NORDUGRID-XXXXXXX-NN
30/6/2015

ARC Web Services Quick Usage Guide

Contents

1 ARC Middleware

The Advanced Resource Connector (ARC) middleware [**?**], introduced by NorduGrid [**?**], is an open source software solution enabling production quality computational and data grids. Since the first release (May 2002) the middleware has been deployed and been used in production environments. Emphasis is put on scalability, stability, reliability and performance of the middleware. A growing number of grid projects, like NDGF [**?**], Swegrid [**?**] or Swiss National Grid Association [**?**] have chosen ARC as their middleware.

This document provides a quick usage guide to the new Web Service (WS) components of ARC, often referred to as ARC1 components. WS design of ARC is based on a service container – the *Hosting Environment Daemon (HED)*, and various capabilities of the grid are implemented as distinct Web Services that reside in HED.

At the time of writing, this comprises the following components:

- an OGSA BES [**?**] compliant execution service – *ARC Resource-coupled EXecution service (A-REX)* [**?**]
- an *Echo* service (for testing purposes)

2 Dependencies

The core part of middleware is written in C/C++. To build the software from source or installing a precompiled binary requires the prior installation of multiple external packages. Client and server packages differ in their dependencies to some degree. The following lists the explicit requirements:

- Mandatory (on client as well as server side):
  - gnu make, autotools (autoconf $\geq$ 2.56, automake $\geq$ 1.8) (build)
  - C++ compiler and library (build)
  - libtool (build)
  - pkg-config (build)
  - gthread-2.0 $\geq$2 .4.7 (build, run)
  - glibmm-2.4 $\geq$ 2.4.7 (build, run)
  - libxml-2.0 $\geq$ 2.4.0 (build, run)
  - openssl $\geq$ 0.9.7a (build, run)
  - e2fsprogs (build, run)
  - doxygen (build)
- Optional (mainly applicable on server side):
  - Swig $\geq$ 1.3.28 (build)
  - Java SDK $\geq$ 1.4 for Java bindings (build, run)
  - Python for Python bindings (build, run)
  - Grid Packaging Tools (GPT) [**?**] (build)
  - Globus Toolkit® 4 [**?**] which contains (build, run):
    - Globus RLS client
    - Globus FTP client
    - Globus RSL
  - LHC File Catalog (LFC) [**?**] (build, run)
  - CppUnit for unit testing (build)
  - Berkeley DB C++ interface (build, run)

The developers of ARC have prepared packages for several Linux distributions*. As a user of any such operating system, the external packages are all readily available.

> Please note that depending on operating system distribution in order to build ARC1 you may need to install development versions of mentioned packages.

3 Getting the software

The software is free to deploy anywhere by anybody. Pre-built binaries for a dozen of Linux platforms can be downloaded from the NorduGrid software repository at `http://download.nordugrid.org/` ("arc1" repository).

The software is released under the Apache Software License version 2 (ASL 2.0) (see the LICENSE file distributed with the software).

You can get the latest source code for ARC1 from the Subversion repository. For details, see:

`http://svn.nordugrid.org`

The NorduGrid software repository hosts the source code, and provides most of the required external software which are not part of a standard Linux distribution.

There are also nightly code snapshots available at:

`http://download.nordugrid.org/software/nordugrid-arc1/nightly/`

Choose the latest date available and download snapshot tarball – for example `nordugrid-arc1-200802201038-snapshot.tar.gz`.

4 Building and Installation

Building from source is currently the recommended way to install ARC Web Services. If you downloaded the tarball, unpack it and traverse into the created directory:

```
tar -zxvf nordugrid-arc1-200802201038-snapshot.tar.gz
cd nordugrid-arc1-200802201038
```

If you obtained the code from the Subversion repository, use the "trunk" directory:

```
cd trunk
```

Now configure the obtained code with:

```
./autogen.sh
./configure --prefix=PLACE_TO_INSTALL_ARC
```

Choose installation prefix wisely and according to the requirements of your OS and personal preferences. ARC1 services should function properly from any location. By default installation goes into `/usr/local` if you omit the "`--prefix`" option. For some modules of ARC1 to work properly you may need to set up the environment variable after installation:

```
export ARC_LOCATION=PLACE_TO_INSTALL_ARC
```

On some systems "`autogen.sh`" may produce few warnings. Ignore them as long as "`configure`" passes without errors. But in case of problems during configure or compilation, collect them and present while reporting problems.

*The nightly build system current comprises Debian, Ubuntu, Fedora, RedHat and OpenSuSE

If the previous commands finish without errors, compile and install ARC1 services:

```
make
make install
```

Depending on chosen installation location you may need to run the last command from `root` account. That should install the following components:

- `sbin/arched` - server executable
- `bin/` - user tools and command line clients
- `lib/` - common libraries used by clients, server and plugins
- `lib/arc/` - plugins implementing Message Chain, Service and Security components
- `include/arc/` - C++ headers for application development
- `libexec/` - additional modules used by ARC1 services - currently only A-REX
- `share/doc/arc` - configuration examples/templates and documentation
- `share/locale` - internationalization files - curently very limited support
- `share/man` - manual pages for various utilities

5 Security and Authorization

ARC1 sevices implement security related features through set of Security Handler and Policy Decision Point components. Security Handler components are attached to message processing components. Each Security Handler takes care of processing own part of security information. Currently ARC comes with the following Security Handlers:

- `identity.map` – associates client's identity with local (UNIX) identity. It uses PDP components to choose local isentity and/or identity mapping algorithm.
- `arc.authz` – calls PDP components and combines obtained authorization decisions.
- `delegation.collector` – parses proxy policy from remote proxy certificate. This Security Handler should be configured under TLS MCC component.
- `usernametoken.handler` – implement the functioanlity of WS-Security Usernametoken profile. It will generate Usernametoken into SOAP header, or extract Usernametoken out of SOAP header and do authentication based on the extracted Usernametoken.

Among available PDP components there are:

- `allow` – always returns positive result
- `deny` – always returns negative result
- `simplelist.pdp` – compares DN of user to those stored in a file.
- `arc.pdp` – compares request information parsed from message and policy information specified in this PDP.
- `pdpservice.invoker` – composes the request, puts request into SOAP message, and invokes the remote PDP service to get the response SOAP which includes authorization decision. The PDP service has similar functionality with `arc.pdp`.
- `delegation.pdp` – compares request information parsed from message and policy information specified in proxy certificate from remote side.

There are examples of A-REX service and Echo service with Security Handlers being used. They may be found in:

`$ARC_LOCATION/share/doc/arc/arex_secure.xml`
and
`$ARC_LOCATION/share/doc/arc/echo.xml`.

There is also a PDP service which implements the same functionality as `arc.pdp`. See `src/service/pdp/README`.

Specifically for `arc.pdp` and `pdpservice`, a formatted policy with specific schema should be managed, for details see:

`$ARC_LOCATION/share/doc/arc/pdp_policy.xml.example`
and
`$ARC_LOCATION/share/doc/arc/Policy.xsd`

For usernametoken handler, there is example about configuration on service side in `$ARC_LOCATION/share/doc/arc/echo.` you can run Echo service by using this configuration file with usernametoken sechandler configuered. For client side, the echo client (`src/client/echo`)can use usernametoken sechandler to authenticate against echo service (see README under `src/client/echo`); there is also a test program in `src/tests/echo/test_clientinterface` which can be compiled and tested against Echo service with usernametoken sechandler configured.

5.1 X509 Certificates

Most of planned and existing ARC1 services use HTTPS as transport protocol so they require proper setup of X509 security infrastructure [**?**]. Minimal requirements are:

- Host certificate aka public key in PEM format
- Corresponding private key
- Certificate of the Certification Authority (CA) which was used to sign the host certificate
- Certificates of CAs of clients which are going to send requests to services (unless of course clients use the same CA as the server).

More information about X509 certificates and their usage in Grid environement can be found in:

`http://www.nordugrid.org/documents/certificate_howto.html`
`http://www.nordugrid.org/documents/ng-server-install.html#security`

For testing purposes you can use the pre-generated certificates and keys available in the code repository:

`http://svn.nordugrid.org/trac/nordugrid/browser/arc1/trunk/doc/sec/TestCA`

Alternatively you may choose to use KnowARC Instant CA service available at:

`https://vls.grid.upjs.sk/CA/instantCA`

The latter is especially usefull if you want to test installation consisting of multiple hosts.

Please remember that it is not safe to use these keys in publicly accessible insallations of ARC. Make sure that even the CA certificate is removed before making your services available to the outside world.

You can put host certificates and private keys anywhere, their location is configurable. Common locations for servers running from root account are:

`/etc/grid-security/hostcert.pem`
and
`/etc/grid-security/hostkey.pem`

respectively.

> Since services have no way to ask for passwords, the content of private key must not be encrypted, neither it should be protected by password. So make sure it is properly protected by means of the file system.

It is possible to configure ARC1 server to accept either a single CA certificate or multiple CA certificates located in the specified directory. The latter option is recommended. The common location is `/etc/grid-security/certificates/` . In that case names of certificate files have to follow hash values of the certificates. These are obtainable by running the command:

```
openssl x509 -hash -noout -in path_to_certificate
```

The corresponding file name for the certificate should be `<hash_value>.0`. The hash value for the pre-generated CA certificate is `4457e417`.

Please make sure the chosen location of certificates is correctly configured in the service configuration file. The configuration for the certificate for TLS MCC should look like this:

```
<KeyPath>/etc/grid-security/hostkey.pem</KeyPath>
<CertificatePath>/etc/grid-security/hostcert.pem</CertificatePath>
<CACertificatesDir>/etc/grid-security/certificates</CACertificatesDir>
```

or

```
<CACertificatePath>/etc/grid-security/ca.pem</CACertificatePath>
```

The same requirements are valid for the client tools for ARC1. You may use the pre-generated user certificate and key located at the same place. Locations of the credentials are configurable, or can be provided to the client tools from the command line.

The set of pre-generated keys and certificates also includes a user certificate in PKCS12 format which you can import into your browser for accessing ARC1 services capable of producing HTML output.

ARC comes with utility `arcproxy` which generates proxy credentials from certificate/private key pair. It provides only basic functionality and is meant for testing purposes only.

IMPORTANT: If during configuration stage you see a message

"OpenSSL contains no support for proxy credentials"

then you won't be able to use proxy credentials generated by utilities like `grid-proxy-init`, `voms-proxy-init` or `arcproxy`. Because of that all user private keys will have to be kept unencrypted.

To avoid providing credential information on command line it is possible to have user configuration file with predefined values. Default location for this file is `/.arc/client.conf`. There is an example installed in `$ARC_LOCATION/share/arc/examples/client.conf.example`. Please edit before copying it into your home directory.

5.2 Proxy Certificate Generation and Usage

As metioned above, ARC comes with proxy generation utility – `arcproxy`, installed in `$ARC_LOCATION/bin`. The usage of `arcproxy` is like:

```
$ARC_LOCATION/bin/arcproxy -P proxy.pem -C cert.pem -K key.pem
  -c validityStart=2008-05-29T10:20:30Z
  -c validityEnd=2008-06-29T10:20:30Z
  -c proxyPolicyFile=delegation_policy.xml
```

By using argument "-c", some constraints can be specified for proxy certificate. Currently, the life-time can be specified by using "`-c validityStart=...`" and "`-c validityEnd=...`", or "`-c validityStart=...`" and "`-c validityPeriod=...`"; the proxy policy can be specified by using "`-c proxyPolicyFile=...`" or "`-c proxyPolicy=...`".

If proxy certificate is used, in the configuration file for service side or client side, the configuration for the certificate for TLS MCC should look like this:

```
<KeyPath>./proxy.pem</KeyPath>
<CertificatePath>./proxy.pem</CertificatePath>
<CACertificatePath>./ca.pem</CACertificatePath>
```

Because normally a proxy certificate file includes the proxy certificate and private key corresponding to the proxy certificate, `<KeyPath/>` and `<CertificatePath/>` are configured to be the same.

Proxy policy can be spefified as constraint. Proxy policy is for constraining identity delegation. Currently, the supported policy is ARC specific policy. Proxy policy is inserted into proxy certificate's "proxy cert info" extenstion in RFC3820's policy language "`NID_id_ppl_anyLanguage`".

6 ARC Server Setup and Configuration

The configuration of the ARC server is specified in an XML file, the location of which is specified as a command line argument with the -c option of "`arched`" daemon. Examples of configuration files with comments describing various elements are available in directory `/usr/share/doc/nordugrid-arc<version>` corresponding to the ARC1 installation.

7 The Echo Service

The *Echo* service is offered purely for testing purposes. It is "atomic" and has no additional dependencies other than what is provided by the Hosting Environment Daemon (HED). An example of an Echo service configuration can be found in `/usr/share/doc/nordugrid-arc<version>/echo.xml`.

7.1 The Echo Client

To use the Echo client, type

```
$ARC_LOCATION/bin/arcecho <message>
```

where `<message>` is the message which the Echo service should return.

8 The A-REX Service

ARC1 comes with OGSA BES compliant Grid job management service called A-REX. To deploy A-REX use example configuration files available in `/usr/share/doc/nordugrid-arc<version>`:

- `arex.xml` – configuration for arched server. Read comments inside this file.
- `arc-arex.conf` – legacy configuration for Grid Manager part of A-REX. This file defines how jobs are managed by A-REX locally. Read and edit it. and edit it to fit your installation. This file defines WS interface of A-REX.

For more detailed information please read Grid Manager documentation [**?**]. Grid Manager runs as part of A-REX service. There is no need to run any additional executable. But you still need to setup its infrastructure as long as you are going to have anything more sophisticated than described in the example configuration. For more information read the previously mentioned document.

Currently, a proper functioning A-REX requires environment variable `$ARC_LOCATION` to be set to the installation prefix of ARC.

A-REX uses HTTPS as transport protocol (although one can reconfigure it to use plain HTTP) so it requires proper setup of X509 security infrastructure. See Section 5.1 for instructions.

Copy example configuration files to some location and edit them. Make sure all paths to X509 certificates and Grid Manager configuration are set correctly. Start server with command:

```
$ARC_LOCATION/sbin/arched -c path_to_edited_arex.xml
```

Look into log file specified in `arex.xml` for possible errors. You can safely ignore messages like *"Not a '...' type plugin"* and *"Unknown element ... - ignoring"*.

If you compiled ARC1 services with Globus support and you see complaints about "libglobus..." and that it cannot open a shared object file, try to add `/opt/globus/lib` to your `$LD_LIBRARY_PATH`:

```
export LD_LIBRARY_PATH=$LD_LIBRARY_PATH:/opt/globus/lib
```

8.1 Testing and Using A-REX (clients)

Now you may use the command line utility "`arcinfo`" to obtain a service description. You can do something like:

```
./arcinfo -c ARC1:https://localhost:60000/arex -l
```

This should produce a description list 1 of the resources A-REX represents. Below you can see an example of proper output.

Listing 1: Example of proper arcinfo output

```
Cluster: localhost
 Health State: ok

Location information:

Domain information:

Service information:
 Service Name: MINIMAL Computing Element
 Service Type: org.nordugrid.execution.arex

Endpoint information:
 URL: https://localhost:60000/arex
 Capabilities:
  executionmanagement.jobexecution
 Technology: webservice
 Interface Name: OGSA-BES
 Supported Profiles:
  WS-I 1.0
  HPC-BP
 Implementor: NorduGrid
 Implementation Name: A-REX
 Implementation Version: 0.9
 QualityLevel: development
 Health State: ok
 Serving State: production
 Issuer CA: /O=Grid/O=NorduGrid/CN=NorduGrid Certification Authority
 Trusted CAs:
  /C=BE/O=BELNET/OU=BEGrid/CN=BEGrid CA/emailAddress=gridca@belnet.be
  /C=FR/O=CNRS/CN=CNRS2-Projets
  /DC=org/DC=ugrid/CN=UGRID CA
  /C=BR/O=ICPEDU/O=UFF BrGrid CA/CN=UFF Brazilian Grid Certification Authority
  /C=DE/O=DFN-Verein/OU=DFN-PKI/CN=DFN-Verein PCA Grid - G01
  /C=PT/O=LIPCA/CN=LIP Certification Authority
  /C=FR/O=CNRS/CN=GRID-FR
```

```
/C=FR/O=CNRS/CN=CNRS2
/C=TR/O=TRGrid/CN=TR-Grid CA
/C=NL/O=NIKHEF/CN=NIKHEF medium-security certification auth
/DC=org/DC=DOEGrids/OU=Certificate Authorities/CN=DOEGrids CA 1
/DC=ch/DC=cern/CN=CERN Trusted Certification Authority
/C=AU/O=APACGrid/OU=CA/CN=APACGrid/emailAddress=camanager@vpac.org
/C=IE/O=Grid-Ireland/CN=Grid-Ireland Certification Authority
/O=Grid/O=NorduGrid/CN=NorduGrid Certification Authority
/DC=RO/DC=RomanianGRID/O=ROSA/OU=Certification Authority/CN=RomanianGRID CA
/DC=bg/DC=acad/CN=BG.ACAD CA
/C=MX/O=UNAMgrid/OU=UNAM/CN=CA
/C=GR/O=HellasGrid/OU=Certification Authorities/CN=HellasGrid Root CA 2006
/C=CL/O=REUNACA/CN=REUNA Certification Authority
/DC=org/DC=balticgrid/CN=Baltic Grid Certification Authority
/C=IT/O=INFN/CN=INFN CA
/DC=me/DC=ac/DC=MREN/CN=MREN-CA
/C=FR/O=CNRS/CN=CNRS-Projets
/C=DE/O=DFN-Verein/OU=DFN-PKI/CN=DFN-Verein User CA Grid - G01
/C=UK/O=eScienceCA/OU=Authority/CN=UK e-Science CA
/C=RS/O=AEGIS/CN=AEGIS-CA
/C=SI/O=SiGNET/CN=SiGNET CA
/C=VE/O=Grid/O=Universidad de Los Andes/OU=CeCalCULA/CN=ULAGrid Certification
/DC=ORG/DC=SEE-GRID/CN=SEE-GRID CA
/C=CH/O=Switch - Teleinformatikdienste fuer Lehre und Forschung/CN=SWITCH Pers
/C=RU/O=RDIG/CN=Russian Data-Intensive Grid CA
/C=HU/O=KFKI RMKI CA/CN=KFKI RMKI CA
/C=JP/O=KEK/OU=CRC/CN=KEK GRID Certificate Authority
/DC=EDU/DC=UTEXAS/DC=TACC/O=UT-AUSTIN/CN=TACC Root CA
/C=AT/O=AustrianGrid/OU=Certification Authority/CN=Certificate Issuer
/C=IL/O=IUCC/CN=IUCC/emailAddress=ca@mail.iucc.ac.il
/DC=TW/DC=ORG/DC=NCHC/CN=NCHC CA
/C=KR/O=KISTI/O=GRID/CN=KISTI Grid Certificate Authority
/DC=LV/DC=latgrid/CN=Certification Authority for Latvian Grid
/DC=NET/DC=PRAGMA-GRID/CN=PRAGMA-UCSD CA
/C=CH/O=SwissSign/CN=SwissSign CA (RSA IK May 6 1999 18:00:58)/emailAddress=ca
/C=MA/O=MaGrid/CN=MaGrid CA
/C=MK/O=MARGI/CN=MARGI-CA
/C=GR/O=HellasGrid/OU=Certification Authorities/CN=HellasGrid CA 2006
/C=TH/O=NECTEC/OU=GOC/CN=NECTEC GOC CA
/C=PL/O=GRID/CN=Polish Grid CA
/C=UK/O=eScienceRoot/OU=Authority/CN=UK e-Science Root
/DC=cz/DC=cesnet-ca/CN=CESNET CA
/C=TW/O=AS/CN=Academia Sinica Grid Computing Certification Authority Mercury
/DC=es/DC=irisgrid/CN=IRISGridCA
/C=JP/O=AIST/OU=GRID/CN=Certificate Authority
/C=JP/O=National Research Grid Initiative/OU=CGRD/CN=NAREGI CA
/DC=BR/DC=UFF/DC=IC/O=UFF LACGrid CA/CN=UFF Latin American and Caribbean Catch
/C=CY/O=CyGrid/O=HPCL/CN=CyGridCA
/DC=CN/DC=Grid/CN=Root Certificate Authority at CNIC
/C=AR/O=e-Ciencia/OU=UNLP/L=CeSPI/CN=PKIGrid
/C=CN/O=HEP/CN=gridca-cn/emailAddress=gridca@ihep.ac.cn
/C=CA/O=Grid/CN=Grid Canada Certificate Authority
/CN=SWITCH CA/emailAddress=switch.ca@switch.ch/O=Switch - Teleinformatikdienst
/DC=CN/DC=Grid/DC=SDG/CN=Scientific Data Grid CA
/C=HU/O=NIIF/OU=Certificate Authorities/CN=NIIF Root CA
/C=IR/O=IPM/O=IRAN-GRID/CN=IRAN-GRID CA
/C=FR/O=CNRS/CN=CNRS
/C=CH/O=Switch - Teleinformatikdienste fuer Lehre und Forschung/CN=SWITCHgrid
```

```
  /C=AM/O=ArmeSFo/CN=ArmeSFo CA
  /C=FR/O=CNRS/CN=GRID2-FR
  /DC=net/DC=ES/O=ESnet/OU=Certificate Authorities/CN=ESnet Root CA 1
  /DC=ch/DC=cern/CN=CERN Root CA
  /DC=IN/DC=GARUDAINDIA/CN=Indian Grid Certification Authority
  /C=DE/O=GermanGrid/CN=GridKa-CA
  /C=SK/O=SlovakGrid/CN=SlovakGrid CA
  /CN=SwissSign Bronze CA/emailAddress=bronze@swisssign.com/O=SwissSign/C=CH
  /DC=EDU/DC=UTEXAS/DC=TACC/O=UT-AUSTIN/CN=TACC Classic CA
  /C=BE/OU=BEGRID/O=BELNET/CN=BEgrid CA
  /CN=SwissSign Silver CA/emailAddress=silver@swisssign.com/O=SwissSign/C=CH
  /C=CH/O=Switch - Teleinformatikdienste fuer Lehre und Forschung/CN=SWITCH Serv
  /C=PK/O=NCP/CN=PK-GRID-CA
  /C=DE/O=DFN-Verein/OU=DFN-PKI/CN=DFN-Verein Server CA Grid - G01
  /C=HR/O=edu/OU=srce/CN=SRCE CA
 Staging: staginginout
 Job Descriptions:
  ogf:jsdl:1.0

Queue information:
 Mapping Queue: default
 Max Total Jobs: 100
 Max Running Jobs: 10
 Max Waiting Jobs: 99
 Max Pre LRMS Waiting Jobs: 0
 Max User Running Jobs: 5
 Max Slots Per Job: 1
 Doesn't Support Preemption
 Total Jobs: 0
 Running Jobs: 0
 Waiting Jobs: 0
 Suspended Jobs: 0
 Staging Jobs: 0
 Pre-LRMS Waiting Jobs: 0
 Free Slots: 10
 Free Slots With Duration:
  P68Y1M5DT3H14M7S: 10
 Used Slots: 0
 Requested Slots: 0

Manager information:
 Resource Manager: torque
 Doesn't Support Advance Reservations
 Doesn't Support Bulk Submission
 Total Physical CPUs: 10
 Total Logical CPUs: 10
 Total Slots: 10
 Non-homogeneous Resource
 Working area is nor shared among jobs
 Working Area Total Size: 15
 Working Area Free Size: 4
 Working Area Life Time: P7D
 Cache Area Total Size: 15
 Cache Area Free Size: 4

Execution Environment information:
 Execution environment is a physical machine
 Execution environment does not support inbound connections
```

```
Execution environment does not support outbound connections
```

Please note that you can run similar `arcinfo` request against any ARC1 service except echo service.

A-REX accepts jobs described in the JSDL language. Example of JSDL jobs are provided in `$ARC_LOCATION/share/doc/` in the files `jsdl_simple.xml` and `jsdl_stage.xml`. To submit job to A-REX service one may use the `arcsub` command:

```
$ARC_LOCATION/bin/arcsub -c ARC1:https://localhost:60000/arex -f $ARC_LOCATION/s
```

If everything goes properly somewhere in it's output there should be a message "Job submitted!", and a job identifier is obtained which will be stored in 'id.xml' file. One can then query job state with the `arcstat` utility:

If everything goes properly somewhere in it's output there should be a message "Submitted the job!". Obtained job identifier is an XML document and is stored in id.xml file. It may then be used to query job state with `arcstat` utility:

```
$ARC_LOCATION/bin/arcstat id.xml
Job status: Running/Submitting

$ARC_LOCATION/bin/arcstat id.xml
Job status: Running/Finishing

$ARC_LOCATION/bin/arcstat id.xml
Job status: Finished/Finished
```

Some of the of A-REX client tools consists of `arcsub`, `arcstat`, `arckill`, `arcget` and `arcclean` commands. For more information please see the man pages of those utilities.

9 Finding more information

Many information about functionality and configuration of various components may be found inside the corresponding configuration XML schemas.

There are several API documents available as well.

10 Contributing

The open source development of the ARC middleware is coordinated by the NorduGrid collaboration. Currently, the main contributor is the KnowARC project [**?**], but the Collaboration is open to new members. Contributions from the community to the software and the documentation is welcomed. Sources can be downloaded from the software repository at:

`http://download.nordugrid.org`

or the Subversion code repository at

`http://svn.nordugrid.org`.

The technical coordination group defines outstanding issues that have to be addressed in the framework of the ARC development. Feature requests and enhancement proposals are recorded in the Bugzilla problem tracking system at:

`http://bugzilla.nordugrid.org`

For a more detailed description, write access to the code repository and further questions, write to the `nordugrid-discuss` mailing list (see NorduGrid website [**?**] for details). Ongoing and completed Grid research projects and student assignments related to the middleware are listed on the NorduGrid website as well.

11 Support, documentation, mailing lists, contact

User support and site installation assistance is provided via the request tracking system available by e-mail to `nordugrid-support@nordugrid.org`. In addition, NorduGrid runs a couple of mailing lists, among which the nordugrid-discuss mailing list is a general forum for all kind of issues related to the ARC middleware.

Feature and enhancement requests, as well as discovered problems, should be reported in the Bugzilla problem tracking system (`http://bugzilla.nordugrid.org`).

Research papers, overview talks, reference manuals, user guides, installation instructions, conference presentations, FAQ and even tutorial materials can be fetched from the documentation section of the NorduGrid website.

Contact information is kept updated on the NorduGrid website.

References