Nordic Testbed for Wide Area Computing And Data Handling

NORDUGRID-TECH-6
7/12/2005

CONFIGURATION AND AUTHORISATION OF ARC (NORDUGRID) SERVICES

Description and Administrator's Manual[*]

A.Konstantinov

[*]Comments to: aleks@fys.uio.no

Contents

1 Introduction

This document describes configuration format and commands common for all services developed us part of ARC (Advanced Resource Connector) [1]. Please note MDS is not *developed* in ARC. It is part of Globus Toolkit^{TM}[2] adopted and modified by NorduGrid project to be included into ARC. Although ARC comes with startup script capable if processing new style configuration format (2) into one used by MDS sections 3 and 4 do not apply to it.

2 Format

Configuration is defined through content of configuration files. Historically each service supports 2 formats of configuration files, so called old and new. By default old configuration files reside in *$NORDUGRID_LOCATION/etc/* and are named after names of services with *.conf* suffix. New configuration is usually placed into single file */etc/nordugrid.conf*. Hence new configuration is usually referenced as *centralised*.

The old configuration file may consist of empty lines, lines containing comment (line starts from #) and command lines with one command per line. Command line starts with command word and may be followed by arguments. Command and arguments are separated by blank spaces. Blank spaces in arguments must be escaped using '\' or arguments must be enclosed in '"'.

The new configuration file may also contain empty lines and comments starting from #. Because it is intended to hold information about multiple services it is separated into sections. Each sections starts from string embraced by '[ ]' and contains name of section on braces. There can be subsections as well like

- *[section name/subsection name/subsubsection name]*.

Each section continues till next section or end of file. Some services may not take ordering of sections and sections into account. While others could be sensitive even to order of commands inside sections. So it is always better to treat whole configuration as order dependent.

One configuration file may have commands for multiple services/modules/programs. Each service get it's own section named after it's name. For example a grid-manager uses section *[grid-manager]*. Subsections are used to reflect internal modular structure of services. Commands in section *[common]* apply to all services.

Sections consists of command line with format

- *command="arguments string"*.

The *argument string* consists of same arguments as in old format. Commands are also mostly same in both kinds of configurations.

Both files support almost same commands. Following commands are defined (examples are given for old format):

3 Common commands

All services developed as part of ARC support following commands:

- ***daemon*** *yes|no* - specifies whether the service should go to background after started. Defaults to *yes*.
- ***logfile*** *[path]* - specifies name of file for logging debug/informational output. Defaults to */dev/null* for daemon mode and *stderr* for foreground mode.
- ***logsize*** *[size [number]]* - specifies maximal *size* of log file and *number* of backups. Default is "*0(unlimited size) 0*".

- ***user*** *[uid[:gid]]* - specifies user id (and optionally group id) to which the service must switch after reading configuration. Defaults to *not switch*.
- ***pidfile*** *[path]* - specifies file where id f the service process will be stored. Defaults to *not write*.
- ***debug*** *number* - specifies level of debug information. More information is printed for higher levels. Currently highest effective number is 3 and lowest 0. Defaults to 2.

4 Authorization

ARC services which have to authorise remote client applications use notion of *group* for authorisation purposes. Each *group* is made of *rules* applied sequentially. If client's credentials pass *all rules* client is treated as belonging to specified *group*.

In old configuration format group definition starts from command ***group*** followed by one argument of name of group (any string). Definition ends with command ***end***. InbetweenIn between there are lines representing rules.

In new configuration format each group is represented by top level section named [***group***] or it's subsection. Each such section represent separate authorisation group and it's name is given by ***name*** command inside that section. If there is no ***name*** command then name of subsection is used.

Authorization is performed by applying set of rules. Rules obey same format as rest of configuration file. Each rules command is made of a *rule word* prepended with optional *modifiers* - [+|-][!]

+ accept credential if matches following rule (positive match, default action),

- reject credential if matches following rule (negative match),

! invert matching. Match is treated as non-match. Non-match is treated as match, either positive ("+" or nothing) or negative ("-").

Processing of rules in every group stops after first positive or negative match or failure is reached. If rule does not match processing continues. Failures are rule-dependant and may be cause by conditions like missing file, unsupported rule, etc.

Following *rule words* and arguments are supported:

- [***subject***] *subject* [*subject* [...]]- accept user with one of specified subjects
- ***file*** [*filename* [...]] - read rules from specified files (format of file similar to Globus grid-mapfile with user names ignored)
- ***vo*** [*ldap://host:port/dn* [...]] - accept users listed in one of specified LDAP directories (uses network connection hence can take time to process)
- ***voms*** *vo group role capabilities* - accept user with VOMS proxy with specified *vo*, *group*, *role* and *capabilities*. '*' can be used to accept any value
- ***group*** [*groupname* [*groupname* [...]]] - accept user already belonging to one of specified groups.
- ***plugin*** *timeout plugin* [*arg1* [*arg2* [...]]] - run external *plugin* (executable or function in shared library) with specified arguments. Execution of *plugin* may not last longer than *timeout* seconds. If *plugin* looks like *function@path* then function *int function(char*,char*,char*,...)* from shared library *path* is called (*timeout* is not functional in that case). Rule mathes if exit code is 0. In arguments following substitions are applied before plugin is started:

  - %D - subject of users's cerificate,
  - %P - name of credentials' proxy file.

- ***all*** - accept any credential

Here is an example of authorization group:

```
[group/admins]
-subject="/O=Grid/OU=Wrong Place/CN=Bad Person"
file="/etc/grid-security/internal-staff"
vo="nordugrid * * admin"
```

References

[1] NorduGrid middleware, the Advanced Resource Connector. http://www.nordugrid.org/middleware/

[2] The Globus Alliance. http://www.globus.org