CATACOMB WebDAV Server

Howto use the ACL version of Catacomb

29.11.2006 - Markus Litz
DLR (German Aerospace Center)
Simulation and Software Technology

This is a updated brief explanation about the ACP capable version of Catacomb. If you are looking for a how-to of the stable version of Catacomb please take a look at the Catacomb website at <http://catacomb.tigris.org>. This explains what this version of Catacomb is, what is different from the other versions and how to install and how to use it. Catacomb is a module for Apache [2] and repository interface for mod_dav <http://www.webdav.org/mod_dav>. The ACL version of Catacomb is still not stable, so be careful with use in productive environment.
Contents

1 Introduction 1

2 What is Catacomb? 2

3 Installation 2
   3.1 What do I need? .................................................. 2
   3.2 Software installation ............................................ 3
   3.3 Configuration .................................................. 4
   3.4 Fire your Apache ................................................ 6

4 DASL Clients 6
   4.1 Read/Write resources ......................................... 7
   4.2 Search ......................................................... 7
   4.3 ACL ............................................................ 7

5 Trouble shooting 7

6 Conclusion 8
   6.1 Call for participation ......................................... 8

References 8

1 Introduction

There are many web servers. Suppose you have one web server which has millions of documents. How would you find particular documents whose author is Jim and which contain a key word such as java. A Catacomb is a perfect solution for that.

The Web-DAV protocol is an extension of HTTP and it enables the user to write to the HTTP server, give properties for resources and other functions. DASL is extension of Web-DAV and it enables the user to search resources in the Web server. Catacomb is a DASL implementation for Apache and mod_dav [1]. It also support basic access control mechanisms.

This document explains how to install the Catacomb module and use it.
2 What is Catacomb?

Briefly, it is:

- DASL implementation for mod_dav and Apache.
- MySQL Back-ended storage system.

DASL specifies the SEARCH method for searching and locating resources. In other words, it enables the user to search resources in the web server using the DASL protocol. Catacomb is the implementation of DASL which allows web server to support a search function.

To implement the search function, Catacomb uses MySQL. That means if you are saving some resources using PUT or MKCOL which are Web-DAV methods, all of the resources and their information are saved in not a file system, but in DBMS.

The access control protocol is used to grant or deny certain rights on the server on a specific user. This Catacomb Version introduces the basic access control protocol.

3 Installation

3.1 What do I need?

You need:

- Linux or Unix system which can be installed Apache - MS-Windows with Cygwin may be possible as well, but is still untested and not recommended, same with AIX.
- Apache 2.0.36 or higher version - at this time it’s not possible to use Catacomb with Apache 2.2.X.
- MySQL 3.2.33 or higher version
- mod_dav_respo ACP version, currently only available from svn branch.
- mod_auth_mysql or another apache auth mechanism.
- subversion client
3.2 Software installation

You need MySQL, and you can download it from the MySQL Web Site. I assume that MySQL is already installed on your machine. There is no special MySQL option for mod_dav_dasl, so you don’t need to reinstall MySQL for Catacomb, if MySQL is already installed. If you have some difficulties to install MySQL, please visit the MySQL homepage. The default MySQL installation directory is /usr/local. Please be sure you have the header files of the MySQL client installed too. In most distributions the MySQL-client header files are a separate package.

You need Apache 2.0.36 or later version for Catacomb. You can download the latest Apache server from the Apache Web Site. When you install Apache, we need two options, enable-dav and enable-so. Catacomb is a mod_dav interface, so we need to enable mod_dav. The Catacomb object is a shared style object, so your Apache must support shared objects. The default Apache installation directory is /usr/local/apache2. You can enable two option by configuring with this command:

\[ \texttt{<shell> ./configure --enable-so --enable-dav --enable-dav-fs} \]

You may also set a alternative installation dir with \texttt{prefix=}/home/whatever/apache, otherwise it installs in /usr/local/apache2.

Now, it is time to get and install Catacomb. You can find the subversion server with the ACP branch in the catacomb/tigris subversion repository \cite{3}. A simple way to get the sources is to use the subversion console client and check out with:

\[ \texttt{<shell> svn co http://catacomb.tigris.org/svn/catacomb/trunk/catacombsvn/branches/acp-} \]

When you configure the Catacomb, you need to select two options, with-apache and with-mysql. Using \texttt{--with-mysql} option, you need to specify the mysql C client library directory, and using \texttt{with-apache}, you will specify the apache directory. The default directory is the \texttt{/usr/local} directory for MySQL and the \texttt{/usr/local/apache2} directory for Apache. We recommend to use the \texttt{with-debug} option so that you get full debug output.
<shell> cd catacomb-acl
<shell> ./configure --with-apache=/usr/local/apache2 --with-mysql=/usr/local --with-dependencies
<shell> make
<shell> make install

For mod_auth_mysql or other apache auth mechanism no catacomb specific options are necessary, so simply install like described in the specific manual.

3.3 Configuration

We need two configurations such as:

- Apache configuration
- DBMS initialization

You want to add these lines into the /usr/local/apache2/conf/httpd.conf file.

Listing 1: Apache modul declaration

```xml
DavDBMSHost localhost
DavDBMSdbName reposDB
DavDBMSUser root
DavDBMSPassword root
DavDBMSTmpDir /tmp/
DavDBMSFileDir /usr/local/apache2/htdocs/repos/
DavDBMSMaxFileSize 102400

<Location /administration>
 DAV repos
 ModMimeUsePathInfo on
</Location>

<Location /repos>
 DAV repos
 AllowOverride All
 DAVDepthInfinity on
 AuthType Basic
 ModMimeUsePathInfo on
 AuthName Catacomb-WebDAV-Server
 require valid-user
</Location>
```
Here are the directive explanations:

- **DavDBMSHost**: specify your MySQL server.
- **DavDBMSDBName**: Your DB name. To make new DB for Catacomb is recommended.
- **DavDBMSId**: User id for the DB. Make sure the user has proper permission for the DB.
- **DavDBMSPass**: User password for the id. If your id doesn’t have a password, just omit DavDBMSPass directive.
- **DavDBMSTMpDir**: Catacomb needs a temporary directory to manipulate the files. Make sure your HTTP daemon has writing permission in the directory.
- **DavDBMSFileDir**: The directory to save files if not all should go into the database.
- **DavDBMSMaxFileSize**: The (optional) filesize about which files going to filesystems instead in db. Set to 0 if you don’t need that.
- **Location**: specifies the URI information. You need to specify DAV provider as "repos". For ACL capability, we need two Locations. One for the repository, and one for all the data of the users and groups.

Second configuration job is creating DB and tables for mod_dav_dasl. DB which is already exist for other usage is reusable for Catacomb, but creating a new DB for Catacomb is strongly recommended. You can create new DB using the mysqladmin command:

```shell
tool>mysqladmin create reposDB
```

This command will create DB named reposDB. There are several tables used by Catacomb. You can create tables by importing table.sql in the Catacomb source directory using this command:

```shell
tool>mysql reposDB < table.sql
```

You might need -u and -p options for MySQL, if you specified a database user for the DB. Finally, you need to import data for your root information of the Catacomb enabled directory. In our example, the directory name is
4 DASL Clients

If you want to test your DASLed server very quickly, the simplest and best way is using telnet. Do telnet to your server and the port. Let's suppose our DASLed server is localhost and port is 8888. We can test by typing the OPTION method:

```
ocean 10> telnet localhost 8888
Trying 127.0.0.1...
Connected to localhost.
Escape character is '\'.
OPTIONS /das1/ HTTP/1.1
Host: localhost

HTTP/1.1 200 OK
Date: Wed, 08 May 2002 09:16:13 GMT
Server: Apache/2.0.36-dev (Unix) DAV/2 DASL/0.1
DAV: 1
DAV: <http://apache.org/dav/propset/fs/1>
MS-Author-Via: DAV
Allow: OPTIONS, GET, HEAD, POST, DELETE, TRACE, PROPFIND, PROPPATCH, COPY, MOVE, SEARCH
```
5 TROUBLE SHOOTING

DASL: <DAV:basicsearch>
Content-Length: 0
Content-Type: text/plain; charset=ISO-8859-1

If you can see SEARCH method in Allow field in the response, your DASLed server is working properly. You also may use the commandline base webdav client cadaver [4], which is great for testing webdav servers.

4.1 Read/Write resources

Unfortunately, you can not copy resources or make directory directly like usual Apache file system directories, since Catacomb use MySQL to save all resources and their data. So you need to use Web-DAV client to read/write your resources in the DASLed server. There are many WebDAV client programs. You can use one of these client and read/write your resources. If you want to only read your resources, a web browser are a good tool. For more information about WebDAV software, visit the WebDAV resource site.

4.2 Search

So far, there is not many DASL clients. When we write this HOWTO, cadaver DASL patch is on developing. The cadaver [4] client will support DASL, when you read this HOWTO. Hopefully there will be lots of DASL clients very soon.

4.3 ACL

There are only a few ACL-capable WebDAV clients. For testing and reading ACLs DAVExplorer [5] is recommend. Setting new permissions should be done with WebDAVPilot [6], a plugin for the eclipse framework.

5 Trouble shooting

If your DASLed server doesn’t work properly, reading error_log in the Apache’s log directory is good idea. Here are simple check lists:

- Did you configure Apache with enable-so and enable-dav option?
6 Conclusion

The Catacomb support of the access control protocol has just started, so please support us with potential error descriptions. We hope Catacomb and this HOWTO could help your Internet environment.

6.1 Call for participation

Catacomb is open source project under Apache License. So anybody is welcome to the project. Please join our development mailing list: catacomb@webdav.org [7], if you have interest.

References


- MySQL daemon is running before Apache server is running?
- Did you specify the right DB information in httpd.conf?
- Temporary directory which is specified httpd.conf is readable and writable?
- ...