



Wacs Installation Guide

Third Edition
for WACS 0.8.3

B "Beaky" King
Published 23rd March 2009

Wacs Installation Guide

by B "Beaky" King

for WACS 0.8.3

Published 23rd March 2009

Copyright © 2006, 2007, 2008 B King

Abstract

WACS is a tool for building Adult Web Sites; it is equally suitable for managing a private collection or building a commercial web site. It has many best of breed features including dynamic filtering, model catalogs, automatic download and powerful search engine. It comes with a powerful API (application programming interface) implemented in both Perl and PHP5 languages to allow web developers to leverage it's facilities from their own programs.

This book describes the actions required to install the WACS System onto a suitable host system (typically a server). The intended audience is system administrators and prospective WACS site managers wishing to install WACS on a machine.

The WACS source code and other documentation and support tools can all be found at the WACS website at Sourceforge [<http://wacsip.sourceforge.net/>]. Commercial add-ons and support options can be purchased from Bevttec Communications Ltd, see their website at Bevttec Communications [<http://www.bevteccom.co.uk/>].

This work is licensed under the Creative Commons Attribution License. To view a copy of this license, visit <http://creativecommons.org/licenses/by/2.0/> or send a letter to Creative Commons, 559 Nathan Abbott Way, Stanford, California 94305, USA.

Table of Contents

1. Introduction to WACS Installation	1
WACS Overview	1
About This Book	1
Goals	1
About The Examples	1
2. Preparation For Installation	3
Preparation Tasks	3
Linux Operating Systems	3
3. Prerequisites	5
Web Server	5
Relational Database	5
Content Storage	5
4. Linux: Options	6
Methods Available	6
Linux Package Install	6
Linux easyinstall	6
Linux Manual Install	7
5. Installing WACS Using Packages	8
Package List	8
RPM Installation Steps	8
Downloading The RPMs	9
RPM Installation	9
Other System Issues	9
DEB Installation Steps	10
Downloading The DEBs	10
DEB Installation	11
RPM/DEB Initial Setup	17
Permissions Issues	17
1. Create the WACS database account (MySQL)	18
2. Create the necessary database schemas	19
3. Create default database contents (optional)	20
4. Import sample model records (optional)	21
5. Import sample set records (optional)	21
6. Installing With EasyInstall	23
Easyinstall: Download	23
Easyinstall: Running	23
7. Manual Installation	25
Caution	25
Manual Installation: Steps	25
8. Upgrading An Existing Installation	33
About Upgrading	33
The Upgrade Command	33
Additional Steps	33
9. Troubleshooting	35
Introduction	35
Troubleshooting Checklist	35
Special Notes About SELinux	36
Index	37

List of Tables

2.1. Software Pre-Requisites For WACS On Linux 4
5.1. List of Packages 8

Chapter 1. Introduction to WACS

Installation

WACS Overview

Welcome to WACS, Web-based Adult Content Server, a free software package for the management of material of an "Adult Nature" (or basically whatever euphemism for porn you prefer). It is web-based and can be used for the management of an existing collection, as a download manager, or as a back-end system for running a commercial adult web site. It is dramatically different from most other image gallery systems in that it understands photo sets and video clips as basic concepts, instead of single photographs. It also includes far more specialised tagging, source, relationship and attribute marking concepts than other more generalised systems.

WACS is extremely configurable, making extensive use of configuration files written in eXtensible Markup Language (XML). This book is one of a collection of manuals we have created to help you through the various aspects of using a system as complex as WACS. This guide is solely targeted to installation - additional guides exist for Users, Configuration and Programming.

About This Book

This book is an installation guide for WACS site managers or system administrators seeking to install the WACS environment on their systems. It does assume a certain amount of familiarity with the normal processes of installing software packages on your systems; the sections on manual installation in particular also assume a basic knowledge of using the Unix operating system (or any other future supported OS platform).

To get the best from this book, you should ideally be familiar with the basic user interface of the WACS applications themselves - the WACS User Guide would be an ideal primer for this and should introduce you to many of the concepts and tools being used here. There is also no substitute for using a real WACS site to get a general feel for how things work and are laid out.

Goals

The task of installing WACS onto a new server system consists of a number of distinct steps; these are:

- Preparing the host system
- Installing the pre-requisite software
- Installing the WACS applications and modules
- Getting a working configuration
- Installing some initial data

Some of these topics will be mentioned briefly here and will be covered in more depth in other guides in the WACS documentation set.

About The Examples

For copyright/licensing reasons, the example images feature sets from photoshoots by the main developer of WACS (Beaky) and a friend of his. These sets will be available on our demonstration site when that

goes live. Please understand that due to the bandwidth and storage costs in running such a server on the internet, and the need to verify (as best we can) that the applicant is an adult, there is a small charge for access to the site.

Chapter 2. Preparation For Installation

Preparation Tasks

Before we even start to install the WACS package, it is very important that we make sure the host candidate system is prepared for the task in hand. To do this, we need to ensure a number of things have been prepared beforehand:

- ensure adequate system resources
- assign and configure for static host name
- review security and access policies

The first of these steps, ensuring adequate system resources, basically involves looking at the sort of material you're intending to store in the WACS system and approximately what the storage requirements will be. If you are looking at holding sets for maybe fifty models who come from a site that specialises in high-resolution images and HD video clips, you may find that an average image set is upwards of 100MB, and an average video clip maybe 500-600MB. If each model has an average of four video vlips and 10 sets, then you're looking at probably 3GB per model, and would need to allocate around 150GB of storage, which with margins for future expansions means about 200-250GB to start off with.

Do remember that on most Linux systems you can use tools like the Logical Volume Manager (LVM) to ease the process of disc space allocation and in particular future expansion when live data is present. It is also perfectly possible to use Network Attached Storage (NAS) devices as the primary storage location for WACS collections.

You also need to make sure you assign a static IP address and hostname to the server system; more details on this and the use of NAS servers is given in the configuration guide. There are also a number of resources on the net to help you through this process; one that appears fairly complete is this one at [howtoforge.com](http://www.howtoforge.com/perfect-server-fedora9) [<http://www.howtoforge.com/perfect-server-fedora9>].



Warning

WACS is not currently compatible with the SELinux enhanced security system - this needs to be reduced to either **permissive** or switched off entirely (**disabled**) for WACS to work. This will affect Fedora and other RedHat-based distributions. It is our intention to resolve this issue by the 0.9.x release series of WACS.

If you're running Fedora (or any other distribution) with SELinux enabled, you will run into problems. WACS does not currently work well with SELinux and you have a choice of either setting it to permissive mode (where it logs problems but does not block things from working) or disabling it entirely. If you disable it entirely, it is much harder to go back to running it later as software updates and the like tend to get their SELinux attributes updated. On the other hand, permissive mode will fill up your log file areas and may slow down system operation somewhat.

Linux Operating Systems

If you are using either the RPM/DEB packages of the WACS applications, or the easyinstall script, and are using the default applications (MySQL in particular), the prerequisite applications will be automatically installed if they are not already present. If not, or you are using a different database (Oracle, or another

like PostGres SQL), you will need to install these applications first as detailed in the table below and then follow the manual install steps:

Table 2.1. Software Pre-Requisites For WACS On Linux

Service	Application	Version	Description
Web Server	Apache	> 2.0	main route of access
Database	MySQL	> 5.0	backend database engine
	Oracle	> 10g	alternative database engine
Perl	Language	> 5.8.0	Language interpreter (required)
Php	Language	> 5	Language interpreter (optional)
Perl::DBI	Library	any recent	Database interface library
Perl::DBD	Driver	for Database	Database driver routine for MySQL or Oracle
XML::Simple	Library	any recent	Parsers for eXtensible Markup Language (XML) files
Data::Dumper	Library	any recent	Essential debugging tool
File::Basename	Library	any recent	Filename manipulation routines
MIME::Base64	Library	any recent	Binary data encoder used with XML files

Chapter 3. Prerequisites

Web Server

WACS is primarily designed to work with the Apache 2 web server as this is the industry leading web server for Linux and Unix platforms. It's also available for the Mac OSX platform from various sources, and even for Microsoft Windows under the name WAMP Server. While other web servers may work fine, we would not recommend using them at this time and stage of WACS development. The RPM/DEB packages automatically drop in configuration file snippets to ensure correct operation of apache 2 on Linux releases with supported packages.

Relational Database

You do need to be aware that the MySQL network layer appears to be extremely sensitive to what the host is called. It needs to have a permanent, static name which is correctly mapped in the hosts or DNS so that *hostname* maps to *ip address* and the *ip address* maps back to THE SAME *hostname*. If this isn't the case, the final part of the installation - creating the database schemas and populating them - may well not work.

Other database engines may not be so sensitive - Oracle 10 and 11 versions are not (by default anyway).

Content Storage

The normal location for content storage is the home directory of the WACS user account which is created when you do either a package or an easyinstall. Obviously putting a large amount of multi-media material into the home directory area of the server may not be desirable so you may wish to consider where it should be placed. As mentioned elsewhere this could be a separate volume or group of volumes on an LVM partition, an external disc drive or even another remote server or NAS server supporting NFS protocols.

The important things are to ensure that the location of the storage is specified in the Wacs configuration file (usually `/etc/wacs.d/wacs.cfg`) ahead of adding the first content and that it remains accessible at the same file system location throughout. This means either arranging for it to be mounted, etc as necessary at boot time or deploying an automounter such as **autofs** or **am-utils** to do that for you.

Chapter 4. Linux: Options

Methods Available

With the Linux Operating System, there are three basic options available to you for installing WACS onto your system:

- Linux Package Installation
- Linux easyinstall script
- Linux Manual Installation

Each option above is progressively more complex than the previous one, but in the process affords more flexibility and configurability. The choice is yours.... On other platforms, you probably want to follow either the manual install procedure or play with the easyinstall script (but expect to have to change at least package names and installer names in the script).

Linux Package Install



Note

This feature was new in Wacs 0.8.1 and is currently available for Fedora 8, 9 and 10 based systems using the RPM package manager and for Ubuntu 8.04 and 8.10 systems using the DEB packaging system. It is our hope to extend the packaged software approach to include other platforms in a future release. It is likely that support for Fedora 8 will be dropped before the next release of Wacs (ie 0.8.4).

Where available for a given distribution and release, there are a number of WACS RPM or .deb packages you can make use of to install the WACS system. If you are using one of the more sophisticated package managers (yum, etc), you need only ask it to install the main wacs package and that tells the package manager what other components it needs to complete the install. This will bring in both the system packages needed - web server, database, perl libraries, etc - and the other parts of the WACS system needed for a working installation. If you are using one of the simpler package managers (rpm etc), it will complain about absence of the required packages until all the dependencies have been installed manually.

Since sourceforge.net doesn't yet seem to support YUM or debian repos properly you will have to download the requisite WACS packages manually in order for the install to proceed.



Warning

In order to conform to the the Fedora packaging guidelines, quite a few of the file locations are different on the packaged version of WACS, from that created by the easyinstall script or manual process. It shouldn't cause problems, but you do need to be aware of it, particularly if moving a configuration file between releases.

Linux easyinstall

The easyinstall script was our pre-packaging approach to installing WACS and can still be used on Fedora and Ubuntu distributions although now it is mostly useful for installing on older versions of Linux that are not supported directly by the packages we create. At present, pending the development of a web based configurator, easyinstall is a more complete solution and the resulting WACS installation is better able

to run "out of the box" with less configuration work still to do. However, it will prove harder to upgrade than the packaged solutions.

Linux Manual Install

This the only option available for any kind of unsupported operating system platform. The instructions later in this guide take you through all of the tasks needed step by step. This does assume some basic familiarity with command line operation of the Linux/Unix environment and a reasonable knowledge of the software installation policies of the operating system in question.

Chapter 5. Installing WACS Using Packages

Package List

In order to install WACS using the packages, you need to download a number of separate packages from sourceforge and have them available for your chosen package manager to find. Make sure you pick the right one for your Linux distribution. The list below details what these packages are:

Table 5.1. List of Packages

Name	Req'd	Description
wacs	Yes	The "Master" package which includes the others
wacs-core	Yes	The core files and user interface apps
wacs-tools	Yes	The collection management tools
wacs-download	Opt	The tools used for automatic download from subscription sites - optional
wacs-hostauth	Yes	Tool used to authenticate users (will not be needed if you have the commercial CRM package)
wacs-samples	Yes	Some sample data files in XML format and perl API programming examples
wacs-doc-pdf	No	Documentation in PDF format
wacs-doc-html	No	Documentation in HTML format (both single and multi page)

As you can see from the above list, you might wish to download the core packages, plus the download tools, plus whichever format of documentation you prefer to use. Unless you're already familiar with WACS, we'd strongly recommend using the master wrapper package (`wacs`) for the installation as it does a number of configuration steps for you. For the examples ahead, we'll assume PDF is the preferred format - your mileage may vary.

RPM Installation Steps



Note

For Ubuntu/Debian DEB package installs, please see the section called "DEB Installation Steps" instead.



Important

Before you start on an installation, please make sure that you have a statically allocated IP address, sensible hostname with a fully qualified domain name and that the machine is fully aware of these settings. For more information on these aspects, please consult the configuration guide. There is also a good guide to doing this at <http://www.howtoforge.com/perfect-server-fedora9> [<http://www.howtoforge.com/perfect-server-fedora9>]

Downloading The RPMs

The first step obviously is to download the appropriate packages for the operating system release, version and processor platform that you intend to run it on. Where a package contains *noarch* that means that it is suitable for any processor architecture running that distribution of Linux. Currently RPM packaged versions are available for Fedora 8 (labeled fc8), Fedora 9 (labeled fc9) and Fedora 10 (labeled fc10) and DEB packaged versions for Ubuntu 8.04 and 8.10. For more information on using the Ubuntu DEB packaged versions, please see the section called “DEB Installation Steps”

For an initial WACS installation (in this example for release 0.8.1 on an x86_64 machine running Fedora 8), you will probably want the following packages:

- **wacs-0.8.3-1.noarch.fc10.rpm**
- **wacs-core-0.8.3-1.noarch.fc10.rpm**
- **wacs-tools-0.8.3-1.noarch.fc10.rpm**
- **wacs-samples-0.8.3-1.noarch.fc10.rpm**
- **wacs-hostauth-0.8.3-1.x86_64.fc10.rpm**

If you plan on making use of the download toolset to connect to subscription sites for automatic downloads (although do be aware that only a very few sites are supported so far), you will also want to get the package called **wacs-download-0.8.3-1.noarch.fc10.rpm**. You may also wish to download one of the two versions of the documentation package: **wacs-doc-pdf-0.8.3-1.noarch.fc10.rpm** or **wacs-doc-html-0.8.3-1.noarch.fc10.rpm** - you can always access the same documentation direct from our sourceforge web site.

RPM Installation

Once you've downloaded the right packages, you need to gain the appropriate privileges and install the packages. There are any number of ways to do this, and you can pretty much use any of them; the example below uses the command-line based yum package manager:

```
# yum install --nogpgcheck wacs*.rpm
[...]
```

It is also possible to do this with the file manager, right clicking on each package file and choosing **Install Package**. The order on this is a bit tricky, but if you start with **wacs-core** and **wacs-hostauth**, then do the other packages and finally do the main wacs package, this should work out OK.

Other System Issues

Once the packages, and their dependencies, have been installed please confirm that both the Apache 2 Web Server (httpd) and the MySQL Database Server (mysqld) are enabled and running. In the GNOME desktop, the System -> Administration -> Services menu will take you to the Service Configuration screen where you need to both enable and start httpd and mysqld if these are not shown as currently running. If you prefer using the command line, the following steps will do the same task:

```
# /sbin/service httpd start
Starting httpd: [ OK ]
# /sbin/service mysqld start
Starting MySQL: [ OK ]
# /sbin/chkconfig --levels 345 httpd on
# /sbin/chkconfig --levels 345 mysqld on
#
```

The final system configuration step before starting work on getting WACS configured is to ensure that SELinux is running in a reduced mode that will not block the WACS components from working. This is only an issue on Fedora and other Red Hat based releases at present. We hope to have this resolved by the next release of WACS. You can determine the current mode of SELinux using the `sestatus` command:

```
% /usr/sbin/sestatus
SELinux status: disabled
%
```

To change the normal operational mode, you need to edit the file called `/etc/sysconfig/selinux` and change the line which reads `SELINUX=enabled` to either `SELINUX=permissive` (generates big log files and slows machine but allows for SELinux to be turned back on later more easily) or `SELINUX=disabled` (which disables it completely but can cause problems in the future if you want to switch it back on). You will also probably want to disable it immediately rather than doing a reboot before you can continue working on WACS - to do this, become root and run the following:

```
# /usr/sbin/setenforce 0
setenforce: SELinux is disabled
#
```

You can check this change has taken effect by using the `sestatus` command again.

DEB Installation Steps



Note

For Fedora/CentOS RPM package installs, please see the section called “RPM Installation Steps” instead.



Important

Before you start on an installation, please make sure that you have a statically allocated IP address, sensible hostname with a fully qualified domain name and that the machine is fully aware of these settings. For more information on these aspects, please consult the configuration guide.

Downloading The DEBs

The first step obviously is to download the appropriate packages for the operating system release, version and processor platform that you intend to run it on. Where a package contains *all* that means that it

is suitable for any processor architecture running that distribution of Linux. Currently DEB packaged versions are available for Ubuntu 8.04 and 8.10, while RPM packaged versions are available for Fedora 8,9 and 10. For more information on using the Fedora RPM packaged versions, please see the section called “RPM Installation Steps”

For an initial WACS installation (in this example for release 0.8.3 on an x86_64 machine running Ubuntu 8.04 LTS), you will probably want the following packages:

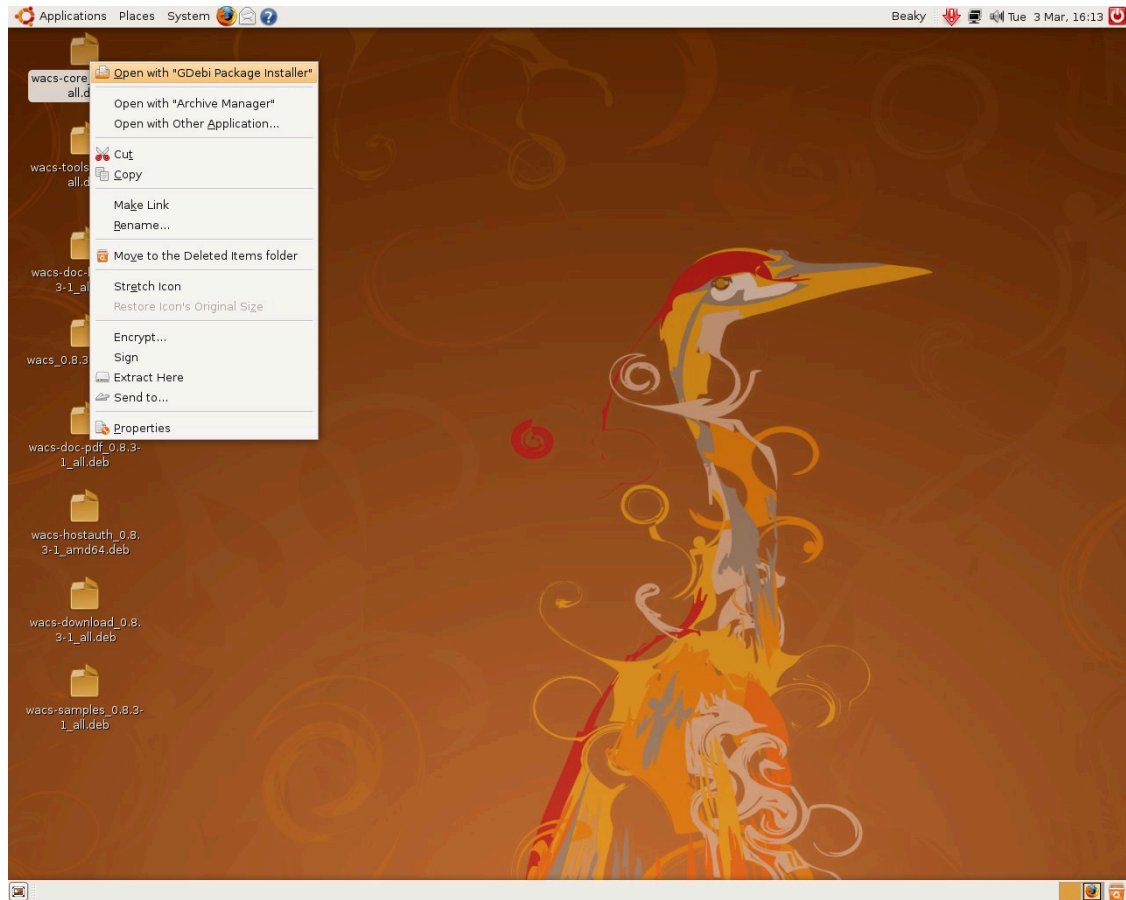
- **wacs_0.8.3-1.all.deb**
- **wacs-core_0.8.3-1.all.deb**
- **wacs-tools_0.8.3-1.all.deb**
- **wacs-samples_0.8.3-1.all.deb**
- **wacs-hostauth_0.8.3-1.amd64.deb**

If you plan on making use of the download toolset to connect to subscription sites for automatic downloads (although do be aware that only a very few sites are supported so far), you will also want to get the package called **wacs-download_0.8.3-1.all.deb**. You may also wish to download one of the two versions of the documentation package: **wacs-doc-pdf_0.8.3-1.all.deb** or **wacs-doc-html_0.8.3-1.all.deb** - you can always access the same documentation direct from our sourceforge web site.

DEB Installation

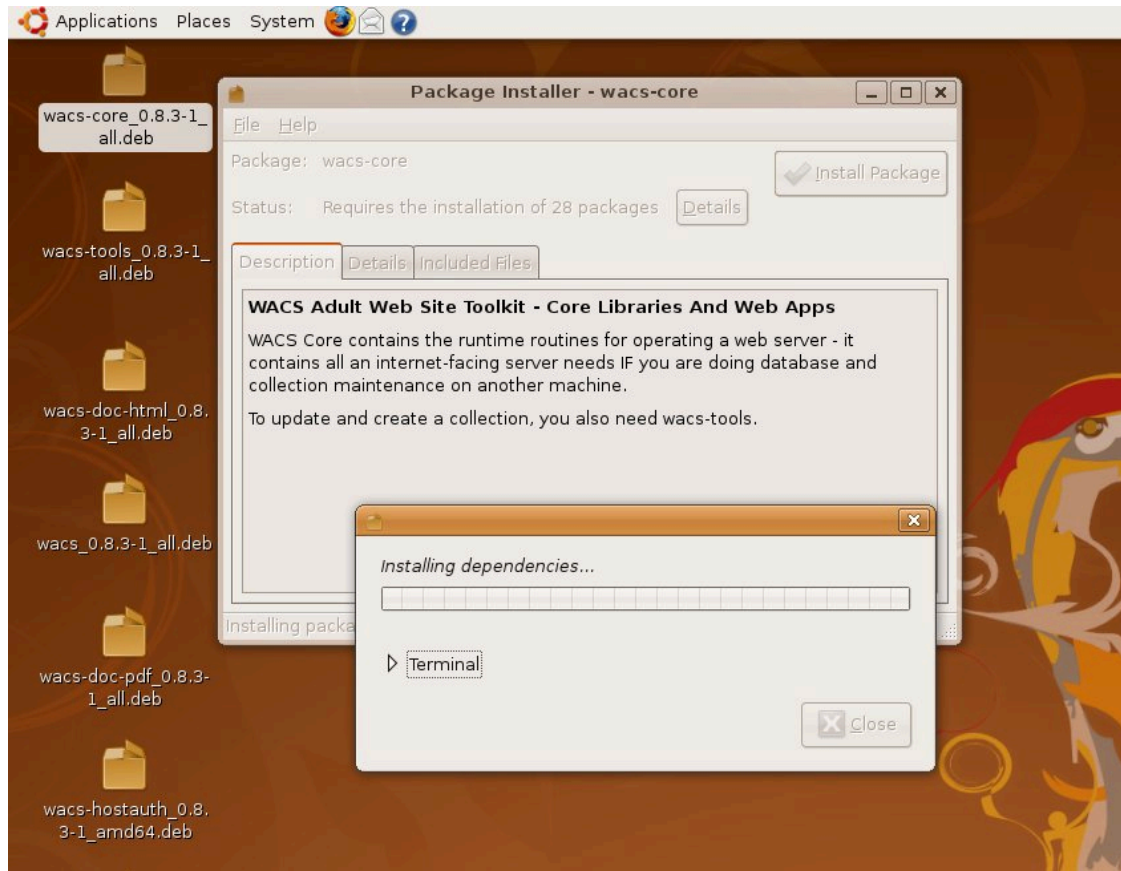
Once you've downloaded the right packages, you need to gain the appropriate privileges and then install the packages. There are any number of ways to do this, and you can pretty much use any of them; the examples given below use the desktop file manager and the **gdebi** installer. We have not used the synaptic package manager in the illustrations because that only really copes with software provided by the main Debian repositories. Unfortunately sourceforge.net does not, as we are writing this manual, support the creation of package repositories.

As it is the **GDebi** is configured to be the first action listed on the actions menu in the GNOME file manager as installed on Ubuntu. You get to this by right-clicking on the downloaded packages and choosing **Install With GDebi** . The order on this is a bit tricky, but if you start with **wacs-core** and **wacs-hostauth**, then do the other packages and finally do the main wacs package, this should work out OK.



Here we have all of the Wacs packages for Ubuntu/Debian downloaded onto our desktop and we start work with `wacs-core_0.8.3-1_all.deb`. We right-click on its icon and choose **Open with "GDebi Package Installer"**. Once selected we're told we're looking at Package: **wacs-core** and that it "Requires the installation of 28 packages".

While this 28 packages figure may sound high, it's what is actually needed for all the Wacs packages and these packages will be requested by whichever Wacs package is installed first. Similarly, if you've already installed any of these prerequisite packages on behalf of another application, the number needing to be installed now will be greatly reduced. What we are installing here are things like the **apache 2** web server, the **php5** programming language, the **MySQL 5** database and so on. This also attempts to install all the various Perl and PHP modules we use for database access and the like. Currently it will install **MySQL 5 server** as a dependency although you can still use Oracle as an alternative back end.



We click on the **Install Package** button and an additional pop-up box appears saying `Installing dependencies...`. This part of the installation procedure can actually take quite a while because on a normal Debian desktop install many of the requested packages will not be available locally. It will therefore make a connection to the internet software repositories and download the likes of **MySQL 5**, **apache 2**, **Php 5** and the perl modules required.

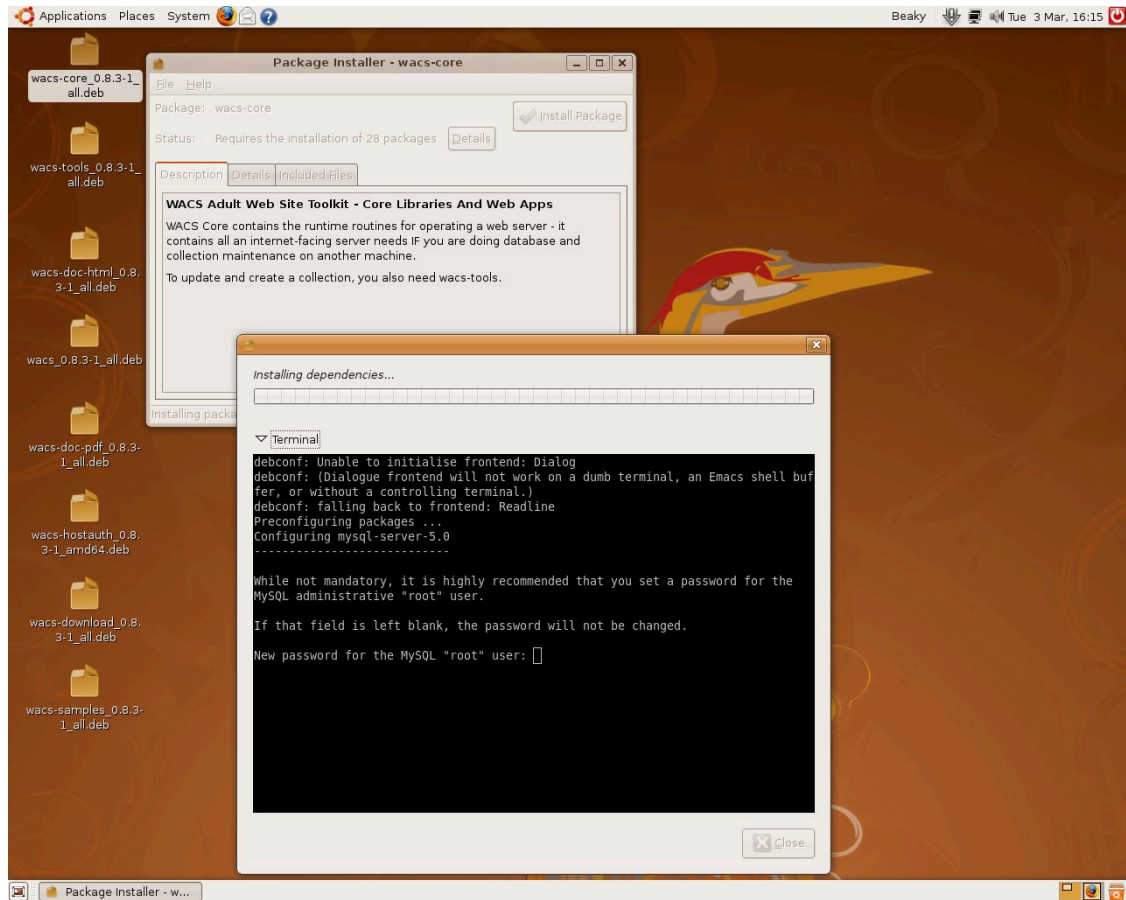


Tip

If you're on a dial-up internet connection, or have a seriously bandwidth limited broadband connection, you might wish to insert an appropriate DVD-ROM of Debian packages and configure the repositories used to include the local DVD-ROM as well. Of course this may or may not work depending on whether any upstream updates to those packages have been issued. An alternative would be to manually install the **MySQL 5 server**, **Apache 2** web server and the **PHP 5** language packages first from the DVD-ROM. This would then just leave the small Perl and PHP module packages still to be installed.

However, you'll notice that confusingly it doesn't show any progress which is unfortunately due to the fact that the pre-requisite MySQL package has actually prompted for some user input. If we click on the arrow to expand the Terminal window, we see the actual problem:

Installing WACS Using Packages



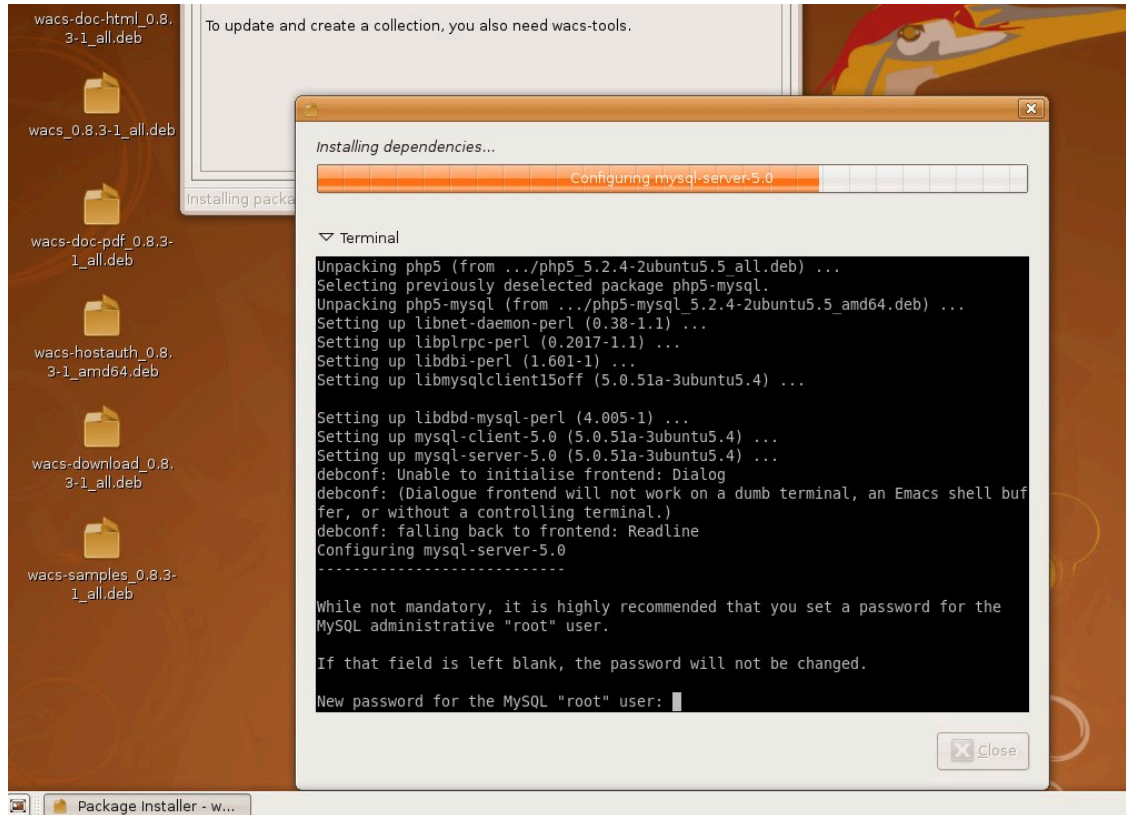
As you can see, it's actually prompting for a new password for the MySQL root user, which we can either enter or just press return to ignore. Generally setting a root password on the MySQL root user is an extremely good idea... With this message acknowledged, the install should actually start making some progress.



Note

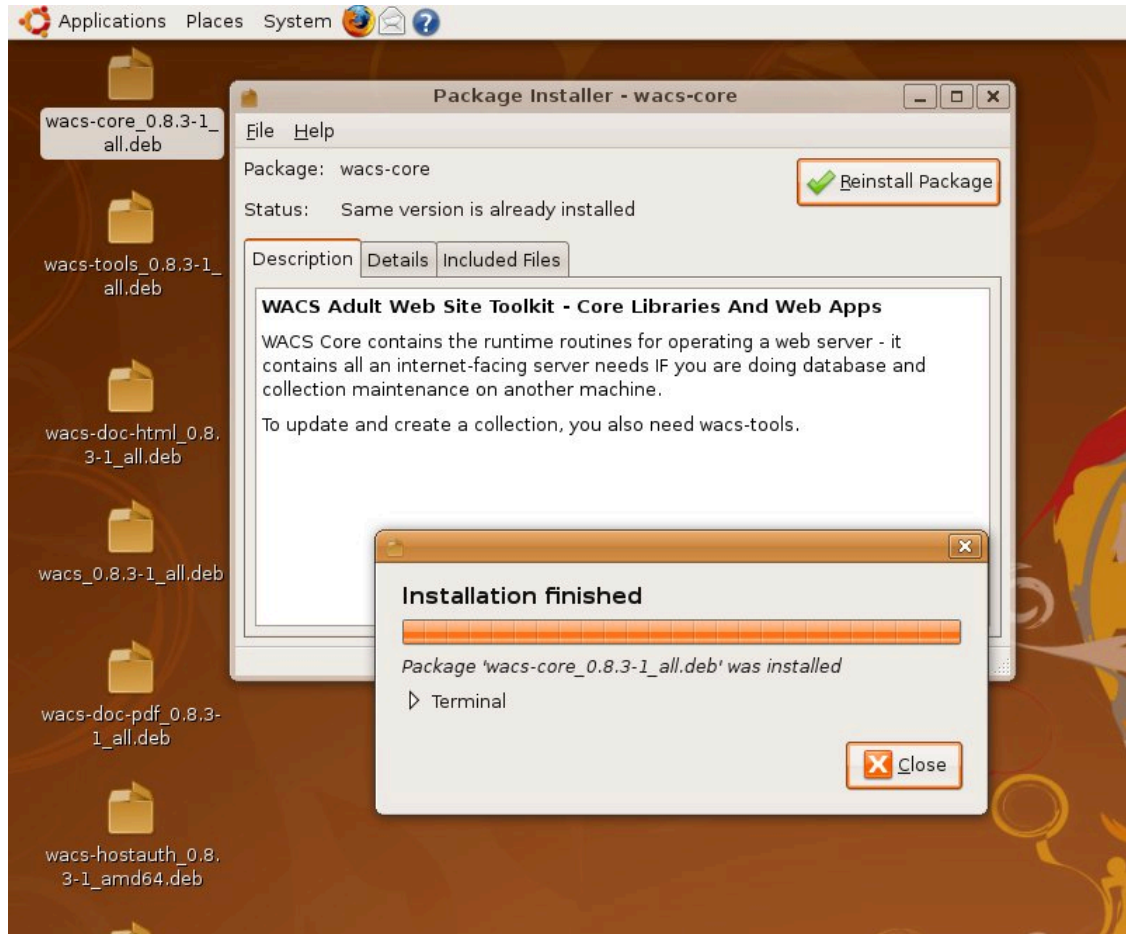
If you already had the **MySQL 5** server installed, and the MySQL root user password was set, the installation may well have proceeded automatically with no intervention required at all.

Installing WACS Using Packages

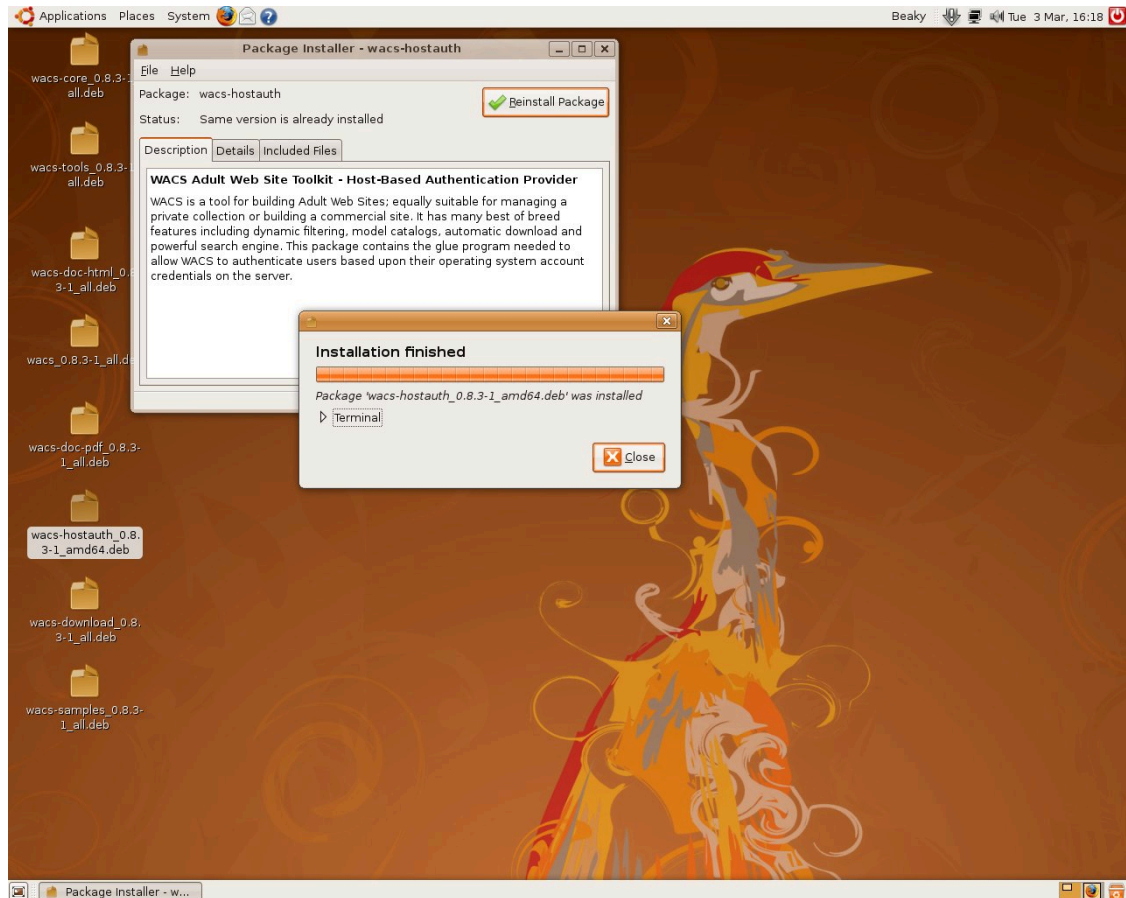


If you didn't choose to set a password, it'll prompt you for it again. If you did, it may well prompt you to enter that password to enable what you're working on. Once the dependencies have all be installed, the **wacs-core** package itself will be installed last. Finally we should see the message *Installation finished* but of course this is merely for **wacs-core** - the installation of the Wacs environment as a whole is not yet complete.

We then need to repeat this installation process with each of the other packages from the wacs suite, with **wacs-hostauth** next and the package called just **wacs** with it's version number last. The reason for this is that the **wacs** package is the “glue” that holds all the parts of Wacs together. In due course we'd hope you could merely install the **wacs** package itself and have all the pre-requisites automatically installed. Unfortunately the current package managers that do the dependency resolution all steadfastly refuse to include other manually downloaded packages in that dependency resolution process, and so it fails. As soon as sourceforge.net starts offering the ability to create package repositories, or we can get Wacs accepted into the repositories of the major distributions, these problems should all disappear.



As each package is installed, you should see this kind of message. Once they're all done, you should be ready to proceed to the next step, which is to get various pieces of infrastructure to start automatically when the system boots and the initial database prepared.



RPM/DEB Initial Setup

Unfortunately it's not easy to have the RPM/DEB packages complete the installation as there are questions that need to be answered as part of the process (like the root password for the database) and that's a serious non-sequitor for an RPM or DEB package. Until we've built a specialised web based installation setup tool (planned for the next release 0.8.4), there are a few steps that need to be done manually the first time WACS is installed to build the database. In addition to appearing here, they can be found in the text file `README.database` in the WACS installation tree. This installation tree is usually `/usr/share/wacs`.

Permissions Issues

The normal action of the RPM/DEB packages is to create a user account to hold all the datafiles, typically called simply `wacs`. Unless you choose to do otherwise, the images and video clips loaded into the WACS system are normally stored in the home directory of this account. For obvious reasons, the security on all the Wacs directories are locked down pretty hard, so you will need to pay attention to it. A new group also called `wacs` is created and initially the `wacs` and web server owner account are added to this. If you wish to read the documentation, samples and configuration files without always having to become the superuser first, you can simply add your own username to the `wacs` group.

There are a number of ways to do this including using the **System->Administration->Users and Groups** option or you can do it in the shell as root with the `usermod` command. Since the interface on Users and

Groups GUI is very different between different distributions and versions, we're going to stick with the command line method as that is portable. To do this you use the `usermod` command if you're using a superuser shell, substituting the `your_name` with your user name:

```
# usermod -a -G wacs your_name
#
```

If you're using Ubuntu you can also do it directly using the `sudo` command:

```
# sudo usermod -a -G wacs your_name
[sudo] password for your_name:
#
```



Warning

After you've added yourself to the `wacs` group, the change will almost certainly not take place within the current session. You will have to log out and log back in again for your membership to be recognised. The `groups` command lists the groups you are currently in; when this list includes `wacs`, things should be working - when it does not, they won't be!

1. Create the WACS database account (MySQL)

Connect to the database as the root user, giving the password as appropriate; if you've not set one the default is blank so just press return when prompted. You then create the database and the user account (once for each place you might be coming from), give access to that user account to the database, flush the contents and then quit. Here's a sample conversation - you obviously need to replace the `'myserver.example.com'` with whatever your real fully qualified domain name is. You might also wish to choose a more secure password, but do remember you need to change it in `/etc/wacs.d/wacs.cfg` (`dbpass` and `phpdbconnect` variables) as well or it just won't work.

Here goes:

```
# mysql -u root -p
Enter password:
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 80
Server version: 5.0.45 Source distribution

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

mysql> CREATE DATABASE wacs;
Query OK, 1 row affected (0.03 sec)

mysql> CREATE USER 'wacs'@'myserver.example.com'
-> IDENTIFIED BY 'wacs';
Query OK, 0 rows affected (0.08 sec)

mysql> CREATE USER 'wacs'@'myserver'
-> IDENTIFIED BY 'wacs';
Query OK, 0 rows affected (0.00 sec)
```

```
mysql> CREATE USER 'wacs'@'localhost'  
-> IDENTIFIED BY 'wacs';  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> GRANT ALL ON wacs.* TO wacs;  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> COMMIT;  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> FLUSH PRIVILEGES;  
Query OK, 0 rows affected (0.00 sec)  
  
mysql> QUIT;  
Bye  
#
```

2. Create the necessary database schemas

The next step is to log in as the wacs user account you just created and run the SQL scripts that create the various database tables. There are scripts provided for both MySQL 5 and Oracle 10, but this example is based upon using the MySQL 5 version. These should be found in `/usr/share/wacs/creation/MySQL5`.

```
# cd /usr/share/wacs/creation/MySQL5  
# mysql -u wacs -p wacs  
Enter password:  
Welcome to the MySQL monitor.  Commands end withh ; or \g.  
Your MySQL connection id is 82  
Server version: 5.0.45 Source distribution  
  
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.  
  
mysql> source create_mysql.sql  
WACS Database Table Creation Script for MySQL  
Commencing Table Creation:  
  1. Photographer  
Query OK, 0 rows affected (0.23 sec)  
  
  2. Vendor  
Query OK, 0 rows affected (0.01 sec)  
  
  3. Sets  
Query OK, 0 rows affected (0.01 sec)  
  
  4. Models  
Query OK, 0 rows affected (0.02 sec)  
  
  5. Assoc  
Query OK, 0 rows affected (0.01 sec)  
  
  6. Idmap  
Query OK, 0 rows affected (0.01 sec)
```



```
7. Download
Query OK, 0 rows affected (0.00 sec)

8. Tag
Query OK, 0 rows affected (0.01 sec)

9. Conn
Query OK, 0 rows affected (0.02 sec)

10. Keyword
Query OK, 0 rows affected (0.01 sec)

Tables Created - Committing Changes
Query OK, 0 rows affected (0.00 sec)

Completed.
mysql> commit;
Query OK, 0 rows affected (0.00 sec)

mysql> quit;
Bye
#
```

3. Create default database contents (optional)



Note

this step is *RECOMMENDED* unless you know precisely what you are doing. Some commercial sites may not wish to preload this data, but should substitute their own alternatives if they want certain features to work.



Warning

If you changed the password in step 1. above from the default value, you *NEED* to have made the matching change to the configuration file before doing this step.

There are three database tables that contain standard values, plus whatever you add to them over time; these are called keywords, photographers and vendors. In this step you will be loading some initial values into these database tables. To do this you need to go to `/usr/share/wacs/samples/database` and run the three populate commands in the `utils` directory on the XML data files there.

```
# cd /usr/share/wacs/samples/database
# /usr/share/wacs/utils/keywordpop keywords.xml
Inserting Entry For Keyword: seethru
Inserting Entry For Keyword: nopanties
Inserting Entry For Keyword: teen
[...]
Inserting Entry For Keyword: cyc
Inserting Entry For Keyword: country
Inserting Entry For Keyword: alley
# /usr/share/wacs/utils/photpop photographers.xml
Inserting Entries For Photographer: DFR
```



```
Inserting Entries For Photographer: SWE
Inserting Entries For Photographer: MAX
[...]
Inserting Entries For Photographer: JAN
Inserting Entries For Photographer: TOB
Inserting Entries For Photographer: HBM
# /usr/share/wacs/utils/vendpop vendors.xml
Inserting Entries For Site: SE
Inserting Entries For Site: WACSD
Inserting Entries For Site: FJ
[...]
Inserting Entries For Site: AMK
Inserting Entries For Site: KPC
Inserting Entries For Site: KHA
#
```

4. Import sample model records (optional)



Note

This is a very optional step but will help you if you're setting up a WACS web site for the first time.

This step loads two sample model records into the database to provide an example of how a typical model record might look. There are two such files provided in `/usr/share/wacs/samples/models` containing details of two models: Sabrina and Roxanne.

```
# cd /usr/share/wacs/samples/models
# wacsimport Sabrina-18.xml
Keyless ID map for JAFN
# wacsimport Roxanne-24.xml
#
```

5. Import sample set records (optional)



Note

This step is not yet possible but will be very shortly. Please keep watching the WACS website for the announcement of our demonstration web site.

You will first need to download the sets that appeal to you, so if you select set number 14 for instance, you will need the `set14.zip` file and the `set14.xml` file. Once you have these downloaded, place them both in the same directory and run the `wacsxmlin` program to load the data from the XML file. The zip file will be automatically unpacked and its contents placed in the images area. In this example, we're going to use the default layout, which if you haven't edited the configuration file will be gallery style (please see the discussion on Site Layout in the administration guide for more details).

```
# cd ~/Download
# wacsxmlin --default set14.xml
Unpacking archive:
  Roxanne07001.jpg
```

```
Roxanne07002.jpg  
Roxanne07003.jpg  
[...]  
#
```

Chapter 6. Installing With EasyInstall

Easyinstall: Download

WACS is obtainable from sourceforge where it is known as WACSip because of a name clash with a different package. The sourceforge site contains the latest code, documentation, news articles, mailing list details and even some screenshots. The URL is <http://wacsip.sourceforge.net> [<http://wacsip.sourceforge.net/>]. You can obtain WACS either by downloading one of the official releases or by pulling the very latest "bleeding edge" version from the Subversion repository. We do not use the older CVS version control system at all; WACS is a new project so it was started fresh under Subversion (svn). The official releases are also mirrored at <http://launchpad.net/wacs> [<http://launchpad.net/wacs/>].

To get the latest official release, go to the sourceforge project page [<http://sourceforge.net/projects/wacsip/>] and click on download. If the latest release is 0.8.1, then download `wacs-0.8.1.tar.gz`. Once downloaded, save it somewhere appropriate for unpacking - the installation will work without the archive, although there are sample configs and documentation files which are not installed anywhere. If for instance you've decided to keep `wacs` in an "src/wacs" directory of your home directory and your web browser has placed the downloaded file on your Desktop, do:

```
% cd
% mkdir -p src/wacs
% cd src/wacs
% tar -xzvf ~/Desktop/wacs-0.8.1.tar.gz
x wacs-0.8.1/README
[ ... ]
% cd wacs-0.8.1
```

Alternatively to get the very latest version from the sourceforge subversion repository, do the following:

```
% cd
% mkdir -p src/wacs
% cd src
% svn co https://wacsip.svn.sourceforge.net/svnroot/wacsip/trunk wacs
[ ... ]
% cd wacs
```

Easyinstall: Running

EasyInstall should be just that, but probably won't be. An installer is a complex thing and depends heavily on the environment around it. If you're running Fedora Core 6, Fedora 7, Fedora 8, Fedora 9 or Ubuntu 7.04 (Feisty Fawn) or 8.04 LTS and accept all the defaults, there is a reasonable chance it'll work. If you're running an older release with either MySQL < 5.0 or Apache < 2.2, you may well have problems... On other RPM-based distributions with the yum updater (SuSE, CENTOS, RHEL) you're in with a chance of it working, or at the very least installing most of the necessary infrastructure for you. Other non-RPM based distributions will almost certainly fail (apart from Ubuntu which should work), but if you feed back the error messages, we'll have a go at fixing it. If using an apt-get based distribution, modifying the places where it checks for "Ubuntu" to whatever your distribution returns when you do an `lsb_release -sir` may well help.

To run `easyinstall`, become the super user (root) and issue the following commands:

```
# cd unpack_location/install
# ./easyinstall
```

and follow the onscreen prompts. Packages invoked by this script will include your package manager (yum, apt-get, etc) and the perl CPAN installer. At the end of the package configuration questions, you will be shown your answers and asked for confirmation; if you answer n for no, you'll be asked the questions again. After that, once the installation phase starts, if you make a mistake in answering a question press <CTRL>-C to abort and start again. Between all the package managers and installers, you may well have to answer a couple of dozen questions in all.

The final system configuration step before starting using WACS is only applicable if you're using a version of Linux which includes the security hardening extension, SELinux. This currently is limited to the Red Hat based distributions like Fedora, Red Hat Enterprise Linux and CENTOS. Rumour has it OpenSuSE will shortly be including SELinux as an option. Unfortunately this release of WACS is not compatible with SELinux and so it'll have to be configured so as to ensure that SELinux is running in a reduced mode that will not block the WACS components from working. We hope to have this resolved by the next release of WACS. You can determine the current mode of SELinux using the `sestatus` command:

```
% /usr/sbin/sestatus
SELinux status:                disabled
%
```

To change the normal operational mode, you need to edit the file called `/etc/sysconfig/selinux` and change the line which reads `SELINUX=enabled` to either `SELINUX=permissive` (generates big log files and slows machine but allows for SELinux to be turned back on later more easily) or `SELINUX=disabled` (which disables it completely but can cause problems in the future if you want to switch it back on). You will also probably want to disable it immediately rather than doing a reboot before you can continue working on WACS - to do this, become root and run the following:

```
# /usr/sbin/setenforce 0
setenforce: SELinux is disabled
#
```

You can check this change has taken effect by using the `sestatus` command again.

At that point the installation should be complete and you'll need to look at the getting started document for how to set up a WACS collection.

Chapter 7. Manual Installation

Caution

This is a complex task and some level of familiarity with the Unix/Linux command line will be needed to be successful. Please make sure that all of the packages/services described in the prerequisites chapter have been installed and are running correctly.

Manual Installation: Steps

1. Create the wacs user account and group and place your own username in the group file (`/etc/group` or distributed name service equivalent). This is easily done with the `useradd` command on Redhat-based distributions:

```
# groupadd -r wacs
# useradd -m -g wacs -r -c "WACS Files Owner" \
  -s /bin/bash wacs
#
```

2. check the web server is working, install the barebones WACS index pages from the `htmlbones` directory of the distribution into your web tree and check you can see it. On a default Fedora Core 5 installation, this would be done with:

```
# cd unpack_location
# mkdir /var/www/html/wacs
# cp -rp htmlbones/* /var/www/html/wacs
#
```

3. For the perl modules, first check whether your operating system distribution includes them - Fedora Core 5 had packages called `perl-DBI` and `perl-DBD-MySQL` - so these could be simply installed with `yum install perl-DBI` and `yum install perl-DBD-MySQL`. The next easiest way to install the necessary perl modules, if they are not already present, is to use the `cpan` command. On some recent releases, the `cpan` command has become optional, you will have to do a `yum install cpan` first. Once you have `cpan`, the necessary perl module installs can typically be done with:

```
# cpan install XML::Simple
# cpan install Data::Dumper
# cpan install File::Basename
# cpan install MIME::Base64
#
```

4. install the `Wacs.pm`, `WacsUI.pm`, `WacsStd.pm` and `WacsId.pm` perl modules into the `site_perl` directory of your system - change the 5.8.8 to whatever your current version of perl is. [NB: note the change of case of the first letter of the perl module name from `wacs.pm` to `Wacs.pm`, and of `wacsid.pm` to `WacsId.pm`]:

```
# cd unpack_location
```

```
# cp modules/wacs.pm /usr/lib/perl5/site_perl/5.8.8/Wacs.pm
# cp modules/wacsui.pm /usr/lib/perl5/site_perl/5.8.8/WacsUI.pm
# cp modules/wacsstd.pm /usr/lib/perl5/site_perl/5.8.8/WacsStd.pm
# cp modules/wacsid.pm /usr/lib/perl5/site_perl/5.8.8/WacsId.pm
#
```

5. install the wacs PAM (Pluggable Authentication Modules) configuration into the `/etc/pam.d` directory. You will also need to compile the `pam_auth` program using the provided make file and then install the binary created into wherever your `tooldirs` configuration variable is set to (a common value is `/usr/local/bin`). If this compilation fails, the most likely cause is that the `libpam` development package is not installed. Then you need to create the `/var/run/wacs` directory where the dynamic leases files are stored and change it's ownership to `apache` (or whatever your web server user is).

```
# cp unpack_location/security/wacs.pam /etc/pam.d/wacs
# chown root.root /etc/pam.d/wacs
# chmod 644 /etc/pam.d/wacs
# cd unpack_location/security
# make -f Makefile all
Building pam_auth.x86_64-Fedora8 ...
cc -o pam_auth.`arch`-`lsb_release -si | sed 's/\ /_/g'`-`lsb_release -sr`
    pam_auth.c -lpam
# ls pam_auth*
pam_auth    pam_auth.c  pam_auth.x86_64-Fedora8
# cp pam_auth.x86_64-Fedora8 /usr/local/bin/pam_auth
# chown root.wacs /usr/local/bin/pam_auth
# chmod u+s /usr/local/bin/pam_auth
# mkdir /var/run/wacs
# chown apache.apache /var/run/wacs
#
```



Note

if you run `selinux` (Security Enhanced Linux) on Fedora Core or Redhat (or another future distro that includes it), you will need to give `apache` privilege to read the `/var/run/wacs` directory - this can be done by changing the context of the directories and files. The commands to do this are:

```
# chcon system_u:object_r:httpd_sys_content_t /var/run/wacs
# chcon -R system_u:object_r:httpd_sys_content_t /var/run/wacs/*
#
```

If the leases file does not exist when you first do this and you encounter problems, try using the second of these two commands again.

6. install the wacs application programs into the `cgi-bin` tree:

```
# cd unpack_location
# cp index/wacs* models/wacs* presentation/wacs* /var/www/cgi-bin/
# cp retrieval/wacs* search/wacs* tag/wacs* /var/www/cgi-bin/
# cp security/wacs* manage/wacs* /var/www/cgi-bin/
# chmod 755 /var/www/cgi-bin/wacs*
#
```

7. install the wacs application programs into the cgi-bin tree:

```
# cd unpack_location
# cp index/wacs* models/wacs* presentation/wacs* /var/www/cgi-bin/
# cp retrieval/wacs* search/wacs* tag/wacs* /var/www/cgi-bin/
# cp security/wacs* manage/wacs* /var/www/cgi-bin/
# chmod 755 /var/www/cgi-bin/wacs*
#
```

8. copy the applications that are just duplicate versions of existing commands and change the appropriate mode variables:

```
# cd /var/www/cgi-bin
# cp wacsmodelpage wacsmphumbs
# editor wacsmphumbs
# cp wacsimgcats wacsvidcats
# editor wacsvidcats
# cp wacsimgcats wacsphotcats
# editor wacsphotcats
# cp wacsimglist wacsvidlist
# editor wacsvidlist
# cp wacsnewsets wacsnewvideo
# editor wacsnewvideo
#
```

edit the file and change the mode variable (thumbsmode in this case). Repeat this process for wacsimgcats becomes wacsvidcats and wacsphotcats, and so on. At the end, make sure all of the copies are executable:

```
# cd /var/www/cgi-bin
# chmod 755 wacs*
```

9. install the configuration file, `wacs.cfg` into a suitable location such as `/etc/wacs.d` or `/usr/local/etc/wacs.d`. Edit this file and make sure the key settings are right for your server, specifically the location of the image files, the location of the video files and the server name in the URLs. You will also need settings for the database user name and password you intend to use, and the environment and path locations needed for the database system you are using. For more information, see the Configuration Guide.

10. create a suitable permanent access control list in the configuration directory chosen above, the supplied `wacs.acl` should provide a suitable template. This step can be skipped if you're only ever going to use lease-based access with logins. For more information on the format of the access control lists, please see the section on security in the Configuration guide.

11. create a suitable owner account for the wacs data tables in your database system. The instructions here cover doing this for both MySQL and Oracle 10g, in that order. With MySQL 5.x, this would be done with:

```
% mysql --user=root
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 17 to server version: 5.0.22
```

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

```
mysql> create database wacs;

mysql> create user 'wacs'@'myserver' identified by 'wacs';

mysql> create user 'wacs'@'localhost' identified by 'wacs';

mysql> grant all on wacs.* to wacs;

mysql> commit;

mysql> flush privileges;

mysql> quit
```

With Oracle 10g, this would be something like:

```
% sqlplus
SQL*Plus: Release 10.1.0.3.0 - Production on Fri Sep 29 14:53:56 2006
Copyright (c) 1982, 2004, Oracle. All rights reserved.

Enter user-name: system
Password: *****

Connected to:
Oracle Database 10g Release 10.1.0.3.0 - 64bit Production

SQL> create user wacs identified by wacs;

User created.

SQL> grant connect, resource to wacs;

Grant succeeded.

SQL> alter user wacs default tablespace main
      quota unlimited on main;

User altered.

SQL> commit;

Commit complete.

SQL> quit
Disconnected from Oracle Database 10g Release 10.1.0.3.0 - 64bit Production
```

Of course there is a huge amount of variance in how any given database is installed, so you will need some knowledge about your installation. For instance, your main tablespace may not be called "main" as it is in the example.

12. login as the database user just created and run the table create SQL script from the creation directory of the wacs distribution. These scripts are called by a single creation script, the one for oracle is called `create_oracle.sql`, the one for MySQL is called `create_mysql.sql`. To run this on MySQL 5.1 using the account created in the step above, you would do the following:

```
% cd unpack_location/creation
% mysql --user=wacs --password=wacs wacs < create_mysql
[...]
```

To run this on Oracle 10g using the account created in the step above, you would do the following:

```
% cd unpack_location/creation
% sqlplus wacs/wacs @create_oracle
SQL*Plus: Release 10.1.0.2.0 - Production on Fri Oct 6 19:11:41 2006
Copyright (c) 1982, 2004, Oracle. All rights reserved.
```

```
Connected to:
Oracle Database 10g Release 10.1.0.3.0 - 64bit Production
```

```
WACS Database Table Creation Script for Oracle
```

```
Commencing Table Creation:
```

```
1. Photographer
```

```
Table created.
```

```
2. Vendor
```

```
Table created.
```

```
3. Sets
```

```
Table created.
```

```
4. Models
```

```
Table created.
```

```
5. Assoc
```

```
Table created.
```

```
6. Idmap
```

Table created.

7. Download

Table created.

8. Tag

Table created.

9. Conn

Table created.

10. Keyword

Table created.

Tables Created - Committing Changes

Commit complete.

Completed.

Disconnected from Oracle Database 10g Release 10.1.0.3.0 - 64bit Production
%

- 13.The penultimate major activity is to install the tools scripts, and if required the download and migrate tools, into a suitable directory, normally this would be `/usr/local/bin`, but it could be put within the wacs tree if desired. `/usr/local/bin` is usually in the default path for all the shells and thus available to user accounts without further work. To install, do:

```
# cd unpack_location
# cp -p tools/* /usr/local/bin
# cp -p download/* /usr/local/bin
# cp -p migrate/* /usr/local/bin
#
```

If you want to put it somewhere else, within the wacs home area would be fine, somewhere like `/home/wacs/bin`, but you will then need to add that directory to the path of your shell. For the C-shell, you would add `set path=(/home/wacs/bin $path)` into the `.cshrc` file in the home directory of your own account and those of other people who might be adding contents to the wacs server. For the Bourne style shells (sh,bash,etc), you would need to add `PATH=/home/wacs/bin:$PATH` and `export PATH` to the `.profile` or `.bashrc` files in the home directories. Once added, depending on the shell, you may need to type `rehash` to rescan the path for the new commands.

- 14.The next step is to populate the vendor database with the sample records, which can be done with:

```
# cd unpack_location/populate
# ./vendpop vendors.xml
Inserting Entries For Site: ATKP
Inserting Entries For Site: AMK
Inserting Entries For Site: ATE
Inserting Entries For Site: SE
#
```

Please contribute back vendor descriptions you create to be included in the next release.

15. Next we need to preload the keywords database table so that the automatic tagging will occur correctly. We do this with:

```
# cd unpack_location/populate
# ./keywordpop keywords.xml
[ ... ]
#
```

16. Finally we need to load the photographers database with some initial example records, which can be done with:

```
# cd unpack_location/populate
# ./photpop photographers.xml
[ ... ]
#
```



Note

Although we have discussed the steps needed to get WACS working under SELinux, we've currently not managed to track down all of the dependencies and in the interests of expediency we have gone ahead with code development without it. At this point we do not consider the WACS 0.8.1 release or any release prior to that to be SELinux compatible.

We therefore recommend that your operating system is configured so as to ensure that SELinux is running in a reduced mode that will not block the WACS components from working. This is only an issue on Fedora and other Red Hat based releases at present. We hope to have this resolved by the next release of WACS. You can determine the current mode of SELinux using the `sestatus` command:

```
% /usr/sbin/sestatus
SELinux status:                disabled
%
```

To change the normal operational mode, you need to edit the file called `/etc/sysconfig/selinux` and change the line which reads `SELINUX=enabled` to either `SELINUX=permissive` (generates big log files and slows machine but allows for SELinux to be turned back on later more easily) or `SELINUX=disabled` (which disables it completely but can cause problems in the future if you want to switch it back on). You will also probably want to disable it immediately rather than doing a reboot before you can continue working on WACS - to do this, become root and run the following:

```
# /usr/sbin/setenforce 0
setenforce: SELinux is disabled
```

#

You can check this change has taken effect by using the **sestatus** command again.

Chapter 8. Upgrading An Existing Installation

About Upgrading

If working on a script to make an installation as complex as WACS work right was tough, imagine how much harder it is to upgrade an existing installation where you have no idea what has been changed and what hasn't? Simply, it can't easily be automated, so upgrading is of necessity a two step process. Run a script for the easy bit, and manually fix up those things the script can't do automatically without danger of destroying data. To help you along, the script and this document give some guidance on the issues, but you do need to have some idea of what version you're coming from and going to in order for that to make sense. The version number of the current version is given on the HTML home page, [index.html](#).

The Upgrade Command

The easy bits are looked after by the upgrade command; to run this download and unpack the new distribution, and as the super user (root) run the following commands:

```
# cd unpack_location/install
# ./upgrade
WACS - Upgrade
-----
[... ]
Do you wish to continue? (y/n): y
[... ]
#
```

At the end of it's run, upgrade will print out some key notes about things that will require manual attention to get the new release working. The section below will give you some guidance on how these may be achieved.

Additional Steps

The upgrade command will give you some information on what extra steps you may need to take to migrate to this release. For example, it may tell you that a new database field needs to be added to a particular model schema. In the transition from 0.5 to 0.6.x the `mrace` field was added, and upgrade will tell you about this. First step is to find the specification of the field from the appropriate SQL script in the creation directory, so for Oracle this will be `creation/ora_models.sql`. From this you will see that the field specification for Oracle is:

```
[... ]
  mrace          varchar2(15),
[... ]
```

You have three options for adding this to the database - you can choose to alter the existing schema (may leave fields in an odd order in describe); you can rename the existing table, create the new one, copy the data across and then repoint any relational constraints to the new table; or you can export your entire

database, create a fresh one and import the records back in (the tools for this are incomplete in 0.6.1 but just about usable). The former is quick and easy if the database supports it but leaves the field list in an odd order; the middle one is more work but produces a fully "normal" schema in the end but requires serious black magic if your database understands relational constraints. The final one is **VERY* experimental* at this point but will improve with time.

Here is a worked example that shows how to use the alter table syntax in Oracle's SQL*Plus command interpreter to add one field called `mrace`:

```
% sqlplus
[... ]
Username: wacs
Password: ****

sqlplus> alter table models
         > add ( mrace          varchar2(15) );

Table altered.
sqlplus> commit;

Commit complete.
sqlplus> desc models
[... ]
  MRACE                                VARCHAR2(15)
sqlplus> quit
%
```

Another issue you need to be aware of is that the upgrade script will not over-write any existing files in the `wacs` web document tree (by default this is `/var/www/html/wacs`) because you may well have tailored them and we wouldn't want to overwrite those. You may well therefore need to look at what is in the `htmlbones` directory and copy some of the new files across into your web tree, or merge the new html into your modified version of the pages.

Chapter 9. Troubleshooting

Introduction

Obviously we hope the installation script will create a running installation for you, but there will no doubt be occasions when it does not. Before seeking help via the mailing lists and other resources on the sourceforge site [<http://wacsip.sourceforge.net/>], there are some things you should clarify. The first of these is to confirm what the status of the various subsystems are. Here's a quick check list:

Troubleshooting Checklist

- Is there an error being reported?
 1. If you get the error message `Can't find lsb_release in order to determine distribution` and you are on an older Fedora Core or CentOS version, try running: **yum install redhat-lsb** and then running the installer again. Upgrades often leave out this package although it should be part of the standard operating system.
 2. If you get the error message `DBI connect('wacs:myserver.myisp.com','wacs',...) failed: Can't connect to MySQL server on 'myserver.myisp.com' (110) at ./vendpop line 39` check that you can ping the hostname of your server locally. Often people don't have things set up so that a machine with an "internet name" can see itself by the same name on the local network. The above error is a symptom of this problem.
 3. Check the apache web server log file in `/var/log/httpd/myserver-errorlog`
 4. Check the system messages in `/var/log/messages`
 5. Check the output from the kernel by running **dmesg** (a common cause of trouble is the SELinux security mechanism, an `avc_denied` message in the output of `dmesg` is a solid clue to this - see comments below on SELinux)
- Is the web server running?
 1. If you point a web browser at the top level URL of your server, do you get a web page back, be it a distribution-supplied test page or previous pages you placed there?
 2. Can you get the wacs main page? (`http://myserver/wacs/`)
 3. Can you get any response from the WACS cgi-bin programs, even so much as a coloured background to a blank screen? (`http://myserver/cgi-bin/wacsnewmodels`)
 4. Is there an httpd process running? (**ps wax | grep httpd**)
 5. Is the HTTPD set to start automatically? (**chkconfig --list httpd**)
- Is the database server running?
 1. See if you can connect using the SQL command line application - called `mysql` for MySQL, `sqlplus` for Oracle 10g.

2. Check for the database processes running - mysqld for MySQL, a whole cluster of oracle* and ora_* processes for Oracle 10g.

Special Notes About SELinux

SELinux is an enhancement to Linux that allows potentially vulnerable services (like an internet-exposed web server) to be operated on a basis where each action the program tries to take needs to be explicitly allowed, rather than the normal allowed unless denied approach of most Unix environments. As such SELinux presents a whole new group of challenges for getting WACS to work, because we have to extend the ruleset as to what is allowed and what is not. It can be done, but it will take work and some experimentation. Wherever we have not used the Operating System supplied packages (Web Server, Database, etc), we're going to have to add those rules. The first thing to check is whether SELinux is enabled - to do this, type:

```
% sestatus
SELinux status:                enabled
SELinuxfs mount:              /selinux
Current mode:                  permissive
Mode from config file:        permissive
Policy version:                20
Policy from config file:      targeted
%
```

If it's either disabled, or is enabled but with a current mode of permissive, it's not actually going to be causing us a problem right now. If it is enabled and enforcing, we've got to work on it. The web server process needs a security context of `httpd_sys_content_t` to be present on any directory it needs to access, so the first step is to add this context to each directory (outside of the normal ones) that it is likely to access. This is done with the **chcon** commands shown above in the manual install chapter - example:

```
# chcon system_u:object_r:httpd_sys_content_t /var/run/wacs
#
```

In addition to the directory gaining the `httpd_sys_content_t` security context, any pre-existing files will also need the same, so this can be done with:

```
# chcon -R system_u:object_r:httpd_sys_content_t /var/run/wacs/*
#
```

To inspect the security context of a file or directory, you use the `-Z` option to the `ls` command:

```
# ls -Z /var/run/wacs/
-rw-r--r--  apache apache system_u:object_r:httpd_var_run_t leases.acl
#
```

While the `easyinstall` script does try to set these for all the areas the web server might go (`/var/run/wacs`, `/etc/wacs.d` and the files area `/home/wacs/*`), any problems which are causing `avc_denied` messages in the `dmesg` output are most likely down to this issue.

Index

D

database

Initial Setup, 17

E

easyinstall, 23

G

Group Membership, 17

I

Installation

Overview, 1

Using Packages, 8

L

Linux Operating Systems, 3

M

Manual Installation, 25

P

packages, 8

DEB for Ubuntu/Debian, 10

Initial Setup, 17

RPMs for Fedora Linux, 8

permissions, 17

S

SELinux

disabling, 10

Special Notes on..., 36

T

troubleshooting, 35

U

upgrade, 33