Setting up a Lightweight DAS server
... and connecting it to Ensembl

Date: May 27, 2003

Author: Andreas Kähäri, European Bioinformatics Institute (EBI)
andreas.kahari@ebi.ac.uk

Version 1.3

1. Introduction

This document is a log of how I installed the LDAS server for personal use. One may read it as a short step-by-step guide on how to set up a Lightweight DAS (LDAS) server on a UNIX workstation and have its data attached as tracks in the “ContigView” display of the Ensembl genome browser (or for whatever purpose one might want to set up a DAS service).

The LDAS server was written by Lincoln Stein at the Cold Spring Harbor Laboratory, and is based on Perl and the BioPerl framework. This is by no means the only easy to use DAS server available, and the site at

```
http://www.biodas.org/
```

also lists the Dazzle server which is based on Java and the BioJava toolkit. If you want to set up a Dazzle server to use with Ensembl, wander off to the Ensembl documentation page at

```
http://www.ensembl.org/Docs/
```

and look for the “DAS Server Install document”, written by Tony Cox at the Sanger institute. That text might also be used as a complement this document.

After reading and carrying out the instructions in this text, you should have a LDAS server running on your UNIX machine, and a simple sample DAS feature track displayed in the Ensembl browser.

The LDAS home web site is located at

```
http://www.biodas.org/servers/
```

and contains, apart from the server itself, additional (authoritive) documentation. At one point or another, you will need to read that documentation as well.

1.1 What we will be doing

I will assume that you have machine running one or another UNIX-like operating system. Root access on the machine is not required, but is preferred as you might run into some security issues otherwise (I will point these out). The system that I used was a standard issue EBI office Red Hat Linux system, release 7.1. I’m most comfortable using command shells derived from the POSIX shell (sh, pdksh, ksh93, bash, ash, zsh, etc.) Users of tcsh and csh will need to translate a few lines, but the rest of you should be ok.

We will quickly go through the installation of the following things:

1. The Apache web server, version 1.3.17 or later. I’m using version 1.3.27.
2. MySQL, version 3.23 or later. I’m using version 3.23.55.
3. Perl, version 5.6.1 or later. At the time of writing, the latest stable version of Perl is 5.8.0. You will also need to install the following Perl bundles from CPAN:
    1. `Bundle::DBI`, version 1.20 or higher. I’m using version 1.32.

   2. `Bundle::DBD::mysql`, version 1.22 or higher. I’m using version 1.2219.
   3. `Bundle::BioPerl`, which is at version 1.2 at the time of writing. ...or at least the `Bio::DB::GFF` module version 0.38 or higher.

   Optionally, you might also want to install the `mod_perl` distribution, version 1.24 or higher, and the `Apache::DBI` module, version 0.88 or higher.

4. The Lightweight DAS server itself. I’m using version 1.09.

I’m assuming that you do not have root access, so you will not be able to write into e.g. `/usr/local` on your UNIX machine. You will need access to a directory that you have permission to write to. You will use this firectory as an “alternative root directory”. Replace any mentioning of `$ALTROOT` with your chosen directory (there’s nothing stopping you from using `/` as the `$ALTROOT` if you have root access). Also, make sure `$ALTROOT/bin` and `$ALTROOT/sbin` are first in you `$PATH` variable. You will eventually also need to cram `$ALTROOT/mysql/bin` in there, so you might as well add it now.

```
$ PATH=$ALTROOT/bin:$ALTROOT/sbin:$ALTROOT/mysql/bin:$PATH
$ export PATH
```

I have been using the GNU Stow program for handling software packages for a couple of years, and I also used it when getting LDAS going. The GNU Stow program, which is a Perl script that requires Perl version 5.005 or later, makes it a lot easier to manage software packages and effectively stops `/usr/local` (or `/opt` or whatever you preferred third party installation root directory might be) from getting messy. Get it from

```
http://www.gnu.org/directory/stow.html
```

or from one of the GNU FTP mirrors. If you’re using GNU Stow, install all programs into `$ALTROOT/stow/`*`packagename-version`* instead of straight into `$ALTROOT`, then stow the package as described in the GNU Stow documentation. Executable binaries and scripts will be symbolically linked from `$ALTROOT/bin`, so the inital `$PATH` won’t need to be changed from what I said above.

2. Installation

2.1 Install a recent version of Perl

This can be done in several ways depending on your UNIX. The by far easiest way if you already have a version of Perl installed is to go through the CPAN shell:

```
$ perl -MCPAN -e shell

cpan shell -- CPAN exploration and modules installation (v1.65)
ReadLine support enabled

cpan> install J/JH/JHI/perl-5.8.0.tar.gz

Beginning of configuration questions for perl5.

Checking echo to see how to suppress newlines...
...using -n.
The star should be here-->*

[...]
```

Installing Perl directly from the unpacked source tarball is quite easy as well (read the INSTALL document in tarball):

```
$ sh Configure

Beginning of configuration questions for perl5.

Checking echo to see how to suppress newlines...
...using -n.
The star should be here-->*

[...]

$ make
$ make test
$ make install
```

At the time of writing, the source tarball for Perl 5.8.0 can be fetched from

```
http://cpan.org/src/README.html
```

The important question during the configuration is “Installation prefix to use?”. Answer with your choise of `$ALTROOT`. Also, you probably want to answer `n` to the question “Do you want to install perl as /usr/bin/perl”.

I had to bypass the testing stage of the installation since one of the tests kept failing. Just add `force` in front of `install` if you’re using the CPAN shell, or just ignore the `make test` step if you have problems with this. It seems to be working fine anyways.

2.2 Install the Apache web server

Depending on whether or not you decide to install the `mod_perl` distribution, the installation of the Apache web server will differ. I will describe how to install the server with `mod_perl` disabled. People wanting to install `mod_perl` should look in the `mod_perl` documentation for installation instructions.

The Apache web server source tarball may be found at

```
http://httpd.apache.org/download.cgi
```

and once it has been unpacked, the configuration, building and installation of the package is, at least for version 1.3.27, a simple matter of saying

```
$ ./configure --prefix=$ALTROOT --with-layout=GNU
$ make
$ make install
```

This should install the executables in `$ALTROOT/bin`, the document root of the server is set to `$ALTROOT/share/htdocs`, the CGI scripts should be located in `$ALTROOT/share/cgi-bin`, the server error and access logs goes into `$ALTROOT/var/log`, and the configutration files goes into `$ALTROOT/etc`. Edit `$ALTROOT/etc/httpd.conf` to suit your setup. The default port that the server will listen to will be set to 8080.

Users of GNU Stow might want to change the document root and the CGI directory (and might then also want to allow symbolic links to be followed in the CGI directory). For a personal server like this, you should be extra careful to only allow access from the hosts that really need to access the server, either by properly configuring a firewall, or by configuring the Apache server itself.

I had to remove a number of plus signs from GNU layout section of the file `config.layout` before running the `configure` script. I also changed the prefix setting in the same section to match the argument given to the `configure` script on the command line.

Pointing a web browser to

http://localhost:8080/

should show the standard Apache boiler plate page (a page entitled “Seeing this instead of the website you expected?”). This means that the web server is functional.

2.3 Install MySQL

Because of a bug in version 2.96 of the GNU C compiler on my system I couldn’t install MySQL from source, so I got the binary distribution tarball instead. Fetch MySQL from the download page at

```
http://www.mysql.com/downloads/
```

Install it as e.g. `$ALTROOT/mysql`. For instructions, see e.g. the `INSTALL-BINARY` document in the tarball if you’re installing a binary distribution of MySQL.

When done, add `$ALTROOT/mysql/bin` to your `$PATH` environment variable (if you haven’t done so already) and start a MySQL deamon:

```
$ PATH=$ALTROOT/mysql/bin:$PATH; export PATH
$ cd $ALTROOT/mysql
$ ./bin/safe_mysqld &
```

You will also have to change the MySQL root user password:

```
$ ./bin/mysqladmin -u root password 'new-password'
$ ./bin/mysqladmin -u root -h uhuru.ebi.ac.uk password 'new-password'
```

Replace “new-password” with the chosen password, and replace `uhuru.ebi.ac.uk` with the name of your machine.

2.4 Install the Perl modules

We do this after installing and starting MySQL since one of the test steps includes connecting to a test database.

It’s easy to use the CPAN shell to install the needed modules (you might need to reconfigure your old CPAN configuration by deleting `~/.cpan/CPAN/MyConfig.pm`):

```
$ perl -MCPAN -e shell
[...]
cpan> install Bundle::DBI
cpan> install Bundle::DBD::mysql
cpan> install Bundle::BioPerl
cpan> exit
```

I had to force install some of the above bundles since one or two test failed. Just add `force` before the `install` command. Also, the `SOAP::Lite` module (needed by `Bundle::BioPerl`), version 0.55, seems to suffer from file permission problems and needs to get manual help to be installed:

```
[...]
inflating: SOAP-Lite-0.55/t/TEST.pl
Couldn’t rename SOAP-Lite-0.55 to [...]/SOAP-Lite-0.55:
Permission denied at [...]/lib/perl5/5.8.0/CPAN.pm line 3903

cpan> exit
$ cd ~/.cpan/build/tmp
$ chmod -R u+w SOAP-Lite-0.55
$ cd SOAP-Lite-0.55
$ perl Makefile.PL
$ make
$ make install
$ cd
$ perl -MCPAN -e shell
cpan> install Bundle::BioPerl
[...]
```

Of course, you may choose to only install the `Bio::DB::GFF` module instead of installing the whole BioPerl bundle (wich also requires the Expat XML parser and the GD graphics library).

The only additional step that needs to be taken is to ensure that the scripts in the `scripts/Bio-DB-GFF` directory of the BioPerl distribution is available in your `$PATH`, maybe by copying them to `$ALTROOT/bin`:

```
$ cp ~/.cpan/build/bioperl-1.2/scripts/Bio-DB-GFF/*.pl $ALTROOT/bin
```

2.5 Install the Lightweight DAS server

Get the LDAS server from

```
http://www.biodas.org/download/ldas/
```

and unpack the tarball. Run Perl on the supplied `Makefile.PL` script and enter the correct paths to the Apache configuration directory and the CGI directory. Then make and install:

```
$ perl Makefile.PL
[...]
$ make
$ make install
```

Edit the LDAS CGI script and specify the location of the `das.conf` dicretory. In my case, I had to change the line

```
$CONF_DIR = ’/usr/local/apache/conf/das.conf’;
```

of `$ALTROOT/share/cgi-bin/das` into

```
$CONF_DIR = ’/scratch/altroot/etc/das.conf’;
```

(`/scratch/altroot` happens to be the value of `$ALTROOT` that I used).

Pointing a browser to

```
http://localhost:8080/cgi-bin/das
```

should now give you a page just containing the words “invalid request” (this is an error message from LDAS, but at the moment it tells us that LDAS alive and well).

3. Adding sample data to the LDAS server

Just to show how to add a simple DAS track to the Ensembl genome browser, let’s use Ensembl to extract some data. Then we’ll add that data to our personal MySQL database and make it available in Ensembl.

For more in-depth documentation about how to properly set up the database, please refer to the “SETTING UP THE DATABASE” section of the LDAS documentation at

```
http://www.biodas.org/servers/LDAS.html
```

3.1 Get the data

DAS servers are of two kinds, reference servers and annotation servers. We’re not setting up a reference DAS server so we won’t need any assembly information. We still need to fetch reference and annotation data though.

Using the Ensembl MartView datamining tool at

```
http://www.ensembl.org/Homo_sapiens/martview
```

do the following selections:

Page	Select
START	Homo sapiens
	Ensembl Genes
REGION	Chromosome 7
FILTER	Known Genes only
	Transmembrane Domains only
OUTPUT	Output type, Structure
	Output format, GTF

Then just press “export” and save the output as e.g. `human_tm_7.gff` on you UNIX account. This file will be turned into our annotation file.

The GFF data in `human_tm_7.gff` needs to be reorganised for the LDAS loading scripts to understand it. The following command sequence does that and saves the result to a file called `human_tm_7.das`:

```
$ awk ’BEGIN { OFS="\t"; print "[annotations]"; }
    {
        print "Gene", $10, $3, $2, $1, $4, $5, $7, $6, $8;
    }’ human_tm_7.gff | tr -d ’;’ >human_tm_7.das
```

The reference file be a tab delimited text file, call it something like `human_tm_7_ref.das`, looking like this:

```
[references]
#id   class       length
7     Chromosome  157432793
```

The length of chromosome 7 my be found at

```
http://www.ensembl.org/Homo_sapiens/mapview?chr=7
```

or by executing the following SQL query (will get the lengths of all chromosomes):

```
$ mysql --host kaka.sanger.ac.uk --user anonymous -e \
    ’SELECT chromosome_id,name,length
     FROM chromosome
     WHERE chromosome_id < 25
     ORDER BY chromosome_id’ homo_sapiens_core_10_30
```

The file name extension on the reference and annotation files must be `*.das`, or the load scripts will be confused. The two files may also be concatenated to form a single load file. If you do this, make sure that the `[references]` section comes before the `[annotations]` section.

3.2 Prepare the database

We need to set up a MySQL database that will hold the data. Make sure there is a MySQL deamon alive on your machine and then...

```
$ mysql --user root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 7 to server version: 3.23.55

Type ’help;’ or ’\h’ for help. Type ’\c’ to clear the buffer.

mysql> CREATE DATABASE human;
Query OK, 1 row affected (0.00 sec)

mysql> GRANT ALL PRIVILEGES ON human.* TO ak@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT FILE ON *.* TO ak@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> GRANT SELECT ON human.* to ak@localhost;
Query OK, 0 rows affected (0.00 sec)

mysql> QUIT;
Bye
```

The first `GRANT` command in this example show all privileges (select, update, create, delete) being granted to users who log in as `ak` (me) from the local machine. You will want to change the user name to your own login name.

The second `GRANT` command grants file permissions to this user so that he can use the bulk loader. Because of the way MySQL’s bulk loading works, the file permission must be granted to all databases (`*.*`) and not just to a single one.

The third `GRANT` command grants `SELECT` permissions to the user running the Apache web server. This enables the web server script to read the `human` database, but not to update or otherwise change it. Note that we in this particular example have *a potential security issue here* since the user allowed to make changes to the database happens to be the same as the user running the web server... On a production or publicly available system, you should really be using two separate users for running the web and MySQL servers.

3.3 Load the data into the database

The LDAS server comes with a Perl script that makes the loading of the data into the database very easy. Just specify what files you want to have loaded and into what database you want to load them:

```
$ ldas_load.pl --create --database human \
    human_tm_7_ref.das human_tm_7.das
human_tm_7_ref.das: loading...
[...]
human_tm_7.das: 4352 records loaded
```

3.4 Configure the server with the new data

The LDAS serve needs to be able to find a configuration file in the `$ALTROOT/etc/das.conf` directory that tells it that there now is data in the database. This configuration file could be called `human.conf` and look like this:

```
[DATA SOURCE]
description = Human Chomosome 7 (test)
adaptor     = dbi::mysqlopt
database    = dbi:mysql:database=human;host=localhost
mapmaster   = http://your.host.name.here:8080/cgi-bin/das/human

[CATEGORIES]
default     = structural

[LINKS]

[COMPONENTS]

[FILTER]
```

This is a very minimalistic configuration file for test purposes and you should replace `your.host.name.here:8080` with whatever the name of your host may be, followed by the correct port to access the Apache web server. Please refer to the LDAS documentation for a proper description of the format of the configuration file.

You should now be able to point your browser at

```
http://localhost:8080/cgi-bin/das/dsn
```

which will give you the data sources of the DAS server. In the reply, you should be able to find the lines

```
<DSN>
  <SOURCE id="human">human</SOURCE>
  <MAPMASTER>http://your.host.name.here:8080/cgi-bin/das/human</MAPMASTER>
  <DESCRIPTION>Human Chomosome 7 (test)</DESCRIPTION>
</DSN>
```

3.5 Adding the source to Ensembl

Finally, fire up the browser again and point it to the Ensembl web site showing the human chromosome 7 at

```
http://www.ensembl.org/Homo_sapiens/mapview?chr=7
```

Click anywhere on the map of the chromosome on the left and locate the “DAS Sources” menu on the page that appears. Choose “Manage sources...” from that menu and click the “Attach DAS” button. Enter

```
your.host.name.here:8080/cgi-bin/
```

and click the “Show sources” button. You should now be able to select the new source and click “Attach DAS” to attach it to the Ensembl browser. You will need to press “Refresh” in the genome browser window to refresh the view.

If you can’t see any features on your track, that’s probably because you’re not viewing an area on which there are any features. Refer to the `[annotations]` section of your load files to see where you might find a feature.

You’re now ready to add your own data to LDAS. Please refer to the LDAS configuration documentation at `http://www.biodas.org/servers/LDAS.html`.