

OpenACS Installation Guide

Roberto Mello (rmello@fslc.usu.edu) and the OpenACS Team

This is the Installation Guide for the ArsDigita Community System port for the PostgreSQL Relational Database Management System. This documentation is based on Philip Greenspun's ACS installation docs with parts of it (marked with [1]s) being quoted here.

This is the April 2001 revision of the documentation.

Table of Contents

1. [What is OpenACS](#)
2. [What you need to run OpenACS](#)
 - 2.1. [Installing Using Binary Packages \(RPM or .deb\)](#)
 - 2.2. [Installing from Source Code](#)
3. [Getting Ready to Install](#)
 - 3.1. [Getting Ready to Untar \[1\]](#)
 - 3.2. [Installing OpenACS from RPMs](#)
 - 3.3. [Installing OpenACS from Debian GNU/Linux .deb files](#)
4. [AOLserver/OpenNSD](#)
 - 4.1. [Installing AOLserver](#)
 - 4.2. [Configuring AOLserver](#)
 - 4.3. [Sample nsd.tcl file](#)
5. [PostgreSQL](#)
 - 5.1. [Which version: Postgresql 6.5.3 or 7 or 7.1?](#)
 - 5.2. [Installing PostgreSQL](#)
 - 5.3. [Some PostgreSQL tips from Don Baccus](#)
 - 5.4. [Notes on PostgreSQL 7.1](#)
6. [Configuring OpenACS itself](#)
 - 6.1. [What is where \(OpenACS directories and what is inside each of them\)](#)
 - 6.2. [Configuring OpenACS](#)
 - 6.3. [Configuring Permissions \[1\]](#)
 - 6.4. [Adding Yourself as a User and Making Yourself a Sysadmin \[1\]](#)
 - 6.5. [CAVEAT for those that want encrypted passwords in the DB](#)
 - 6.6. [Closing Down Access \[1\]](#)

6.7. [Making sure that it works \(and stays working\)](#)

6.8. [Ensure that your service automatically starts on boot \(or any other time the service dies\).](#)

7. [Everything works, now what ?](#)

7.1. [Backing up your PostgreSQL databases](#)

7.2. [If you are using the e-commerce module](#)

8. [Where do go I for help and how can I help ?](#)

8.1. [Why Not MySQL ?](#)

8.2. [Some Useful Links](#)

9. [Contributed Items](#)

10. [Acknowledgements](#)

[Next](#)

What is OpenACS

1. What is OpenACS

According to Philip Greenspun:

The ArsDigita Community System it is a toolkit of software that will help you build Web services with a collaborative dimension, ranging from knowledge management within companies to B2C ecommerce to product support and community among the customers. The software is free and open-source and has been tested in heavy use since 1995.

The software and underlying philosophy are documented in a full-length textbook for Web developers (also free and available online at <http://photo.net/wtr/thebook>) ([Using The ACS](#)). We highly recommend you to read this book. It will save you lots of trial-and-error, time, grief and will help you understand how to build a great web service. [Chapter 3](#) is of special interest, because that's where the data model and ideas of the ACS are explained, but the whole book is great.

If you are serious about building a web service, you should also read [ArsDigita Server Architecture](#), a way of building and delivering reliable web services.

The port for the PostgreSQL RDBMS was started and coordinated by Ben Adida (who also helped to write the original ACS) and Don Baccus, and is maintained as a community project.

The combination of GNU/Linux, AOLserver/OpenNSD, PostgreSQL and OpenACS enables you to build very sophisticated web sites with completely free software. The development and demo web site for OpenACS is <http://openacs.org>.

2. What you need to run OpenACS

There are now two main ways to install everything you need to run OpenACS. The first is to make use of packages designed to make installation easy. This is only useful if a set of packages exists for the particular operating system and OS distribution which you are using. The second way is the more traditional one, obtaining source code in tarballs and installing from those.

2.1. Installing Using Binary Packages (RPM or .deb)

There are binary packages available for [Debian GNU/Linux](#) and for [Red Hat Linux 6.2](#). RPM packages for Red Hat Linux 7.1 are expected to become available soon.

2.1.1. Debian GNU/Linux .deb packages

[Debian GNU/Linux](#) has OpenACS .deb packages all ready to install. Thanks to Brent Fulgham for providing the Debian packages.

2.1.2. Red Hat Linux 6.2 RPM packages

Very recently, a set of binary RPM packages have been created which make installing OpenACS on a Red Hat Linux 6.2 machine quick and simple. These are still being tested but look very promising. They are currently available at <http://www.xc.org/jonathan/openacs> but may soon become available at the [OpenACS software page](#). Thanks to Jonathan Marsden for providing these Red Hat Linux RPM packages.

2.1.3. RPMs using Apache and mod_aolserver from ArsDigita

ArsDigita is providing RPM packages for OpenACS with mod_aolserver on Apache. They are available at the [OpenACS software page](#). Note that installing these will require recompiling some of these RPMs from their source RPMs, though an install script simplifies this.

2.2. Installing from Source Code

- [AOLserver](#) - A free, Tcl-enabled, multi threaded, powerful web server that is behind huge sites such as aol.com and digitalcity.com. AOLserver/OpenNSD is also a complete web development platform being designed from start to have efficient access to databases. We support AOLserver/OpenNSD 3. NOTE: In february a fork of the AOLserver project was started to better address user contributions and requests. The project is called OpenNSD and its home page is at <http://www.opennsd.org>)

(Note: ArsDigita and Robert Thau came up with a `mod_aolserver` for the Apache web server, which intends to replicate the AOLserver API on Apache. It has performance issues when compared to AOLserver, but if you have to use Apache, it gives you a way to use OpenACS. It's available at the [OpenACS Software](#) page.

- [PostgreSQL](#) - A free, powerful SQL92-based Relational Database Management System with advanced features including Multi-variant Concurrency Control (unlike in the more common table-locking model, readers don't wait for writers and writers don't wait for readers, very crucial in a high-volume web server environment). Version 7 is the supported version.

- *Some flavor of UNIX or Windows.* Our development is almost entirely done on [GNU/Linux](#) (and when not done in GNU/Linux, it is tested on it) and for now, this is the only operating system we can help you with. Some very big web sites run on GNU/Linux and it has proven itself as a great platform for web servers, plus it is free software. [Red Hat Linux](#) and [Debian GNU/Linux](#) are the reference distributions for AOLserver and OpenACS, but if you know what you are doing, you can use any GNU/Linux distribution (and many people do).

To get OpenACS working on Windows you will have to get PostgreSQL working on it first, which is beyond the scope of this document. Please refer to the PostgreSQL documentation for that information. The OpenACS-specific instructions should need very little adaptation. Please contribute the documentation for Windows if you are going to do it.

- *The PostgreSQL driver* for AOLserver, available at [OpenACS software page](#)..

- The OpenACS distribution, available at [OpenACS software page](#)..

[Prev](#)

[Home](#)

[Next](#)

What is OpenACS

Getting Ready to Install

3. Getting Ready to Install

This section describes the filesystem location conventions of OpenACS, as well as how to initially unpack the files. These locations vary, depending on which installation approach is used.

Untarring is what those installing from source code will do; those installing from RPMs will do something using `rpm`, those installing from `.debs` will use `apt-get`.

3.1. Getting Ready to Untar [1]

We recommend rooting Web server content in `/web` (typically a symlink to a large mirrored disk drive). Since most servers these days are expected to run multiple services from multiple IP addresses, each server gets a subdirectory from `/web`. For example, `http://scorecard.org` would be rooted at `/web/scorecard` on one of our machines and if `http://jobdirect.com` were on the same box then it would be at `/web/jobdirect`.

For the sake of argument, we're going to assume that your service is called "yourdomain", is going to be at `http://yourdomain.com` and is rooted at `/web/yourdomain` in the Unix file system. Note that you'll find our definitions files starting out with "yourdomain.com".

- download `openacs-3.2?.tar.gz` into `/tmp/openacs-3.2?.tar.gz` (where ? is the version number)
- `cd /web`
- `tar xzvf /tmp/openacs-3.2?.tar.gz` (creates a directory "openacs-3.2?")
- `mv openacs-3.2? /web/yourdomain`

3.2. Installing OpenACS from RPMs

The use of `/web` is conventional in the ACS community, but conflicts with the Linux File Hierarchy Standard. When installing from binary RPMs, the service is installed under `/var/lib/aolserver/servers/` and the default OpenACS service is therefore installed at `/var/lib/aolserver/servers/defaultacs/` -- all this happens transparently as the RPMs are installed. With the appropriate OpenACS RPM file in the current directory, the single command

```
rpm -Uvh *.rpm
```

should be all that is required.

3.3. Installing OpenACS from Debian GNU/Linux .deb files

OpenACS Debian packages are included in Debian's unstable distribution (woody), so you'll need the appropriate entries in your `/etc/apt-sources.list` to get it. All you need to do is some apt-getting:

```
apt-get install openacs
```

and that's it. NOTE: At the time of writing, the OpenACS Debian packages are outdated.

[Prev](#)

[Home](#)

[Next](#)

What you need to run OpenACS

AOLserver/OpenNSD

4. AOLserver/OpenNSD

4.1. Installing AOLserver

The from source approach places AOLserver under `/home/aolserver`. The RPMs and DEBs use `/var/lib/aolserver`.

4.1.1. Installing AOLserver from source

We wrote a [Simple AOLserver Install Guide](#). For other instructions, see the [AOLserver docs](#).

It was verified that, as of June 2000, AOLserver has problems compiling with `pgcc` (used by GNU/Linux distributions such as Mandrake and Stampede). I could only get AOLserver 3 to compile once I got rid of `pgcc` in my Mandrake 6.1 box and loaded `egcs` (standard in Red Hat.)

You must create a user that AOLserver is going to run as, since it will not run as root. I usually use `nsadmin` or `ns` (for NaviServer, AOLserver's original name).

4.1.2. Installing AOLserver from RPMs

This is already taken care of by the `rpm` command mentioned in a previous section. Note that the `nsd.tcl` configuration file is installed into `/etc/aolserver/nsd.tcl`, and is adapted to work with no hand editing for most OpenACS sites, bringing up the web server on port 8000 by default.

4.1.3. Installing AOLserver from .deb files

This is taken care of by the `apt-get` command in a previous section.

4.2. Configuring AOLserver

Note: AOLserver 3 can use both `.ini` and `.tcl` files for initialization. We are dropping support for old `.ini` files because `.tcl` files are much more flexible. Those who are used to `.ini` files and want to use them will know what to do :-)

We are including a sample working `nsd.tcl` file as reference that you can use, but you must configure it for your own needs. Just moving it to your AOLserver directory and running AOLserver will not work. The file is well commented so you should be able to understand it. Look at the AOLserver docs for other parameters you could have.

Pay special attention to the following sections in the `nsd.tcl` file:

```
ns_section "ns/db/drivers"  
ns_section "ns/db/pools"  
ns_section "ns/server/${server}/tcl"  
ns_section "ns/server/${server}"
```

You can only have one ArsDigita Community System running from a single nsd process (though you can have as many ACS servers as you like on a physical machine; each just needs its own process). The reason for this is that each ACS installation needs to source its own parameter file (see second to last line in the nsd.tcl file).

In the ns/server/\${server} section, if you want to use our fancy custom error responses and such, uncomment the lines

```
ns_param NotFoundResponse "/global/file-not-found.html"  
ns_param ServerBusyResponse "/global/busy.html"  
ns_param ServerInternalErrorResponse "/global/error.html "  
ns_param ForbiddenResponse "/global/forbidden.html "  
ns_param UnauthorizedResponse "/global/unauthorized.html"
```

then go into the www/global/ directory and edit these files to suit. [1]

4.3. Sample nsd.tcl file

You can find well-commented, almost ready nsd.tcl file [right here](#). Rename it to nsd.tcl, chown it to your AOLserver user and edit to your needs. There are only a few things that need changes. Read the comments.

[Prev](#)

Getting Ready to Install

[Home](#)

[Next](#)

PostgreSQL

5. PostgreSQL

5.1. Which version: Postgresql 6.5.3 or 7 or 7.1?

PostgreSQL 7 is out at the time of this writing. It comes with some exciting new features such as referential integrity, `to_char`, optimizer overhaul and an enhanced `psql`.

The more advanced features of PG 7 are critical to OpenACS (such as `lztext` which will allow bigger comments, static pages, and referential integrity) so that's the version we support. Upgrading from PG 6.5 to PG 7 is not too hard, although you'll need to `pg_dump` and restore your database, plus an `initdb`.

Postgresql 7.1 is even more featureful and provides useful performance gains too. However it is new enough that few OpenACS installations use it yet. OpenACS 3.2.4 had a few SQL queries which do not work under Postgresql 7.1, these are fixed in OpenACS 3.2.5.

5.2. Installing PostgreSQL

5.2.1. Installing Postgresql from Source

5.2.1.1. Installing Postgresql itself from Source

I wrote a bare-bones installation guide for PostgreSQL, see Simple PostgreSQL Install Guide ([simple-pg-install.html](http://www.postgresql.org/docs/admin/install855.htm)).

For more generic downloading and compiling instructions see <http://www.postgresql.org/docs/admin/install855.htm>.

5.2.1.2. Compiling the PostgreSQL driver for AOLserver

Note: The `nspostgres.so` driver included in the AOLserver distribution *will not work* with OpenACS. Please use the driver available at openacs.org/software.adp. You will need the AOLserver source distribution to compile the driver. We plan to offer binaries at some point.

Also, to do any compiling in C, you'll need a compiler and the right libraries installed in your system. AOLserver, PostgreSQL and the PG driver were tested with `gcc` (the GNU Compiler Collection) and `gmake` (GNU Make). You will need `gcc` (egcs in Red Hat, `pgcc` in Mandrake/Stampede) and the `glibc` (GNU C library) installed (for GNU/Linux distributions this usually means packages like `glibc` and `glibc-devel`).

The PostgreSQL driver now comes with the AOLserver distribution (`nspostgres.so`), but if you are having problems, you can always get the latest PostgreSQL driver from [OpenACS software page](#) and compile it. If you are using the driver that comes with AOLserver, skip the next step.

Edit the Makefile to include the correct path to your PostgreSQL and AOLserver directories. If you are using the RPM version of PostgreSQL, make sure you have the `devel` package installed as well. You need to pay attention to these lines:

```
PGLIB=/usr/local/pgsql/lib # Where your PG libraries are installed
PGINC=/usr/local/pgsql/include # Where your PG includes are installed
NSHOME=/home/aolserver # Where your AOLserver is installed
NSINC=/usr/local/src/aolserver3_0/include # Where you untarred AOLserver
```

Do a `make` and then `make install`. The file `postgres.so` will be copied to the AOLserver's `bin` directory.

If you are running PG 7, make a symbolic link from `libpq.so.2.0` pointing to `libpq.so.2.1` because AOLserver looks for

libpq.so.2.0 when loading the driver:

```
cd /usr/local/pgsql/lib
ln -s libpq.so.2.1 libpq.so.2.0 (as user postgres)
```

5.2.1.3. Loading the data model

Make sure PostgreSQL is running fine with all the environment variables set (the RPM version does that all for you).

- Login as `postgres` (the PostgreSQL super user) and create a user for AOLserver in PostgreSQL. If your AOLserver runs as `nsadmin`, that should be the user to create with the command `createuser nsadmin`. In PG 6.5, you will be asked if the user is a super user and allowed to create databases, respond YES (y) to both. In PG 7 it will ask you if this user is allowed to create databases and if this user is allowed to create new users, respond YES (y) to both as well.

From now on, become the user AOLserver will connect to PostgreSQL as (e.g. `nsadmin`).

- Come up with a name for your Database (Usually it will be the name of the web service you're setting up. I'll use `yourdb` as example). Then create the database with the command: `createdb yourdb`.

- cd to the `www/install` directory of the OpenACS distribution and load the country/state/zip codes with the command :

```
./load-geo-tables yourdb
```

- cd to the `www/doc/sql` directory. If you are running the RPM version of PostgreSQL, edit the file `postgres.sql` and uncomment the following lines, commenting the two similar lines right below them:

```
--create function plpgsql_call_handler() RETURNS opaque
--as '/usr/lib/pgsql/plpgsql.so' language 'c';
```

- Edit the file `load-data-model.sql`. Uncomment the line `\i postgres65.sql` only if you are running PG 6.5.x.

(Optional - Deprecated) If you are running PG 6.5.3 and have the Tcl package loaded (or compiled `--with-tcl`) you may comment the `\i postgres65.sql` line and uncomment the `\i postgres-pgtcl.sql` line.

- Load the data model into `yourdb` with the command:

```
psql -f load-data-model.sql yourdb
```

Alternatively and for debugging purposes you can do the following (NOTE: The PDF output for some reason, turns the "greater than" sign that comes after 2, to "62". It should read `2> datamodel.txt`):

```
psql -f load-data-model.sql yourdb 2> datamodel.txt
```

to save PG's output to a file called `datamodel.txt`, which you can review and look for errors. If you have a bunch of `ERROR` messages in this file, then you forgot to configure one of the OpenACS files.

If anything goes wrong, it is easier to simply destroy the db (command `dropdb yourdb`) and recreate it after you've reviewed your steps.

- Do a `psql yourdb`. You should end up with a prompt after a couple of messages indicating that it has successfully connected with the database. Do a `\d` to see all the tables in your db. Once you're certain you can connect from the account via `psql`, you should have no problem connecting via AOLserver. [DRB]

5.2.2. Installing PostgreSQL from .deb files

The `apt-get` command mentioned earlier should install all the packages you need. You'll need the `postgresql` and `postgresql-pl` packages and if you'd like bboard searches to work, you'll also need the `postgresql-tcl`.

5.2.3. Installing PostgreSQL from Red Hat RPMs

If you want to use RPMs (the suggested approach on Red Hat machines), please use the official PostgreSQL RPMs. They are available at [Postgres' website](#). Lamar Owen maintains those RPMs and is also a member of the OpenACS team, so we know exactly how those packages are done, and our tests and documentation are based on that.

Those installing the `aolserver` binary RPM should *also* install the companion `aolserver-postgresql` RPM instead of compiling the driver from source code. Be consistent with this -- either install both from RPMs, or both from source code.

5.3. Some PostgreSQL tips from Don Baccus

Note: These are not absolutely necessary for running OpenACS, but I included here because it tells you how to get more out of PostgreSQL. It is good reading especially if you want to do something serious with your copy of OpenACS.

You'll need to make sure the Postgres postmaster is running, first. This is the process that forks a Postgres backend when AOLserver (or any other application, including PSQL) initiates a backend connection. I've got my `.ini` file configured so idle connections never get released by AOLserver, so this forking happens only once per connection per lifetime of the server (the `MaxOpen` and `MaxIdle` in the pools section of the `nsd.ini`, as in the example above).

Here's the command I use to run Postmaster from my `/etc/rc.d/init.d/postgresql` script:

```
su -l postgres -c '/usr/local/pgsql/bin/postmaster -B 1000 -o "-S 2000" -S -D /usr/local/pgsql/data'
```

For the RPM version should be something like this:

```
su -l postgres -c '/usr/bin/postmaster -B 1000 -o "-S 2000" -S -D /var/lib/pgsql/data'
```

Jonathan Marsden suggests that for the RPM version of PostgreSQL 7.1, it should be instead something more like this:

```
su -l postgres -c "LC_ALL=C /usr/bin/pg_ctl -D $PGDATA -p /usr/bin/postmaster -o '-B 2000 -o -S2000' start >/dev/null 2>&1" < /dev/null
```

Some explanations - "`-B 1000`" tells it to allocate 1000 blocks of shared memory (rather than the default 64, which is way puny). I've compiled my copy of postgres with a 16K blocksize, so this is 16MB of shared memory space, i.e. the most postgres will use without a kernel recompile. If you've compiled with the default 8K blocksize (RPM version), "`-B 2000`" will work. You needn't do this for testing, but for an active system helps a lot.

The '`-o "-S 2000"`' tells each backend to use up to 2 MB (2000 x 1KB) of RAM for sorting, etc before spilling to disk.

The other "`-S`" (to the postmaster itself, don't confuse with the above where `-o` is used to pass flags to forked backends) tells it to run "silently", in the background.

`-D` is used to pass the path to the database which you've hopefully already run `initdb` on, etc.

5.3.1. How to increase the blocksize in PostgreSQL

Again, this is not required to run OpenACS.

By default PostgreSQL is compiled with a blocksize of 8 Kb. You can compile PostgreSQL to have 16 Kb blocksize instead,

which will allow for bigger text and lztext data types. (NOTE: This is completely unnecessary in PostgreSQL 7.1, as it will be rid of this limitation)

Refer to the [Simple PostgreSQL Installation Guide](#) for instructions on how to do this for versions of Postgresql before 7.1.

5.4. Notes on PostgreSQL 7.1

The long-awaited 7.1 version of PostgreSQL, as usual, brings some exciting features and improvements like OUTER JOINS, optimizer improvements, fixes in PL/PgSQL, etc. The purpose of this section is to let you know what OpenACS 3.2.x items will behave differently under PG 7.1. This is what we know so far, if you find something else, please let us know.

- In *bookmarks.sql* you can comment the lines where the *chr(int4)* function is declared, for PG 7.1 comes with an identical one (contributed by Krzysztof Kowalczyk).

[Prev](#)

AOLserver/OpenNSD

[Home](#)

[Next](#)

Configuring OpenACS itself

6. Configuring OpenACS itself

Although installing AOLserver and OpenACS in different ways will put its files under a different directory (`/home/aolserver/servers/yourdomain/` or `/var/lib/aolserver/servers/defaultacs/`, for example), the directory structure within OpenACS remains the same, no matter what method was used to install OpenACS.

6.1. What is where (OpenACS directories and what is inside each of them)

- `/bin` - executables used by the toolkit (e.g. WatchDog)
- `/parameters` - OpenACS configuration file(s)
- `/tcl` - OpenACS Tcl scripts library. Your AOLserver config needs to point to here as its Tcl library. The definitions for the modules are here.
- `/templates` - where templates are stored. Templates modify how a page is displayed according the user's preference (e.g. text x graphics) and language (e.g. with language extensions `.fr`, `.pt`, etc.) Not all modules are template-enabled yet, but expect that to change in ACS 4.
- `/users` - user specific files. Used by the home page module for example.
- `/www` - Where all the pages live. Each module has a subdirectory here (e.g. `www/bboard`)
- The `www/register` directory contains the login and registration scripts. You can easily redirect someone to `/register/index.tcl` to have them login or register. [1]
- The `www/pvt` directory is for user-specific pages. They can only be accessed by people who have logged in. [1]

6.2. Configuring OpenACS

Note: Starting with version 3.2.5, the supplied default configuration file `ad.tcl` has been made generic. It may work fine completely unchanged for many installations, and in any event should now need less editing than in previous versions.

Also, RPM users wishing to further customize their OpenACS installation should edit `defaultacs.tcl` rather than renaming `ad.tcl`.

- In the `parameters` directory of the OpenACS tree, rename the `ad.tcl` file to the name of the virtual server you are running in AOLserver (*server1.tcl* for example)
- Each module of the ACS is configured in the sections of this file, with a heading such as `ns/${server}/acs` (if you are using the included `ad.tcl` file). If you are using `.ini` file, than the headers will look like `[ns/server/yourservername/acs]`. In this case, replace all occurrences of `yourservername` with the actual name of the virtual server configured in AOLserver (such as `photonet`, or `server1`).
- Edit the `parameters` to fit your needs, otherwise your website will show `Yourdomain Network` and `webmaster@yourdomain.com` all over. If you want to change how some of these are used, a good place to look is

/web/yourdomain/tcl/ad-defs.tcl. [1] There are lots of comments in the file to help you out and the documentation of each individual module can be found at <http://openacs.org/doc>.

- [READ THE CAVEAT](#) in section 6.5 if you choose to save encrypted password in the db (EncryptPasswordsInDBP=0 or ns_param EncryptPasswordsInDBP "0" in nsd.ini or nsd.tcl respectively).

6.3. Configuring Permissions [1]

You need to protect the proper administration directories of the ACS. You decide the policy. Here are the directories to consider protecting:

- /doc (or at least /doc/sql/ since some AOLserver configurations will allow a user to execute SQL files)
- /admin (this directory is already protected in latter OpenACS releases).
- any private admin dirs for a module you might have written that are not underneath the /admin directory

6.4. Adding Yourself as a User and Making Yourself a Sysadmin [1]

The ArsDigita Community System will define two users: system and anonymous. It will also define a user group of system administrators.

After starting AOLserver, you'll want to:

- add yourself as a user to the system, at <http://yourservername.com/register/>
- add yourself as as member of the site-wide administration group.

To do this, log out as yourself (there's a link at [Your Workspace http://yourservername.com/pvt/home.tcl](http://yourservername.com/pvt/home.tcl)) and then log in as the system user (email of "system"). Change the system user's password (the default is `changeme`). Visit the the User Groups Admin pages at <http://yourservername.com/admin/ug/> and add your personal user as a site-wide administrator.

Now you're bootstrapped!

6.5. CAVEAT for those that want encrypted passwords in the DB

If you want to save encrypted passwords in the database, you'll have to do some things manually to get ACS working because the default users `system` and `anonymous` come with plain text passwords.

This is what you need to do:

1. create a login for you (as described above). Your password will be saved encrypted in the database. Go into psql and do a `select user_id,first_names,password from users;` to see all the users in your database.
2. Next, change the system user password for the password of the user you just created. Let's say the encrypted password for your user was something like `0xabcdef` (or whatever), then do a `update users set password='0xabcdef' where user_id=1;`
3. Now go back to your browser, and logout as your user (from <http://yourservername.com/pvt/home.tcl>), login as system with the same password you used for your user, add your user to the administration group as described above, and then change the system and anonymous passwords (from <http://yourservername.com/admin/users>).

6.6. Closing Down Access [1]

The ACS ships with a user named "anonymous" (email "anonymous") to serve as a content owner. If you're operating a restricted-access site, make sure to change the anonymous user's password (the default is `changeme`).

6.7. Making sure that it works (and stays working)

Run the acceptance tests in <http://photo.net/doc/acceptance-test.html>.

Note: The first part of the above page is aimed at the original version of ACS for Oracle. You can replace that first part by going to `psql` (PostgreSQL interactive SQL tool) and doing some tests:

```
$ su - youraolserveruser
$ psql yourdb
yourdb# \d
yourdb# select * from users;
```

The first `psql` command is going to list all your tables (under PG 6.5) or all your relationships (under PG 7) and the second will show all the records in the `users` table.

The other sections of the acceptance-test can be used either under the Oracle or the PostgreSQL versions of the ACS.

6.8. Ensure that your service automatically starts on boot (or any other time the service dies).

Note: The binary RPMs of Postgresql and AOLserver use the Red Hat init script system. So users of these RPMs can simply do

```
/sbin/chkconfig postgresql on
/sbin/chkconfig aolserver on
```

to ensure that these services are restarted at boot time.

This section was taken from *The Hitchhiker's Guide to the ACS*, written by the ArsDigita folks.

This step should be completed as root. This can break every service on your machine, so proceed with caution.

- Copy this [restart-aolserver](#) into `/tmp/restart-aolserver.txt`
- This script needs to be SUID-root, which means that the script will run as root. This is necessary to ensure that the aolserver processes are killed regardless of who owns them. However the script should be in the *web* group to ensure that the users updating the web page can use the script, but that general system users cannot run the script. You also need to have Perl installed and also a symbolic link to it in `/usr/local/bin`.

```
$ su - ; Enter root password.
# cp /tmp/restart-aolserver.txt /usr/local/bin/restart-aolserver # chown root.web
/usr/local/bin/restart-aolserver
# chmod 4750 /usr/local/bin/restart-aolserver
# ln -s /usr/bin/perl /usr/local/bin/perl
# su - nsadmin
```

- Test the *restart-aolserver* script by making sure all servers are dead, starting a new server, and then killing it. You should see the following lines. nsadmin and typing

```
$ killall -9 nsd
nsd: no process killed
$ /home/aolserver/bin/nsd -u nsadmin -g web -t /home/aolserver/service_name.tcl
$ restart-aolserver service_name
Killing 23727 23728 23729 23730
$ killall -9 nsd nsd: no process killed
```

The numbers indicate the process ids (PIDs) of the processes being killed. It is important that no processes are killed by the second call to killall. If there are processes being killed, it means that the script is not working.

- Assuming that the restart-aolserver script worked, login as root and open /etc/inittab for editing.

```
$ su - ; Enter root password
# emacs -nw /etc/inittab
```

- Copy this line into the bottom of the file as a template, making sure that the first field nss is unique.

```
nss:2345:respawn:/home/aolserver/bin/nsd -u nsadmin -g web -i -t
/home/aolserver/service_name.tcl
```

- *Important:* Make sure there is a newline at the end of the file. If there is not a newline at the end of the file, the system may suffer catastrophic failures.
- Still as root, enter the following command to re-initialize /etc/inittab.

```
# killall -9 nsd
# /sbin/init q
```

- *Important:* See if it worked by running the restart-aolserver script again.

```
# restart-aolserver service_name Killing 23750 23753 23754 23756
```

If the processes were killed, congratulations, your server is now automated for startup and shutdown.

7. Everything works, now what ?

- Add your site to the OpenACS Sites list: <http://www.openacs.org/sites.html>
- Read the [OpenACS Getting Started Guide](#).
- Read the [Installing Monitors](#) section of The Hitchhiker's Guide to the ACS. It will teach you how to install programs that will monitor your web server and take actions in case something goes wrong. It is written for ACS/Oracle, but it can be easily adapted to OpenACS (except for Cassandra). We intend to port those documents to OpenACS in the near future.

7.1. Backing up your PostgreSQL databases

See *Backup Tips from Don Baccus*, in the *OpenACS Getting Started Guide*.

7.2. If you are using the e-commerce module

You should be aware that the e-commerce module makes use of the *ImageMagick* suite of graphics manipulation programs to generate picture thumbnails, specifically the *convert* program. Unfortunately the path to this program is hard-coded into the OpenACS code and this path may not match where your copy of *convert* is installed.

To find out where your *convert* is and make a symlink from it to where OpenACS thinks it is installed, do the following:

```
# which convert
/usr/X11R6/bin/convert
ln -s /usr/X11R6/bin/convert /usr/local/bin/convert
```

8. Where do go I for help and how can I help ?

- <http://openacs.org/bboard> - the OpenACS installation and configuration bulletin board.
- <http://openacs.org/sdm> - all bug reports and feature requests should go here (including corrections/suggestions to this documentation).
- If you want to help with the OpenACS project, email ben@mit.edu
- <http://yourservername.com/doc>. Once you have your OpenACS system setup, the /doc directory contains documentation about all modules and many other aspects of the toolkit.

8.1. Why Not MySQL ?

We get this question many times. Ben Adida wrote a short article explaining why OpenACS doesn't use MySQL (which made into Slashdot, giving openacs.org 37,000+ hits on that day). Read it at <http://www.openacs.org/why-not-mysql.html>.

You can read more about this on the article that explains why Sourceforge.net has moved away from MySQL, at <http://www.phpbuilder.com/columns/tim20001112.php3>.

8.2. Some Useful Links

- aolserver.com and especially aolserver.com/doc for AOLserver info and documentation (also included in the source distribution).
- [ArsDigita's AOLserver Presentation](#)
- [ArsDigita Systems Journal](#) for lots of good quality information on AOLserver, the ACS and building great web services.
- [OpenForce Inc](#), provides commercial support, customization and other services using OpenACS.
- [PostgreSQL's TechDocs](#), has lots of useful extra documents, tips and tricks for PostgreSQL.

9. Contributed Items

Several user have contributed to the OpenACS project. These are the items that our community has found useful and requested to be included in the documentation:

- [Backup Tips from Don Baccus](#)- Tips for optimizing and backing up your PostgreSQL databases, including a script that can be used from within OpenACS.
- [Red Hat Updater](#)- Michael Cleverly has written this Tcl script to keep your Red Hat box up-to-date.

[Prev](#)

10. Acknowledgements

Several people have contributed to this document in either content or error reporting and I would like to thank them. Please let me know if I forgot to include your name.

Contributors (in no specific order):

Philip Greenspun, Ben Adida, Don Baccus, Michael Cleverly, Janne Blonqvist, Jonathan Ellis, Janine Sisk, Jade Rubick, Chris Hardy, Jonathan Marsden.

[Prev](#)[Home](#)[Contributed Items](#)