDII LDAP URI (default is "ldap://lcg-bdii.cern.ch:2170") Default: "ldap://lcg-bdii.cern.ch:2170"
<b>-s, --ssl</b>	Query for SSL brokers Default: False

## config

ARC CE configuration control

```
arcctl config [-h] ACTION ...
```

### Config Actions

<b>ACTION</b>	Possible choices: dump, get, describe, set, brief
	DESCRIPTION

### Sub-commands:

#### dump

Dump ARC CE running configuration

```
arcctl config dump [-h]
```

#### get

Print configuration option value

```
arcctl config get [-h] block option
```

### Positional Arguments

<b>block</b>	Name of configuration block (without square brackets)
<b>option</b>	Configuration option name

#### describe

Describe configuration option

```
arcctl config describe [-h] [-r REFERENCE] block option
```

### Positional Arguments

<b>block</b>	Name of configuration block (without square brackets)
<b>option</b>	Configuration option name

### Named Arguments

<b>-r, --reference</b>	Redefine <code>arc.conf.reference</code> location (default is <code>"/usr/share/arc/examples/arc.conf.reference"</code> ) Default: <code>"/usr/share/arc/examples/arc.conf.reference"</code>
------------------------	---

## set

Change configuration option value

```
arcctl config set [-h] [-o] [-r REFERENCE] [--dry-run]
                block option value [value ...]
```

### Positional Arguments

<b>block</b>	Name of configuration block (without square brackets)
<b>option</b>	Configuration option name
<b>value</b>	Configuration option value

### Named Arguments

<b>-o, --override</b>	For multivalued options override config values (default is add another one) Default: False
<b>-r, --reference</b>	Redefine arc.conf.reference location (default is “/usr/share/arc/examples/arc.conf.reference”) Default: “/usr/share/arc/examples/arc.conf.reference”
<b>--dry-run</b>	Write the modified config to stdout instead of changing the file Default: False

## brief

Print configuration brief points

```
arcctl config brief [-h] [-t {storage,logs}]
```

### Named Arguments

<b>-t, --type</b>	Possible choices: storage, logs Show brief only for provided options type
-------------------	--

## test-ca

ARC Test CA control

```
arcctl test-ca [-h] ACTION ...
```

### Test CA Actions

<b>ACTION</b>	Possible choices: init, hostcert, usercert DESCRIPTION
---------------	---



**Sub-commands:****init**

Generate self-signed TestCA files

```
arcctl test-ca init [-h]
                    [-d {md2,md4,md5,mdc2,sha1,sha224,sha256,sha384,sha512}]
                    [-v VALIDITY] [-f]
```

**Named Arguments**

- d, --digest**           Possible choices: md2, md4, md5, mdc2, sha1, sha224, sha256, sha384, sha512  
Digest to use (default is “sha256”)  
Default: “sha256”
- v, --validity**        Validity of certificate in days (default is 90)  
Default: 90
- f, --force**            Overwrite files if exists  
Default: False

**hostcert**

Generate and sign testing host certificate

```
arcctl test-ca hostcert [-h]
                        [-d {md2,md4,md5,mdc2,sha1,sha224,sha256,sha384,sha512}]
                        [-v VALIDITY] [-n HOSTNAME] [-f]
```

**Named Arguments**

- d, --digest**           Possible choices: md2, md4, md5, mdc2, sha1, sha224, sha256, sha384, sha512  
Digest to use (default is “sha256”)  
Default: “sha256”
- v, --validity**        Validity of certificate in days (default is 30)  
Default: 30
- n, --hostname**        Generate certificate for specified hostname instead of this host
- f, --force**            Overwrite files if exists  
Default: False

**usercert**

Generate and sign testing user certificate

```
arcctl test-ca usercert [-h]
                        [-d {md2,md4,md5,mdc2,sha1,sha224,sha256,sha384,sha512}]
                        [-v VALIDITY] [-n USERNAME] [-t]
```

### Named Arguments

- d, --digest**           Possible choices: md2, md4, md5, mdc2, sha1, sha224, sha256, sha384, sha512  
Digest to use (default is “sha256”)  
Default: “sha256”
- v, --validity**        Validity of certificate in days (default is 30)  
Default: 30
- n, --username**        Use specified username instead of automatically generated
- t, --export-tar**      Export tar archive to use from another host  
Default: False

## 3.2 ARC Configuration Parser

Nordugrid ARC configuration parser

```
usage: arconfig-parser [-h] [--debug {CRITICAL,ERROR,WARNING,INFO,DEBUG}]
                      [--load] [--save] [-r RUNCONFIG] [-c CONFIG]
                      [-d DEFAULTS] [-b BLOCK] [-o OPTION] [-s]
                      [-e {bash,json}] [-f EXPORT_FILTER]
```

### 3.2.1 Named Arguments

- debug**            Possible choices: CRITICAL, ERROR, WARNING, INFO, DEBUG  
verbosity level (default is “WARNING”)  
Default: “WARNING”

### 3.2.2 Runtime configuration

Work with runtime configuration that includes default values

- load**            load ARC runtime configuration  
Default: False
- save**            save ARC runtime configuration  
Default: False
- r, --runconfig**   runtime config file location (default is “/var/run/arc/arc.running.conf”)  
Default: “/var/run/arc/arc.running.conf”

### 3.2.3 Configuration files

Initial ARC configuration files

- c, --config**            config file location (default is “/etc/arc.conf”)  
Default: “/etc/arc.conf”
- d, --defaults**        defaults file location (default is “/usr/share/arc/arc.parser.defaults”)  
Default: “/usr/share/arc/arc.parser.defaults”

### 3.2.4 Getting values

Get blocks and configuration option values

- b, --block**            block name (can be specified several times)
- o, --option**         option name
- s, --subblocks**      match subblocks against supplied block name(s)  
Default: False
- e, --export**         Possible choices: bash, json  
export configuration to the defined format
- f, --export-filter**   limit bash export to specified options only