EUROPEAN MIDDLEWARE INITIATIVE

EUROPEAN MIDDLEWARE INITIATIVE

GRIDSITE – ADMINISTRATOR GUIDE

Document version:	**2.0.0-1**
EMI Component Version:	**1.x, 2.x**
Date:	**February 6, 2013**

This work is co-funded by the European Commission as part of the EMI project under Grant Agreement INFSO-RI-261611.

CONTENTS

GridSite – Administrator Guide

EMI

February 6, 2013

1 BUILD AND INSTALL GUIDE

This Guide explains how to build GridSite from source, and how to install the server components alongside an Apache 2.0 webserver. There is a separate Config Guide which explains how to modify the httpd.conf file, and how to set up other files and directories used by the system. You should look through all of this Building and Installation Guide to decide which is the easiest route for your system.

1.1 INSTALLING WITH RPM

If you are installing on Linux with a binary RPM release, you can skip most of this Guide, install the binary rpm(s) and go straight to the Config Guide.

RedHat 9, Fedora, RHEL, Scientific Linix: This is the simpler case, since the standard release includes a suitable version of Apache 2.0: just install the gridsite-...-1.i386.rpm to get the various GridSite components.

Earlier, eg RedHat 7.3: This is more complicated because you must also install a back-ported Apache 2.0 RPM or build it from source.

GridSite also depends on shared libraries from libcurl and libxml2, and the RPMs distributed as part of the standard RedHat, from 7.3 onwards, are sufficient.

With the RPMs installed, you can proceed to the Config Guide.

1.2 REQUIREMENTS FOR BUILDING GRIDSITE FROM SOURCE

GridSite is currently only supported on Linux, but should be straightforwardly portable to other Unix platforms where the GNU build tools are available.

GridSite consists of a core library (libgridsite[.so|.a]), an Apache module (mod_gridsite.so), a CGI utility (gridsite-admin.cgi) and some command line tools (htcp, urlencode.)

All of the components use the GridSite library, and this in turn depends on libcurl and libxml2. You will need the development versions of these packages installed before you can proceed.

1.3 BUILDING GRIDSITE WITH MAKE

Our download area at https://www.gridsite.org/download/ includes a tar-ball distribution of the sources, which can be unpacked and used to build GridSite from source. (Bleeding-edge developers can get the current snapshot of the same files from our CVS area.)

GridSite needs a copy of the Apache 2.0 include files to build, and the location of this is set by the MYCFLAGS variable in the top-level Makefile. For manual builds, the default MYCFLAGS=-I/usr/local/include/httpd is used. If you wish to use the GridSite module with Apache 2.0 installed elsewhere, you should change the MYCFLAGS variable to point to the includes directory installed by the development part of that Apache 2.0 distribution.

```
make
make install
```

will build all components and install them all under the default locations of /usr/local/[lib|bin|include|sbin] The default prefix for manual builds is /usr/local, as set by the prefix variable in the top level Makefile (/usr is the default for RPMs.)

1.4 BUILDING GRIDSITE WITH RPM

For RedHat Linux and derivatives, building with RPM is recommended. The command make rpm in the top level of the source tree will build the GridSite and htcp binary RPMs in the directory ../RPMTMP/RPMS/i386 relative to the working directory. An SRPM is put into ../RPMTMP/SRPMS This build assumes the Apache 2.0 includes are in /usr/include/httpd.

For other configurations, you can modify the assumed location of the Apache 2.0 includes by changing the MYCFLAGS variable in the rpm target near the foot of the top level Makefile. Building Apache 2.0

If it is not possible to use binary RPMs of Apache 2.0, then it can be built from source using the build-apache2.sh script found in the GridSite docs directory. The script includes instructions on how to build from the tarballs distributed by the Apache Foundation. (it removes the -C option from "configure -C" in the .spec file and builds the RPMs under the current directory.)

If these targets do not work on your build platform, the Makefile and the scriptlets in the included SPEC files are a good starting point for building Apache by hand yourself. The complexities of this are outside of the scope of this Guide, but you are welcome to ask for assistance on the GridSite Discussion List, although http://www.apache.org/ is a better starting point for purely Apache problems.

EMI

2 CONFIG GUIDE

This guide is intended for webmasters setting up GridSite with an Apache 2.0 webserver. We assume you have root access to the server machine to do this. There is a separate Administration Guide for people administrating areas of GridSite websites or fileservers, or managing GridSite's DN List groups. That is, for people managing files on the server rather than the server itself.

2.1 INSTALLATION

We assume you have installed Apache 2.0 and GridSite, using the Building and Installation Guide where necessary. This Config Guide assumes installation has been done under /usr. For an alternative tree like /usr/local, the relative paths should be the same.

Installation should have given you an Apache 2.0 httpd binary at /usr/sbin/httpd and a set of standard Apache 2.0 modules in /usr/lib/httpd/modules/ including the standard mod_ssl and our mod_gridsite.so module.

GridSite 1.1.x also includes some commands and man pages in /usr/bin and /usr/share/man/man1 (urlencode, htcp and other ht* commands.)

2.2 CERTIFICATES

You must also install the CA root certificates of the CA's used by the users you wish to talk to. These should be installed in /etc/grid-security/certificates as files like 01621954.0, and RPMs and tar files for many common European and North American CAs are available via the EU Grid PMA.

This location also has VOMS server certificate RPMs which install into the /etc/grid-security/vomsdir directory. You may also manually install VOMS server certificates into that directory with any filename. (GridSite currently parses the certificate itself when looking for a match, rather than checking the filename.)

The server itself needs a certificate to supply to clients that use HTTPS connections. You should apply for this from your Certification Authority (for example, the UK e-Science CA) and your request must use the advertised hostname of your server (the one that appears in URLs and not, for instance, the canonical name of the host itself.) This advertised hostname should appear in the Distinguished Name of your request. (For example /C=UK/O=eScience/OU=Manchester/L=HEP/CN=www.gridpp.ac.uk) For compatability with standard browsers, the /CN= component should not include any Globus-style service name (so not /CN=host/www.gridpp.ac.uk) If possible, you should also include the advertised hostname as a DNS Subject Alternative Name. Consult your CA first if you're in any doubt about how to compose your certificate request.

Once you've got your certificate, Apache uses the certificate and private key in PEM format. If you obtained your certificate and key in PKCS#12 or .p12 format (eg by exporting from a web browser), you can convert the .p12 file to .pem with the following commands:

```
openssl pkcs12 -in ck.p12 -clcerts -nokeys -out hostcert.pem
openssl pkcs12 -in ck.p12 -nodes  -nocerts -out hostkey.pem
```

Copy the PEM files to /etc/grid-security/ as hostcert.pem (which should be world readable) and hostkey.pem (which should only be readable by root):

```
chown root.root hostkey.pem hostcert.pem
chmod 400 hostkey.pem
chmod 444 hostcert.pem
```

2.3 HTTPD.CONF

/etc/httpd/conf/httpd.conf is the key to configuring the Apache 2.0 webserver. The directives in this file determine which files the server will publish, how they are handled, which areas are writeable and who can access them. Through mod_gridsite.so, the GridSite system itself is configured by directives in this file.

The easiest way to get started is to examine the example httpd.conf files we provide with GridSite, in the doc directory.

Please note: this version of GridSite is not compatible with the SHM SSL session cache - use the DBM or per-process caches instead.

2.4 HTTPD-FILESERVER.CONF

httpd-fileserver.conf is an example configuration file to use Apache/GridSite as a read/write HTTP(S) fileserver, including comments on how to get the server up and running.

2.5 HTTPD-WEBSERVER.CONF

httpd-webserver.conf is an example configuration file to use Apache/GridSite as a Web Server (that is, primarily for interactive use with a browser) including comments on how to get the server up and running.

2.6 GRIDSITE DIRECTIVES

The mod_gridsite reference lists all the GridSite httpd.conf directives.

To start serving files, make a directory /var/www/htdocs owned by nobody.nobody, including the .gacl access control file described below, and add the following directive to the HTTPS <Directory> section:

```
GridSiteMethods GET PUT DELETE
```

If you wish to accept Globus GSI Proxies as well as full X.509 user certificates, set GridSiteGSIProxyLimit to the depth of proxy you wish to accept. (As a _rough_ guide: 0=No Proxies; 1=Proxy on user's machine; 2=Proxy owned by running Globus job; 3=Proxy delegated by a Globus job.) GACL access control

The GACL reference explains the XML access control files used by GridSite. These allow flexible policies to be written, in terms of X.509 user certificates, GSI proxies, VOMS attribute certificates, DN List groups and DNS hostnames.

For example, to give all clients read and list permission:

```
<gacl>
<entry>
  <any-user/>
  <allow><read/><list/></allow>
</entry>
</gacl>
```

To enable writing, add DN List, Person or VOMS entries to the file. For example:

```
<gacl>
<entry>
  <any-user/>
  <allow><read/><list/></allow>
</entry>
<entry>
  <person>
```

```
  <dn>/C=UK/O=eScience/OU=Manchester/L=HEP/CN=Andrew McNab</dn>
  </person>
  <allow><write/></allow>
</entry>
</gacl>
```

The GACL file that governs a directory is stored as .gacl in that directory. If no .gacl is present, then GridSite will search the parent directories in ascending order until one is found.

EMI

3 ADMINISTRATION GUIDE

This Guide is intended for people administrating areas of GridSite websites or fileservers, or managing GridSite's DN List groups - that is, how to use GridSite to manage other people's access to parts of the site - for example, people's write access to areas devoted to specific subprojects.

There is a separate User Guide which explains how to authenticate to the server with X.509 certificates, and how to manage files via a standard web browser or with command-line HTTPS clients. You should be familiar with the User Guide to fully understand this Admin Guide.

You may also find the Config Guide useful to understand how the Apache webserver is configured with GridSite extensions. If you are also the Apache webmaster for your site, you will definitely need to read the Config Guide to create the httpd.conf file. However, if you only need to manage webpages and files, then this Admin Guide and the User Guide should be sufficient.

3.1 GROUPS AND DN LISTS

GridSite defines groups of people using plain text DN Lists - that is, lists of people's certificate DNs. Each DN List has a URL which uniquely identifies the list (and may also allow other sites to obtain the list and use it themselves.) For example, the list of all GridPP members is https://www.gridpp.ac.uk/dn-lists/gridpp (note that it's https:// not http:// - this means that other sites that download the list can check the certificate of www.gridpp.ac.uk and know they're talking to the authoritative source of the lists.)

The system can also have a number of other DN Lists which are associated with specific groups of people and perhaps with specific areas of responsibility of the website. If the DN List directory URI is /dn-lists/ then there is a full list of the DN Lists exported by the server at that URI (for example, https://www.gridpp.ac.uk/dn-lists/)

If you have permission to modify a DN List, you can start changing it by going to /dn-lists/ (via HTTPS), using the "Manage directory" button and finding the URL of your DN List in the listings. You may need to go down into a subdirectory to find your list. For example, https://www.gridpp.ac.uk/dn-lists/atlas is in the atlas subdirectory of /dn-lists/ (You may wish to bookmark the listing of such a directory if you frequently work with one.)

DN List directories are managed by the ACLs described in the next section, and if you have write permission, you can edit the lists already there, and add new lists with the same prefix (this means you can readily create your own subgroups.)

3.2 ACCESS CONTROL LISTS

DN Lists appear in the Grid Access Control Lists (GACL) used by GridSite. These are stored as .gacl files in directories: if the .gacl file is present, it governs access to the directory; if it is absent, then the parent directories are searched upwards until a .gacl is found.

The GridSite GACL Reference explains the XML format of these files, but they can be edited using the ACL editor built into the GridSite system by people who have the Admin permission within the ACL.

If you have this permission in a given directory, when you view directory listings or files in that directory you will see the option "Manage Directory" in the page footer. This allows you to get a listing of the directory and the .gacl file will appear at the top if it's present. If not, then there will be a button to create a new .gacl file with the same permissions as have been inherited by that directory from its parent.

GACL allows quite complex conditions to be imposed on access, but normally you can think of an ACL as being composed of a number of entries, each of which contains one condition (the required credential) and a set of allowed and denied permissions.

Credentials can be individual user's certificate names or whole groups of certificate names if a DN List is given. (You can also specifiy hostname patterns using Unix shell wildcards (eg *.ac.uk) or EDG VOMS attribute certificates - see the GACL Reference for details.)

Permissions can be Admin (edit the ACL), Write (create, modify or delete files), List (browse the directory) or Read (read files.) Permissions can be allowed or denied. If denied by any entry, the permission is not available to that user or DN List (depending on what credential type was associated with the Deny.)