OrangeFS Windows Client

OrangeFS Development Team

September 2011

OrangeFS Windows Client

The OrangeFS Windows Client enables Windows systems to access OrangeFS/PVFS2 file systems. This document will guide you through the installation, operation and configuration of the Client. Complete information about OrangeFS can be located at http://www.orangefs.org.

Contents

Installation

The OrangeFS Windows Client can be installed using the self-extracting installer for the appropriate processor type (32-bit or 64-bit). Download and run `orangefs-client-win32.exe` or `orangefs-client-win64.exe`. At this time, you cannot run the 32-bit installer on a 64-bit OS.

Requirements

- Operating System
  - Windows Vista or Windows 7
  - Windows Server 2008 or Windows Server 2008 R2 (all editions; Server Core installation not currently supported)
- Hardware
  - 30MB disk space
  - Other requirements dependent on usage; minimum requirements very low

Running the Installer

Note: It is best to run the installer as an administrative user (Administrator, for example).

1. Click **Next** on the Welcome page.
2. You will be prompted for an installation location. Use the default or select a different location and click **Next**.
3. Click **Install** to install the Client.
4. You will be prompted for file system URI, mount point and user mapping type.
   File System URI: The DNS name/IP address and port number of a OrangeFS/PVFS2 file system server, in a URI format. The format is `tcp://{hostname}:{port}/{FS name}`.
   Example: `tcp://myhost.com:3334/pvfs2-fs`. Port 3334 is the default.
   Mount Point: A drive letter (E:-Z:) where the file system will appear. Select **Auto** to use the first available drive letter (starting with E:).
   User Mapping: Corresponds to a type of user mapping described below. If you are not sure, select **List** as the settings can be changed later.
   Click **Next** to continue.
5. You will then be shown a page based on the user mapping type selected.
   List Mapping: Enter one Windows user ID and the OrangeFS (Linux/UNIX-based) UID and primary GID for mapping. Additional users must be added to the configuration file manually.
   Certificate Mapping: You may use each user's profile directory or a prefix directory.
   LDAP Mapping: You may use defaults for Microsoft Active Directory or Novell eDirectory. You may then specify LDAP settings.
   For complete details on the user mapping options available, see the section, "User Mapping," later in this document.
   Click **Next** to continue.
6. On the last page, you may choose to start the OrangeFS services. Doing this will mount the OrangeFS file system. If your configuration isn't complete, you can leave the box unchecked and start the service later (see the next section, "Operation").
   Click **Finish** to complete the installation.

Uninstalling the Client

1. From the **Windows Start Menu**, select **Control Panel**, then **Programs and Features**.
2. Locate and select the **OrangeFS Client** item, and click the **Uninstall** button above.
3. Follow the uninstaller steps to remove the Client.
4. Remove configuration files under `C:\OrangeFS\Client` (by default) and the `C:\OrangeFS\Client` directories.

Operation

To run the Client, use the Services utility (**Control Panel** > **Administrative Tools** > **Services**) to start the DokanMounter and OrangeFS Client services. (Dokan is third-party software that mounts the file system transparently.) You can stop the services using the same utility. It is normally not necessary to stop the DokanMounter service. The services are set to start automatically on system startup.

When running, the file system appears as a removable drive at the drive letter (E:-Z:) specified in the configuration file (see the next section, “General Configuration”).

Currently the Client can only mount one file system at a time.

You can interact with files and directories in the file system in the same way as local files. For example, they can be viewed in Explorer, listed in Command Prompt and accessed using program API functions, such as `fopen`.

One limitation is that directories can be created in Explorer, but files cannot. The way to create a file is to use the application corresponding to the file type and save it to the file system. Security is enforced by mapping the Windows user ID to an OrangeFS Linux/UNIX-based UID. (For more information, see the later section, “User Mapping.”) The user ID then has permissions to files and directories based on the OrangeFS UID. New files created on Windows will have the mapped UID as owner, the mapped primary GID as group, and permissions mask 755 (rwxr-xr-x). You may mark the file as read-only on Windows to remove owner write permissions.

To troubleshoot problems, check the Application Event Log in the Event Viewer utility. You can also turn on detailed debugging (see the next section).

For information on configuring the file system to meet your application’s needs, see the OrangeFS system documentation.

General Configuration

The Client uses two configuration files, `orangefstab` and `orangefs.cfg`. These are located in the installation directory `C:\OrangeFS\Client` by default.

Because the configuration files can be altered to change security information, only administrative users should be able to change it (For security information, see your Windows documentation.)

The `orangefstab` file is in the format used for Linux/UNIX mtab (file system mounting) files. On Windows, only the file system URI is of real importance.

Here is a sample `orangefstab`:

```
tcp://orangefs.acme.com:3334/pvfs2-fs /mnt/pvfs2 pvfs2 defaults,noauto 0 0
```

Since only one file system can be mounted, only one line can be used.

The first field is the important one. It is a URI that specifies an OrangeFS file system server. The format is `tcp://{hostname}:{port}/{FS name}`. The only protocol supported on Windows is TCP. The default port is 3334. The file system name can be determined from the server configuration file (default pvfs2-fs).

The second field is the internal UNIX-style mount point. This value should be the same for all clients (Windows or Linux/UNIX). The other fields should be left as-is above.

The bulk of the configuration information is contained in `orangefs.cfg`. The file is a text file that contains lines in the form:

```
{keyword} [option value]
```

You can also specify comments using the # character:

```
# This is a comment.
```

The first keyword to discuss is `mount`. This keyword is used to specify the drive letter that the file system mounts on.

Example:

```
mount O:
```

will mount the file system on O: drive. (You must include the colon.) If the mount keyword is not used, the first alphabetically available drive, starting with E:, is used.

The `user-mode` keyword selects the user mapping mode. It **must** be included in the file, or the Client will not start. The option value must be `list`, `certificate` or `LDAP`.

Example:

```
user-mode list
```

For more on user mapping keywords, see the next section, “User Mapping.”

Finally, the `debug`, `debug-file` and `debug-stderr` keywords are used to get detailed debugging information. If you specify the `debug` keyword by itself, Client-related messages are recorded in `orangefs.log` in the installation directory (`C:\OrangeFS\Client` by default). You can change the name and location of the log file by using the `debug-file` keyword:

```
debug-file C:\Temp\myfile.log
```

You can also use any of the debugging flags available with OrangeFS. For a list of these flags, see the OrangeFS system documentation. The Client flag is `win_client`. With this line:

```
debug win_client io msgpair
```

you would record debugging information about the Client, I/O and message pair operations. The `debug-stderr` keyword is used with no option value, and causes debugging messages to be printed to the console. This keyword is only useful if `orangefs-client.exe` is run as a normal executable (not as a service).

User Mapping

The Client maps Windows user IDs to OrangeFS Linux/UNIX-based UIDs for authentication. The `user-mode` keyword in `orangefs.cfg` specifies the type of user mapping. There are three types of user mapping, detailed below.

List Mapping

This simple form of mapping allows you to list Windows user IDs and their corresponding OrangeFS UIDs and primary GIDs. The list is created in `orangefs.cfg`. Here is the format of each line:

```
user {Windows User ID} {UID}:{GID}
```

Example:

```
user ofsuser 500:100
```

Lines specifying users must come after the line containing the `user-mode` keyword.

File operations originating from the specified Windows user ID will be carried out on OrangeFS as the specified UID.

Certificate Mapping

The Client can use X.509 certificates to identify users. The certificates contain the UID and GID to be used on the OrangeFS server. Because OrangeFS currently expects trusted clients, the certificates *do not provide true security*. However, they will limit the actions of typical users, such as deleting files they do not own. Note that support for untrusted clients will be added to OrangeFS in an upcoming release.

Identifying Certificate Format

The certificate that identifies the OrangeFS user is called the identifying certificate. It is a proxy certificate, which allows authorization on behalf of an "end entity," in this case a user. This user is represented by a user certificate.

Proxy certificates contain authorization information in a data field known as a policy. For the Client, the policy is a UTF-8 string in the form `{UID}/{GID}`. For OpenSSL, the proxy specification for UID 250 and primary GID 100 looks like:

```
language=id-ppl-anyLanguage
pathlen=0
policy=text:250/100
```

More information on generating this certificate is provided below.

Certificates and Validation

The identifying certificate is only useful if it can be validated against its signing certificate. The signing certificate may also need to be validated against the certificate that signed it, and so on, forming a certificate chain. Ultimately, the chain must end at the trusted, self-signed certificate of a certificate authority (CA). More information on how to obtain these certificates is provided below.

Globus Toolkit/MyProxy Certificates

Since OrangeFS is often run in a grid environment, instructions for generating client certificates using Globus Toolkit, which contains a package called MyProxy, will be provided.

Installing Globus Toolkit

Installation instructions for Globus Toolkit can be obtained at http://www.globus.org/toolkit/docs/latest-stable/. The Quickstart instructions will provide a default configuration for MyProxy, including a CA called `SimpleCA`.

There are many different security options that can be configured. For example, a third-party certificate authority may be used. As long as the identifying certificate follows the format above, the client will accept the certificate.

Locating the CA Certificate

If `SimpleCA` is being used, the default CA certificate is `$HOME/.globus/cacert.pem`, where `$HOME` is the home directory of the user who installed SimpleCA, typically `root`.

If a third-party CA is being used, the certificate will be located in an implementation-dependent location. The security administrator of the grid should be able to locate the file.

The CA certificate needs to be copied to the Client system after installing the Client. For the location of the file, see "Client Certificate Locations" below.

Using Grid-based certification

To use grid-based certification, the user must first have a user certificate. To obtain this certificate, the user runs `grid-cert-request` to generate a certificate request file. At that time, the user specifies the certificate pass phrase. This file is then e-mailed (for example) to the CA organization, where a human agent will review the request and return a user certificate signed by the CA certificate. The certificate will be stored in `$HOME/.globus/usercert.pem`. If the grid installation is using `SimpleCA`, the certificate request can be processed by a local administrator using the `grid-ca-sign` command.

The `grid-proxy-init` command can then be used to obtain a proxy certificate. A file (`cert-policy`, for example) is needed to contain the policy text, which is formatted `{UID}/{GID}`. The file would contain `250/100` for a user with UID 250 and GID 100. The `grid-proxy-init` command can be used to generate the proxy certificate with our example `cert-policy` file, as follows:

```
grid-proxy-init -policy cert-policy -pl id-ppl-anyLanguage
```

The user enters the certificate pass phrase and the proxy certificate is generated. To simplify this command, the OrangeFS installation package includes the script `Tools\pvfs2-grid-proxy-init.sh`. This will generate the policy file and run `grid-proxy-init`. The resulting proxy certificate is stored by default at `/tmp/x509up_u{UID}`. *Example*: `/tmp/x509up_u250` for UID 250.

This certificate must be transferred to the Client system, along with the user certificate (see above). For the file location, see "Client Certificate Locations" below. The proxy certificate must be renamed `cert.0`, and the user certificate `cert.1`.

Delegating Identities for Clusters

The use of identifying proxy certificates allows the identity of the user to be separated from the actual Windows user ID making a file system request. This ability is useful for clusters.

For example, a user has Windows user ID JSmith. However, when he executes a job on a cluster node, the job scheduler uses Windows user ID ClusterUser.

The system administrator could set the certificate directory prefix to `C:\ClusterWork`. A directory called `ClusterUser` would be created under `ClusterWork`. The job scheduler would transfer certificates to the `C:\ClusterWork\ClusterUser` directory. When `ClusterUser` makes file system requests, it will use JSmith's certificates, so requests will be

made using JSmith's UID on the file system. When a different user uses the node, that user's certificates will be used.

Certificate Expiration and Renewal

For performance, the Client caches the OrangeFS user identity (UID/GID) until the proxy certificate expires. By default, Globus Toolkit proxy certificates expire after 12 hours. If jobs requiring more time are expected, a means for the user to renew the certificate should be provided.

One way to do this is to have the user to run `grid-proxy-init` again. This will overwrite the current proxy. Then the new proxy certificate can be transferred to the Client system (overwriting the current certificate) without interrupting the current job.

Client Certificate Locations

The certificates are stored as PEM-format files on the Client system. The identifying certificate's name is `cert.0`. Because the identifying certificate is associated with a Windows user, by default it is stored in its user's profile directory. On most systems this is `C:\Users\{username}`.
Example: `C:\Users\jsmith`

Alternatively, a certificate `prefix` directory can be specified in the client configuration file, by default `C:\OrangeFS\Client\orangefs.cfg`. Use the `cert-dir-prefix` keyword to specify this directory. The user's username will be appended as a directory name to the prefix directory. Here's an example configuration file statement:

```
cert-dir-prefix M:\OrangeFS Users
```

For user jsmith, the identifying certificate will be `M:\OrangeFS Users\jsmith\cert.0`.

The identifying certificate must be verified by its end-entity (sometimes called a user) certificate. This certificate should be placed in the same directory as the identifying certificate, with the name `cert.1`. Additional intermediate certificates can be placed in the same directory with names `cert.2`, `cert.3`, and so on.

The CA certificate is placed in the OrangeFS CA directory with the name `cacert.pem`. By default this is `C:\OrangeFS\Client\CA\cacert.pem`. This path can be changed in the configuration file using the `ca-path` directive in the configuration file:

```
ca-path M:\OrangeFS Certificates\orangefs-cacert.pem
```

If this certificate changes, the Client service must be restarted.

LDAP Mapping

LDAP (Lightweight Directory Access Protocol) mapping allows the Windows user ID to be looked up in an identity directory that supports LDAP. Some example LDAP directories are Microsoft Active Directory and Novell eDirectory. Consult your directory documentation for information on LDAP.

LDAP options are specified in `orangefs.cfg`. The keywords described below must follow the `user-mode ldap` line.

Connecting over LDAP

First you must specify the host computer running LDAP. This done with the `ldap-host` keyword in the following format:

```
ldap-host ldap[s]://{hostname}:{port}
```

If `ldaps` is specified, a secure connection is used; otherwise, the connection is plain text. The default secure port is 636, and the default plain text port is 389, but you alter the port as shown above. *Example*:

```
ldap-host ldaps://myldaphost.acme.com:1636
```

You may bind to the directory anonymously if it allows, or you may specify a user and password with the `ldap-bind-dn` and `ldap-bind-password` keywords:

```
ldap-bind-dn {bind (login) user DN}
ldap-bind-password {password}
```

Example:

```
ldap-bind-dn cn=orangefs-user,ou=special,o=acme
ldap-bind-password S3crt!
```

Because the password is stored plain text in the configuration file, you must give the binding user minimal rights to the directory. For more information, see “LDAP Security” below.

Search Options

The Client will search LDAP for the Windows user ID making the file system request. The search options configure how the directory will be searched.

First, the `ldap-search-root` keyword specifies the DN of the directory container object where the search should begin.

```
ldap-search-root ou=cluster-users,o=acme
```

The `ldap-search-scope` keyword can be one of either `onelevel` or `subtree`. If `onelevel` is specified, only the object specified with `ldap-search-root` is searched—no descendant objects (sub-containers) are searched. If `subtree` is specified, the object specified with `ldap-search-root` is searched along with all descendant objects. The default is `onelevel`.

```
ldap-search-root subtree
```

The Client will form an LDAP search string in the form:

```
(&(objectClass={ldap-search-class})({ldap-naming-attr}={Windows user ID}))
```

The `ldap-search-class` keyword specifies the object class that the user object must be. Typical values are `user` or `inetOrgPerson`.

```
ldap-search-class User
```

The `ldap-naming-attr` keyword indicates the attribute on the user object that must exactly match the Windows user ID. Consult your documentation for whether the comparison is case-sensitive (typically it is not). Typical values might be `cn` or `name`.

```
ldap-naming-attr cn
```

Attribute Options

The `ldap-uid-attr` and `ldap-gid-attr` keywords specify the attributes which store the OrangeFS UID and primary GID respectively. The Client retrieves these values for use on the file system.

```
ldap-uid-attr uidNumber
ldap-gid-attr gidNumber
```

LDAP Security

Because the LDAP binding password is stored as plain text, you must give the binding user minimal rights to the LDAP directory. Alternatively, minimal rights can be given to users who bind anonymously—no password is stored in this case. Here are rights to consider:

- Rights to search objects in the search root and below
- Rights to read the object class, naming attribute, UID attribute and GID attribute from searchable objects
- No write/delete/administrator rights

For performance, UID/GID credentials are cached for a time after lookup. If rights need to be revoked, the OrangeFS Client service should be restarted.

You should also use an encrypted connection to LDAP if possible, by specifying `ldaps` in the host URI.

Troubleshooting

As mentioned earlier, startup errors will be logged to the Windows Event Log.

The configuration file has some strict requirements, so the Client will log an error to Event Log and exit. The event message should give an exact explanation of the problem with the configuration file. Correct the problem and restart the OrangeFS Client service.

Make sure network connectivity is available between the Client system and the server hosting OrangeFS. Check firewall settings and network access lists.

For information about the `debug` and related keywords, see the earlier section, “General Configuration.” The generated file `orangefs.log` can be used to diagnose problems. A file named `service.log` is also created in the installation directory when debugging is enabled, and can provide more detail on startup errors.

Note that many debug messages are low-level and require extensive knowledge of OrangeFS/PVFS2 to interpret. For more information, consult the OrangeFS and PVFS2 system documentation.

Free and commercial support is available at http://orangefs.org.

Appendix A: Source Code

The intention of the OrangeFS team is to provide all source code needed for building the Client.

Currently, a source code package is available on http://orangefs.org. (The Windows package is separate from the Linux/UNIX package.) Build instructions will be released at a later date.