

The ProServer DAS Server

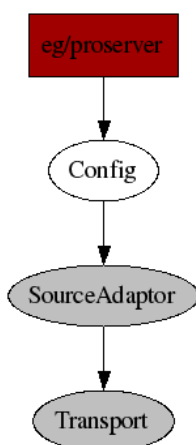
R. M. Pettett 2006-04-21

ProServer is a simple, extensible DAS server framework built around POE.

The architecture consists of:

- A server script which runs as a daemon
- A configuration module holding settings for sources
- A number of source adaptors, generally one for each back-end data source
- A number of back-end data transport helpers for handling database connections etc.

These fit together in the following way:



Grey modules denote virtual classes. SourceAdaptor subclasses implement methods for the DAS calls they provide, e.g. build_features, build_types, build_entry_points.

ProServer is not yet fully featured up to the DAS 1 specifications, so if there appears to be missing functionality, there probably is. However, ProServer in its current form is used successfully in production in many Sanger Institute projects.

Requirements

- Perl (tested on 5.6.1 and above)
- DBI for database connectivity
- Getopt::Long
- POSIX
- Sys::Hostname
- CGI
- POE
- Bundle::LWP
- Config::IniFiles
- Compress::Zlib

Before commencing with the installation of ProServer, please ensure you have a working Perl installation with the components listed above. ProServer has been successfully installed on Tru64, Linux, Mac OSX and Windows.

Downloading ProServer

A recent tarball of ProServer can be downloaded from its homepage:

<http://www.sanger.ac.uk/proserver/>

The very latest version (CVS HEAD) is here

```
cvs -d :pserver:cvsuser@cvs.sanger.ac.uk:/cvsroot/Bio-Das-ProServer login
```

When prompted enter the password CVSUSER, then:

```
cvs -d :pserver:cvsuser@cvs.sanger.ac.uk:/cvsroot/Bio-Das-ProServer checkout  
Bio-Das-ProServer
```

Installing ProServer on UNIX

```
cd Bio-Das-ProServer
```

edit eg/proserver.ini to (de)activate desired services

```
/usr/local/bin/perl Makefile.PL  
make test
```

Note there currently aren't m(any) tests.

```
eg/proserver -c proserver.ini
```

Installing ProServer on Windows

It is possible to install ProServer on Microsoft Windows using either the ActivePerl port of perl for Win32 <http://www.activestate.com/Products/ActivePerl/> or the Cygwin environment <http://www.cygwin.com/>

Installing on Cygwin

You may need to install the following Cygwin packages:

```
Devel/cvs  
Deve/gcc  
Devel/make  
Interpreters/perl  
Web/wget
```

Start a cygwin shell.

```
ln -s /usr/bin/perl /usr/local/bin/perl
```

Follow the UNIX install instructions

Writing Extensions for ProServer

In order to implement extensions for new data sources, classes should inherit and extend ProServer::SourceAdaptor functionality, namely the data retrieval methods which are applicable for the new data set. For a full list of supported methods please look at the SourceAdaptor.pm code. Basic SourceAdaptor examples include cosmic.pm and swissprot.pm.

SourceAdaptors retrieve data using a transport. Depending on the format of any new data set, it may be necessary to implement a new transport module. These already exist for some common sources – mysql, oracle, SRS getz and are simple command-line or socket-handling modules.

In order to run a server from your SourceAdaptor and data set, it needs to be configured. Edit lib/Bio/Das/ProServer/Config.pm and add the settings describing your adaptor. You may wish to turn other default, sources off.

Before starting the server it is necessary to build the Bio-Das-ProServer source tree. Change to the top-level and run

```
/usr/local/bin/perl Makefile.PL  
make
```

Once this is complete, change to the 'eg' directory and run ./proserver with your preferred arguments.

Example: “I need to present a hypothetical MySQL database with some data in.”

1. Download the ProServer sourcecode and ensure your system meets the installation requirements.
2. cd lib/Bio/Das/ProServer/SourceAdaptor/
3. Copy either `simplifiedb.pm` or `cosmic.pm` to `myadaptor.pm`
4. Edit `myadaptor.pm` – modify the package name to end `::myadaptor` and change the SQL to represent the database structure
5. cd ../../../../../../
6. cp eg/proserver.ini to eg/myconfig.ini
7. Edit eg/myconfig.ini
8. Copy the [image] section and change to [myadaptor].
9. Set adaptor = myadaptor
10. Add your database credentials to host, port, username, dbname, password
11. /usr/local/bin/perl Makefile.PL
12. make
13. eg/proserver -c eg/myconfig.ini
14. The service should bind all available network interfaces, or that specified in the interface field under [general].