



Wacs Configuration Guide

Fifth Edition

for WACS 0.8.5

B "Beaky" King

Published 16th March 2010

Wacs Configuration Guide

by B "Beaky" King

for WACS 0.8.5

Published 16th March 2010

Copyright © 2006, 2007, 2008, 2009, 2010 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 configuration files used by the system which are all implemented in the Industry Standard eXensible Markup Language (XML). The intended audience is web developers and WACS site managers who wish to tailor an existing WACS installation to meet their precise requirements; people merely wishing to use or manage an existing WACS installation may well find the default configurations provided suffice.

The WACS source code and other documentation and support tools can all be found at the WACS website at Sourceforge [<http://wacsip.sourceforge.net/>] and at the Wacs page at launchpad [<https://launchpad.net/wacs/>]. There is a demonstration site for WACS available at PinkMetallic.com [<http://www.pinkmetallic.com/>] - *CAUTION: contains adult material*. 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

| | |
|--|----|
| I. WACS Configuration: Introduction | 1 |
| 1. Introduction to WACS Configuration | 2 |
| WACS Overview | 2 |
| About This Book | 2 |
| About The Examples | 2 |
| 2. WACS Configuration Architecture | 3 |
| Architectural Overview | 3 |
| Quick Intro To XML | 3 |
| Permanent Configuration | 4 |
| Transient Configuration | 5 |
| 3. Prerequisites | 6 |
| OS Configuration | 6 |
| Static IP Address And Name | 6 |
| ISP Requests | 7 |
| Port Forwarding | 8 |
| NFS File Sharing | 8 |
| Database Configuration | 9 |
| 4. Altering The Configuration | 11 |
| Altering: An Introduction | 11 |
| Changing Colours | 11 |
| 5. Caching And Performance Tuning | 15 |
| Objectives | 15 |
| The usedirect Mechanism | 15 |
| Using The Scaled Cache | 16 |
| 6. Custom Configurations | 17 |
| Introduction | 17 |
| Vendor Style Menus in Wacs 0.8.x and later | 17 |
| Background Info | 17 |
| How To Configure Vendor Mode | 17 |
| Running Multiple WACS instances | 19 |
| Background | 19 |
| How To Configure Multiple WACS Instances | 19 |
| II. WACS Configuration: Reference | 22 |
| 7. Database Environment | 23 |
| Overview: Database | 23 |
| wacs.cfg: Database Environment Section | 23 |
| Database Engine Notes | 24 |
| Database: MySQL 5 | 24 |
| Database: Oracle | 24 |
| 8. Database Tables | 25 |
| Overview: Tables | 25 |
| wacs.cfg: Database Section | 25 |
| 9. Server Details | 26 |
| Overview: Server | 26 |
| wacs.cfg: Server Section | 26 |
| 10. FSLoc: File System Locations | 28 |
| Overview: FSLoc | 28 |
| wacs.cfg: fsloc section | 28 |
| 11. Security | 31 |
| Overview: Security | 31 |
| wacs.cfg: security | 31 |

| | |
|---------------------------------------|----|
| 12. Download | 32 |
| Overview: Download | 32 |
| wacs.cfg: Download | 32 |
| 13. Colours | 33 |
| Overview: Colours | 33 |
| wacs.cfg: Colours Section | 33 |
| 14. Layout | 34 |
| Overview: Layout | 34 |
| wacs.cfg: Layout Section | 34 |
| 15. Precedence | 36 |
| Overview: Precedence | 36 |
| wacs.cfg: Precedence | 36 |
| 16. Debug | 37 |
| Overview: Debug | 37 |
| 17. WACS Configuration: Menus | 38 |
| WACS Menu System Overview | 38 |
| Basic Default Menus | 38 |
| Menu XML tokens | 39 |
| Basic Menu XML Types | 39 |
| Sample Menu | 39 |
| Menu Handlers | 39 |
| The Special handler Menu | 40 |
| 18. WACS Web Apps: Presentation | 41 |
| Presentation Applications | 41 |
| wacsnewmodels | 41 |
| wacsnewsets | 41 |
| wacsnewvideo | 41 |
| wacsshow | 42 |
| wacsvidshow | 42 |
| wacsideshow | 42 |
| wacsfp | 43 |
| 19. WACS Web Apps: Indices | 44 |
| Index Applications | 44 |
| wacsimgcats | 44 |
| wacsimglist | 44 |
| wacsvidcats | 44 |
| wacsvidlist | 45 |
| wacsphotcats | 45 |
| wacslistphotog | 45 |
| wacsstats | 45 |
| 20. WACS Web Apps: Models | 46 |
| Models Applications | 46 |
| wacsmodelthumbs | 46 |
| wacsmodelindex | 48 |
| wacsmodelpage | 48 |
| wacsmpthumbs | 49 |
| wacsmpmini | 49 |
| 21. WACS Web Apps: Tag | 50 |
| Tag Applications | 50 |
| wacstagsets | 50 |
| wacslisttags | 50 |
| 22. WACS Web Apps: Search | 51 |
| Search Applications | 51 |
| wacsimgselect | 51 |

| | |
|--|----|
| wacsvidselect | 51 |
| wacsmodelselect | 51 |
| 23. WACS Web Infrastructure: Retrieval | 52 |
| Retrieval Applications | 52 |
| wacsfull | 52 |
| wacsframe | 52 |
| wacsimg | 52 |
| wacsindex | 53 |
| wacspage | 53 |
| wacsthumb | 53 |
| wacszip | 53 |
| wacsusc2257 | 53 |
| 24. WACS Web Infrastructure: Tags | 54 |
| Tag Infrastructure Applications | 54 |
| wacsaddtag | 54 |
| wacsdeltag | 54 |
| wacsmarktag | 54 |
| Index | 56 |

List of Figures

| | |
|------------------------------|---|
| 2.1. Logical Structure | 4 |
|------------------------------|---|

List of Tables

| | |
|---|----|
| 2.1. List of Permanent Config Files | 4 |
| 2.2. List of Transient Config Files | 5 |
| 3.1. Steps To Get Network Connections To Work | 9 |
| 7.1. Configuration: Database Section | 23 |
| 8.1. wacs.cfg: Tables | 25 |
| 9.1. Configuration: Server | 26 |
| 10.1. FSLoc: File Storage Locations | 28 |
| 10.2. FSLoc: Icon & Glyphs | 28 |
| 10.3. FSLoc: Applications & Documents | 29 |
| 11.1. Security Section: wacs.cfg | 31 |
| 12.1. Download Section: wacs.cfg | 32 |
| 13.1. Colours Section: wacs.cfg | 33 |
| 14.1. Layout Section: wacs.cfg | 34 |
| 15.1. Precedence Section: wacs.cfg | 36 |
| 17.1. The Default Menus | 38 |
| 17.2. The Menu XML Types | 39 |
| 20.1. URL Options to WacsModelThumbs | 47 |

List of Examples

| | |
|--|----|
| 2.1. Simple WACS Configuration file | 3 |
| 3.1. Determining The Ethernet Address | 6 |
| 15.1. Sample Precedence Section | 36 |
| 16.1. Sample of Global Debug Setting | 37 |
| 16.2. Sample Of Specifically Targetted Debug | 37 |
| 17.1. Sample Menu | 39 |
| 17.2. Sample handler Menu | 40 |

Part I. WACS

Configuration: Introduction

This section of the WACS Configuration guide aims to offer a general overview of the configuration system and a few examples of how to make use of it to tailor your WACS installation to your precise requirements.

Chapter 1, *Introduction to WACS Configuration*

Chapter 2, *WACS Configuration Architecture*

Chapter 3, *Prerequisites*

Chapter 4, *Altering The Configuration*

Chapter 5, *Caching And Performance Tuning*

Chapter 6, *Custom Configurations*

Chapter 1. Introduction to WACS Configuration

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 will mention the absolute basics of XML in passing but it does really assume familiarity with the basic XML concepts of tags, attributes, and nested layout. If you have a basic familiarity with either XML or it's close cousin, HTML, you should be fine.

About This Book

This book is a configuration guide for WACS site managers or web developers seeking to customise the WACS environment to better suit their exact needs. In most cases, a perfectly running system will be created by the installer script, and none of the changes described here will be needed in order to create or operate a WACS server. What this provides is access to the huge range of control and configuration options that WACS uses both in it's own extensive tool set and in applications developed using the WACS API (Applications Programming Interface) libraries which are available in both Perl and PHP languages.

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. We have a WACS demonstration site available at PinkMetallic.com [<http://www.pinkmetallic.com/>] - *CAUTION: contains adult material.*



Tip

In addition as you work with this book, it will be distinctly advantageous to have administrator access to a test-bed WACS system without a live user community as many of the configuration options will directly affect the end user experience.

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 are available for download from our WACS demonstration site at PinkMetallic.com [<http://www.pinkmetallic.com/>] - *CAUTION: contains adult material* Access to the PinkMetallic.com [<http://www.pinkmetallic.com/>] site is currently free, but a small charge may have to be introduced at a later date if revenues from referrals and donations aren't as hoped for.

Chapter 2. WACS Configuration Architecture

Architectural Overview

The WACS system contains a number of components; the basic web applications, the web applications for collection administration, some command line tools and an application programming interface (API) which allows programmers access to the facilities the WACS infrastructure provides from their own programs. Most of these aspects are provided for in the standard WACS distribution with the exception of the PHP5 programming language API which is provided as a separate download. All of these components are tied together by a common configuration system which is based upon the eXtensible Markup Language (XML) standard.



Tip

WACS uses a very simple dialect of XML with a very simple single attribute mechanism. It's designed to be as easy to read and edit as possible

Within the WACS system we have two types of configuration files: Permanent and Transient. The permanent files are those that are either created by the installer when the WACS software is installed or created by the WACS site administrator from one of the provided templates. The transient configurations are those that are created and updated during usage of the WACS system; transient configuration files are always changing and will be recreated if they are lost by a system reboot or similar. It is extremely unlikely you will ever need to make any changes to the transient files, but understanding them will aid with fault finding.

Before we look at the two types, let's have a very quick look at a simple example of a WACS configuration file.

Quick Intro To XML

Example 2.1. Simple WACS Configuration file

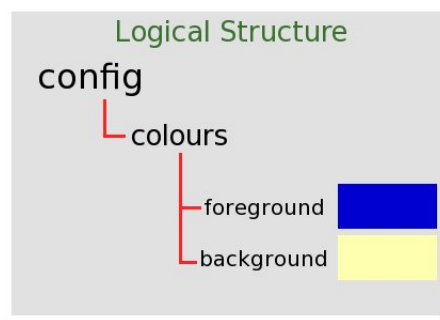
```
<?xml version='1.0' standalone='yes'?>
<config>
  <colours>
    <foreground>#0000D0</foreground>
    <background>#FFFFB0</background>
  </colours>
</config>
```

The above example is actually a valid WACS configuration file which will specify that most of the text is a bright blue and the standard background is a bright yellow. Not probably the most subtle or desirable of colour schemes, but the significant difference in brightness between the two colours used should at least render it very readable.

Let's work through this line by line - the top line is simply a designation that this is an XML file and conforms to version 1.0 of the XML specification. You need it to be there for it to be recognised as XML

but you'll never need to change it. The second line states that we're starting a *config* file, which we do with what's known as an open tag which consists in the word bracketed by less than (<) and greater than (>) symbols. It *absolutely must* be matched by a partner close tag which consists of the same word prefixed by a standard forward slash (/) character still surrounded in a pair of less than/greater than brackets. This means that the <config> must be matched by a </config> at the end. Since it starts first, *config* is considered the outermost element - the diagram below may help you visualise it.

Figure 2.1. Logical Structure



Within the overall *class* of config, we have a second level element called <colours>. In fact, it is one of many such that you will see when looking at WACS XML files. Within the section <colours> we have two attribute elements *foreground* and *background*. Each of these has an attribute, in this case a colour specified in the standard HTML RGB notation - these should be instantly recognisable to most web developers and designers. These attributes are simply enclosed within the open (eg <foreground>) and (eg </foreground>) close tags of the attribute.



Tip

The topic of colour selection is a complex one we can't really cover here. The key points are that the colours you choose shouldn't clash but should be sufficiently different from each other as to make the text stand out from the background enough to be easily readable. A web search for "web safe colours" should give you some useful links - a good starting point would be The Wikipedia Web Colors [http://en.wikipedia.org/wiki/Web_colors] page.

Permanent Configuration

Table 2.1. List of Permanent Config Files

| Filename | Purpose |
|----------|---|
| wacs.cfg | The main WACS configuration file: drives most of the code behaviour, display formatting and layout aspects of the WACS system. This file includes configuration for web address, file systems locations, permissions, colours and layout control. |
| menu.cfg | Provides customised menus for all of the WACS applications - these can either be new menus defined for your own applications, or redefinitions of the provided standard menus to add or remove options. |
| wacs.acl | The access control list for permanent access - this file lists the machines and implied usernames and customisation options for machines with static IP addresses that are granted permanent access without authentication to the WACS system. |

Transient Configuration

Table 2.2. List of Transient Config Files

| Filename | Purpose |
|------------|---|
| leases.acl | This file lists the current and recent leases granted for access to the WACS system - it contains IP address, username, role and preference settings. |



Note

An additional transient configuration file is planned for a future release of WACS to allow for preference and temporary suspension features on permanent access configurations.

Chapter 3. Prerequisites

OS Configuration

There are a number of prerequisites to successful operation of a WACS server, and the first group of these depend on the web server host operating system being correctly configured. Later ones depend on the configuration of the database and NFS file serving sub-system to work.

Static IP Address And Name

For any web-based application to be successfully deployed, one of the most important things is that the webserver be reachable on a reliable URL. This is unfortunately not as straight forward as it sounds as in most other fields of activity, networked computers just work fine if allocated *the next available network address*. This is not the case with web servers. The first step then to getting a host configured to be a web server is to ensure it will always appear as the same name on the same IP (network) address each time it boots up.

In a typical home or small office network, the network router box (ADSL or Cable router) can be configured to do this for you. The first step is to determine the ethernet address of the host concerned; you can do this with the **ifconfig** command when run as root. It is usually best to specify the relevant network interface which will usually be the first ethernet interface, known as `eth0`. The output should look something like this:

Example 3.1. Determining The Ethernet Address

```
# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:11:2F:54:86:D5
          inet addr:192.168.1.10  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::211:2fff:fe54:86d5/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:100576 errors:0 dropped:0 overruns:0 frame:0
          TX packets:79057 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:64722164 (61.7 MiB)  TX bytes:51569420 (49.1 MiB)
          Interrupt:17

#
```

You need the `HWaddr` which in this case is `00:11:2F:54:86:D5`. Log into the administrative interface of your router, and look for either DNS or LAN settings; you should be able to find an option here to reserve an address for a specific machine. You need to pick a name and an IP address to be allocated to it; most home networks use the *Class-C non-routed address range* which means the network numbers start `192.168.` - your router will have probably chosen either `192.168.0.1` or `192.168.1.1` for it's own address. You need to stick to whatever the router has chosen for the first three couplets of numbers as that will be your local network number. Lets assume we want to call the server *nemesis*, then we'd be looking to enter the following into the network number reservation list in the router configuration.

| Configuration Item | Example Value |
|--------------------|-------------------|
| Ethernet Address | 00:11:2F:54:86:D5 |

| Configuration Item | Example Value |
|--------------------|---------------|
| IP Address | 192.168.1.10 |
| Host Name | nemesis |



Warning

Depending upon the ADSL/Cable router design you may also want to specify your internet domain and you may need to modify the automatically allocated address range it uses to avoid conflicts. You'll have to refer to the manual for the router to determine this.

You may additionally want to set the hostname within the operating system itself to the same name, and you might want to fix the address in the server's own configuration to make it come up on that network address even if the ADSL/Cable router is down when it reboots. How this is done varies even between Linux distributions, so you will have to refer to the documentation of your particular distribution on how to do this.

ISP Requests

You will probably also want to ask your Internet Service Provider for a number of specific changes to be made to your account. The minimum you almost certainly want to ask for is "a static IP address" - this means that whenever your ADSL router makes a request for an Internet address to be allocated to it, it will always be given the same one. If you don't want to let on what it's actually for, and remember that a WACS server may well be holding content in violation of their terms and conditions of service, you can say that your employer wants to know a fixed network address to add to their security access list in order to allow you to access their corporate network from home.



Warning

This terms and conditions thing is annoying. If you have your WACS system secured so that only people with passwords you've issued them can access it, your ISP will probably never know and to some extent it's an invasion of privacy for them to dictate to you what you may or may not do with your own system. Do remember that WACS is released in the public domain in the hope that it will be useful, but the license *explicitly* disclaims liability for any damages caused through it's malfunction or use. It's not impossible that some bug in our security code *might* now or at some point in the future expose your collection unintentionally. You have got to understand that that risk is on your head if you make your server accessible to the internet. *Quite apart from this* be wary of throwing it open to general access as people from netnanny services hunting down sources of unsuitable material could end up blacklisting their whole IP address range with serious commercial consequences for them and other mainstream web sites hosted by them. In that situation they'll have a genuine grievance with you and may well come after you with lawyers which is usually pretty nasty...



Tip

While talking to your ISP, it's worth saying to them that you wish to access files on your system from outside, for your personal use only, and that you'd like to know if any protocols or port numbers are blocked at their firewall. Many (unfortunately) block http, but you can always move the actual port number used by your WACS web server to a different one, providing you know that you're going to have to.

The second thing that is useful is a standard domain name for your home network; this can be a domain name you've bought or merely a fixed name within their namespace. If you get a domain name, for instance,

mydomain.info, it's helpful to get them to name the system in the DNS records for that domain, ie nemesis.mydomain.info so that the same hostname works both inside your local network and outside in the general internet. Note that if you only get the static IP address from your ISP, you can buy a domain name through a third party domain name seller and configure that to give the static IP address allocated to you by your ISP when asked for nemesis.mydomain.info. The cost of this can be very low, especially on the lesser domains like .net, .info, etc.

Port Forwarding

If you want your WACS server to be accessible from the internet, you will have to configure your ADSL/Cable router to do port forwarding. For basic web serving, all you need is TCP/IP port 80 (http) open. You do this by going into your routers *NAT Firewall Rules* configuration option and set up port forwarding rules for port 80/HTTP to forward to whatever address you set your server up with; if you did this with the example above, the rule would be:

```
port 80/tcp (http) is forwarded to 192.168.1.10 port 80
```



Note

On many simpler router boxes, the port number can only be forwarded to the same port number on the destination host, so it won't ask you the destination port number. port 80 will be implicit for a port 80 forward.

If your ISP's firewall rules cause you to have to move to a different port, you will need to configure the apache web server to listen on the other port number too (Port 80 is in the standard httpd.conf file, you can add a second line, say Port 8080 straight after it. You will also need to modify the wacs.cfg file's server section to have http://nemesis.mydomain.info:8080/ instead of http://nemesis.mydomain.info/ in each attribute. It does work just fine though...

NFS File Sharing

A network of Linux, Macintosh and other types of Linux/Unix based operating systems can share files with each other using a protocol called Network File System (NFS). WACS is completely compatible with this process so long as the corresponding database is shared as well. It is also perfectly possible to set up WACS to manage a collection housed on a separate network attached storage (NAS) device so long as it supports NFS. With these being available from under 70 euro (US\$100), it makes a very effective way to house your WACS-managed media collection.

To allow another machine to access the WACS collection archive directly from your web server, you need to edit the /etc/exports file on the WACS server system. As with the forced allocation of a fixed IP address for the server system itself, you may wish to do the same for the laptop to ensure that it always receives the same local name and address for security reasons. Assuming you've named your laptop fawn, the /etc/exports file would then read (on a Linux server) something like this:

```
/home/wacs fawn(rw,no_root_squash,async)
```

The nfs server process would then need to be started; on most linux distributions if the file /etc/exports exists and has entries in it, it will be automatically started on boot. It can usually be manually started too with the command /etc/init.d/nfs start. On the client machine, in this example the laptop called fawn, you merely add an extra line into the /etc/fstab file that reads:


```
nemesis:/home/wacs /home/wacs nfs defaults 0 0
```

Of course these commands are subject to various differences between machines based upon dialects of the commands, the actual locations and the specific names of the machines and file systems. Read the appropriate HowTo Documents and manual pages for your operating system version for more details.

Database Configuration

Within a network of machines, you may wish to use some of the WACS tool sets from another machine on the network. In order to use some of the facilities, you will need to make sure that the database server will allow network connections, that a copy of the WACS configuration file is available to the programs you're trying to use on that other machine and that the database client side code has been installed. The table below is a basic checklist on how to get things working - refer to the database documentation for specifics.

Table 3.1. Steps To Get Network Connections To Work

| Database | Task | Description |
|----------------|------|---|
| Oracle 10i/11i | 1 | Start TCP network listener on database server: <code>\$ORACLE_HOME/bin/lsnrctl start</code> - if that doesn't work, check for database entity in <code>\$ORACLE_HOME/network/admin/listener.ora</code> file and try start again. Same command supports <code>stat</code> and <code>stop</code> requests. |
| | 2 | Install Oracle instant client or similar software on the client machine (ie laptop or workstation). Create entry for database server in client's <code>\$ORACLE_HOME/network/admin/tnsnames.ora</code> file. Install the <code>wacs-core</code> module and the <code>perl DBD::Oracle</code> driver. |
| | 3 | Copy the server's <code>wacs.cfg</code> across to the client, changing the locations specified in the <code>fsloc</code> section as necessary for where they are imported on the client system. |
| | 4 | Make sure you can connect to the database using the credentials in the <code>wacs.cfg</code> file using the sqlplus SQL interpreter. |
| MySQL 5 | 1 | Edit the <code>/etc/my.cnf</code> file on the database server and make sure the line <code>bind-address=0.0.0.0</code> is present and has not been commented out - this ensures TCP mode connections are enabled. If you make a change, restart the MySQL server. |
| | 2 | Install the WACS software on the client machine - if using packages this will bring in all the necessary runtime modules. |
| | 3 | Copy the server's <code>wacs.cfg</code> across to the client, changing the locations specified in the <code>fsloc</code> section as necessary for where they are imported on the client system. Additionally change all references to <code>localhost</code> to the servers name - mostly these will be in the database section. |
| | 4 | Make sure you can connect to the database using the MySQL root user using the mysql SQL interpreter. Then determine the <code>wacs</code> database credentials from <code>wacs.cfg</code> and create and then use them as per the notes below. |

With the likes of Oracle 10g, that is all that is necessary. For MySQL, you additionally have to tell the MySQL server that the `wacs` account is allowed to access the database from that location, so you would need to do the following steps on the server to allow that to happen:

```
nemesis% mysql -u root -p
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 USER 'wacs'@'fawn.mydomain.info'
-> IDENTIFIED BY 'wacs';
Query OK, 0 rows affected (0.00 sec)

mysql> CREATE USER 'wacs'@'fawn'
-> IDENTIFIED BY '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
nemesis%
```

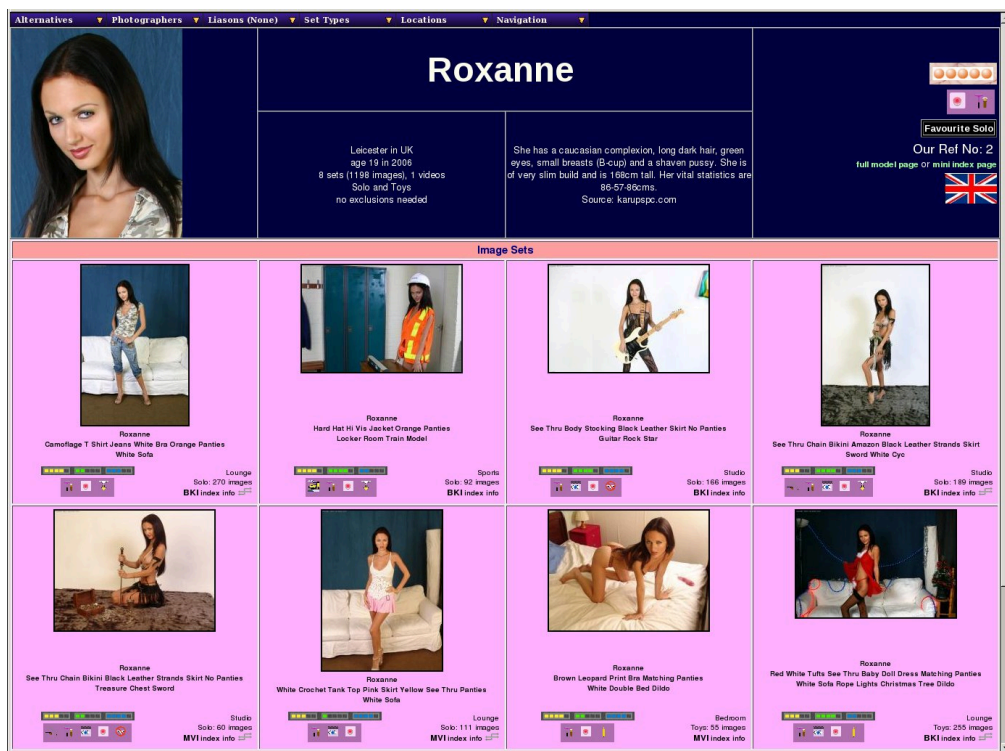
You will also need to make sure that the actual hostname, eg nemesis in our examples, is used in the wacs.cfg file rather than localhost. While localhost will work fine on nemesis itself, fawn won't work with that.

Chapter 4. Altering The Configuration

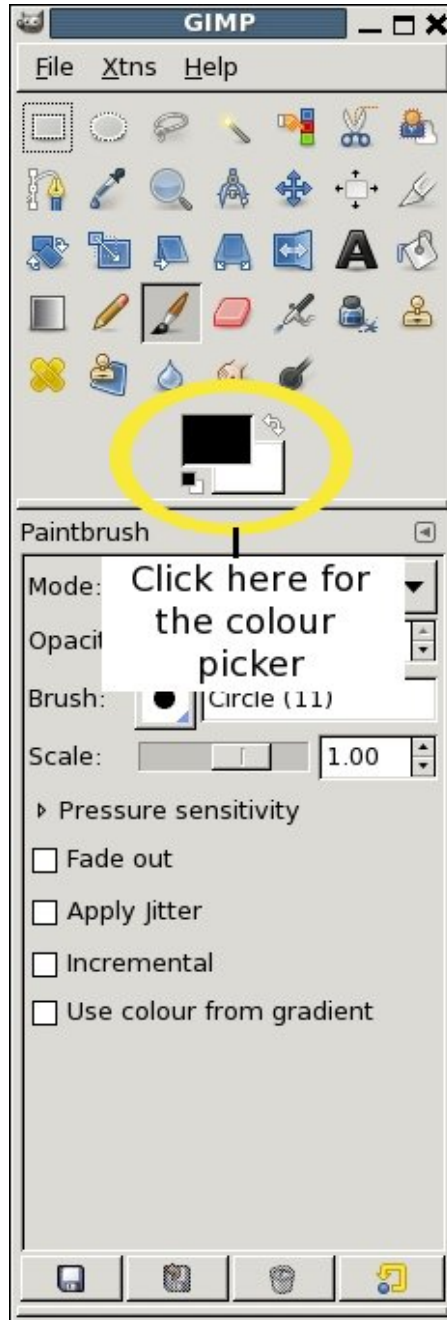
Altering: An Introduction

After the fairly dry material about configuring the server and internet connection, we're going to look at a few practical examples of how you can alter the configuration of the WACS system to better reflect your needs, desires or even whims. We're going to look at how you would set about changing the colours, the page layouts and the menus. In all cases, you need to know where the `wacs.cfg` file for the site you wish to modify is, have write access to it and a suitable text or XML editor available to you. Our examples are going to show you the raw XML file, but in a specialist custom XML editor you would merely expand the trees to reach the same attribute and alter it there.

Changing Colours

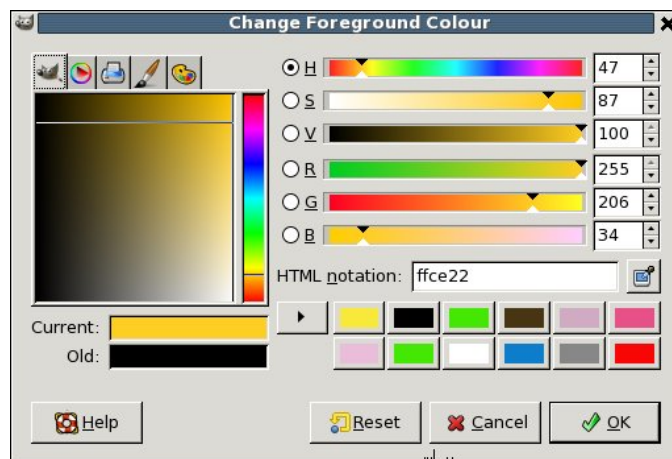


For the first worked example of making a configuration change, we're going to assume that you don't like our rich pink backgrounds and would prefer something in orange with dark red text. (Actually, you'll probably want more pastel shades, but we'll leave you to your own perversions - we like bright colours). The first step is to select the colours you'd like and for this some kind of colour picker is definitely useful - the excellent Gnu Image Manipulation Program (Gimp) has a very suitable one of these. To invoke the Gimp, either go to your Applications -> Graphics menu or type: **gimp** at a command line prompt.



Tip

Do remember to make sure that the foreground and background colours you pick have a good level of contrast between them to ensure readability.



Once you've got the colour picker up, you can use the vertical rainbow column to choose the general colour you're looking for and then the larger area to the left of it to choose the exact level of saturation. Here we've picked a modest orange (!) and if you look over to the right, we're shown its hexadecimal RGB value (aka web colour) which is **ffce22**. We then do the same procedure and pick a nice dark red, which in this case is **720a1d**. Armed with these values, we need to just capitalise the letters, add a hash and we have the web colours - **#FFCE22** for the background, and **#720A1D** for the foreground.

We then go to our `wacs.cfg` file and find the section labelled *colours*. This will look something like this:

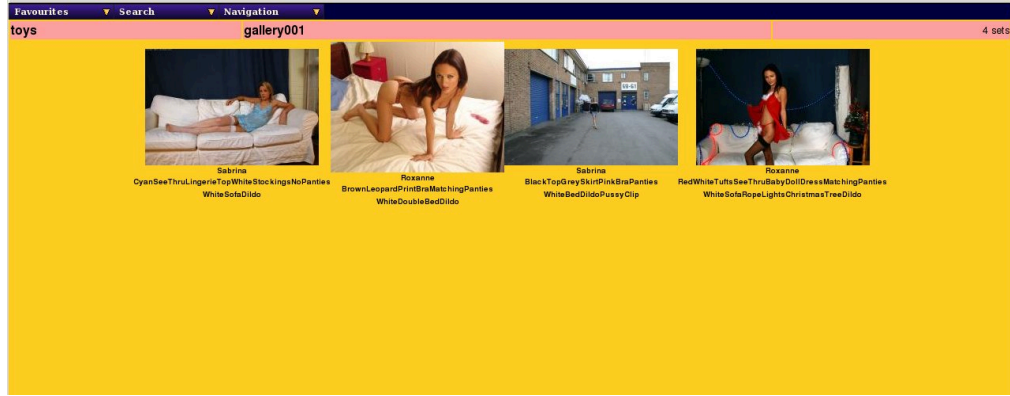
```
<colours>
<foreground>#000000</foreground>
<background>#FFE8FF</background>
<setbackground>#FFB0FF</setbackground>
<vidbackground>#FFFFB0</vidbackground>
<dvdbackground>#B0FFB0</dvdbackground>
<divider>#FFA0A0</divider>
<mastheadforeground>#F0D8F0</mastheadforeground>
<mastheadhighlight>#FFA1FF</mastheadhighlight>
<mastheadbackground>#000040</mastheadbackground>
<indexcolours>Y</indexcolours>
</colours>
```

The two entries here we are going to change are **foreground** and **background** with the background being the orange and the foreground (text colour) being the dark red. The new entries will be:

```
<foreground>#720A1D</foreground>
<background>#FFCE22</background>
```

Once we've made this change and write the file, the change should be immediate with the new colours displayed in all their glory! And in fact, here's a sample screenshot with the new values:

Altering The Configuration



Bet that got you reaching for the sunglasses! Still, the red text is actually working quite nicely, but we probably need to tone down that orange a little bit. Still, the point is you can change it. Do remember that many of the headings and banners on pages are using the masthead colours instead and you might wish to make those complimentary, but possibly reversed to those shown here. In fact, you could use exactly the same values in reverse, which results in a page something like this:

```
<mastheadforeground>#720A1D</mastheadforeground>  
<mastheadbackground>#FFCE22</mastheadbackground>
```



Actually I think the orange looks better as a text colour on the darker red, but there you go. As with all things, some experimentation is in order if you want to find a pleasing combination. Hopefully this has given you some idea of how to control the colours in WACS.

Chapter 5. Caching And Performance Tuning

Objectives

There are two methods provided by the Wacs system to improve performance on a heavily used Wacs system. One of these is a simple strategy of caching the most frequently used icons, and the other is bypassing a significant amount of the security mechanism to improve the speed of loading of images.

In this chapter we will look at how these two approaches differ and why you might wish to select to use either one or both of these methods. They are not mutually exclusive but there are constraints on when and how it may be useful to use both methods.



Warning

Both of these methods are newly introduced and may well still have significant bugs and issues - proceed with caution and test thoroughly before using in production servers.

The usedirect Mechanism

The **usedirect** mechanism requires a bit of a rethink about how the filesystem on the web server is laid out. It offers some very distinct speed advantages on loading of web pages, but at the same time has some significant disadvantages. The default configuration for Wacs is set so that it is not used. When usedirect is used, the modelpages and others give an actual URL for the icon for each set rather than giving a link making reference to the wacsimg program. This means that the icon is obtained directly from the webserver without any intermediary application being called and will conventionally be cached and that cache utilised where possible.

However doing this moves the set icon outside of the access control afforded by Wacs, so technically it is unprotected - if you're using the security mechanism to restrict content from spouses or minors, this is really bad. If you're using the security mechanism in a subscription web server, people being able to get to a single 120 x 120 pixel thumbnail of each of the sets you have, but not the set itself, is probably of no concern at all. Do bear in mind however that if no web page contains a reference to it, people are unlikely to find the thumbnail's full URL unless they have access to your web browsers history, cache or a corporate firewall log. To avoid being too obvious in such histories or firewall logs wacs normally invokes icons via a fairly anonymous call to wacsimg with a set number as an argument - this would not normally trigger any automatic corporate firewall content flagging systems and such a URL is only valid from the authenticated host itself during the time period of the lease. The Logout option allows for the voluntary relinquishment of a lease ahead of it's expiry time, making this even safer. Other addresses would not be able to display it, nor would the URL be any use after the expiry of the lease. When usedirect is used, the name is obvious, permanent and accessible - it might well trigger corporate firewall content flaggers.

WACS can be simultaneously used in both ways - you could use usedirect when at home, and normal access methods when elsewhere. The consequences of using it elsewhere do of course fall on your head - the above comments about corporate firewalls are informational only and in time, who knows, some rulesets may be updated to specifically detect wacs usage.

To enable usedirect operation, you need to ensure that the fsloc section configuration options **seticons**, **officons**, **vidicons**, **dvdicons** and **addicons** all point to locations within the web server document tree (ie `/var/www/html` or similar). The generate command can then be used to re-create the auto-generated

icons under **seticons** if required; the others will have to be moved (or copied) from their previous locations. From this point on, the icons should someone know the URL by some means, will be accessible to all (unless you restrict them again through the web servers access control mechanism). To enable usedirect functionality, you then add the URL of the web server icon directory to the configuration file using the **usedirect** keyword.

Using The Scaled Cache

While the name is slightly misleading and came from it's previous usage to store only rescaled versions of the icons, the **scalecache** directive specifies a directory in which to place re-scaled versions of the set icons. Each time a specific sized version of an icon is requested, it will be created if needed and copied into a directory tree placed in the directory pointed to by the **scalecache** configuration variable if that directory exists. If **scalecache** is not set, it will be retrieved from the usual icon locations and scaled if necessary. The gain here is that the scale cache is organised by set number rather than name, so no database interactions are required to determine if the cached icon is available - the **wacsimg** command is still used but it runs for a much shorter time and does much less.

The way this works is that if **wacsimg** is asked for the standard icon for set 14, it simply checks if there is an icon for set 14 at the standard size in the scale cache. If it isn't there, or **scalecache** is not enabled, it connects to the database, looks up set 14's location and goes and looks for the icon in that location. The main issue with **scalecache** is that set numbers could potentially "go stale" and it might produce the wrong icon. Additionally, it can take up quite a bit of space. Best practice would probably be to delete or at least expire entries from it over a period of a week or so. This will ensure that active sets, like the newly added ones, are cached but that the cache itself is not swamped by unused and obsolete cached icons.

Chapter 6. Custom Configurations

Introduction



Note

This chapter replaces the previous application notes which describe how to do certain specialised configuration tricks in WACS. Unless you're trying to use the old vendor-based layout style or running multiple sites on the same web server, you probably don't need to read this chapter!

Vendor Style Menus in Wacs 0.8.x and later

Background Info

One of the big new features in Wacs 0.8.0 was the introduction of the configurable menu system. With this we did an extensive review of how our existing menus were constructed and decided to remove the old separation between those images housed in a dedicated area for each upstream source and those housed in more general areas. This supplier-identified versus content type based architecture may have suited our original development environment but we don't think it really reflects how most people would visualise and use folders in a normal WACS installation.

So the new default layout in 0.8.0 gets rid of the vendor area distinction completely. However if existing Wacs users who adopted the old vendor based architecture want to preserve it, or others have some special reason for wanting it organised that way, there are still facilities to keep it the way it was. This document takes you through the changes needed to reproduce the old layout using the 0.8.0 release.

How To Configure Vendor Mode

Step 1: Enable vendormode in the configuration file

The first thing you need to do is to enable `vendormode` in the configuration file. By default in Wacs 0.8.0 and later, this is set to `N` (for No). If no setting is specified, it will default to `No`. To change this you need to edit the global Wacs configuration file which is usually to be found in `/etc/wacs.d` and called `wacs.cfg`. You should be able to edit this using whatever your preferred text editor is.

Look through the file and see if there is a section called `layout` as this is where the `vendormode` attribute needs to be. If there is already a `vendormode` entry, make sure it is set to `Y` (the upper case letter 'y', not Yes). It should now read:

```
<vendormode>Y</vendormode>
```

If there is not already a section called `layout`, you will need to add one (this is the most likely case). You do not need to worry about creating a new section as any variable will always be defaulted if it isn't explicitly mentioned. The complete new section entry should now read:

```
<layout>  
<vendormode>Y</vendormode>
```

</layout>

Step 2: Adding The Menu Entries

The next step is to add custom menus for the second and third pull down menus on the Wacs Front Page (**wacsfp**) which is the same location for these menus as it was in the earlier releases. These are known as **fpmenu2** and **fpmenu3** respectively. Due to the way this feature works it's *STRONGLY* recommended that you use these two menus as is. However the actual requirement is that the general (ie non-vendor) menu entries be on either menu **fpmenu1** or **fpmenu2**, and that the vendor based ones be on menu **fpmenu3** and higher (to **fpmenu7**). The values of the variables **isarea** and **vsarea** change at **fpmenu3** when **vendormode** is enabled.

The menus are controlled by the `menu.cfg` file which is to be found in the same location as the `wacs.cfg`, usually `/etc/wacs.d`. Any menu entry defined in here will be the complete definition of that menu; any menu not defined will take the default definition. If the `menu.cfg` file already exists you need to leave out the top (XML identification) line and the second and final lines that define this as a menu definition file (the `<menu>` and `</menu>` entries). You obviously also need to check that menus **fpmenu2** and **fpmenu3** are not already defined within this file. The resulting `menu.cfg` should contain the following:

```
<?xml version='1.0' standalone='yes'?>
<menu>
  <fpmenu2>
    <menutitle>Sections: General</menutitle>
    <default>#CGIURL#wacsimgcats</default>
    <entry>
      <type>foreach</type>
      <variable>isarea</variable>
      <title>#OPTDESC# - images</title>
      <target>#CGIURL#wacsshow/sarea=#OPTIONS#</target>
    </entry>
    <entry>
      <type>foreach</type>
      <variable>vsarea</variable>
      <title>#OPTDESC# - videos</title>
      <target>#CGIURL#wacsvidshow/sarea=#OPTIONS#</target>
    </entry>
  </fpmenu2>
  <fpmenu3>
    <menutitle>Sections: Vendors</menutitle>
    <default>#CGIURL#wacsvidcats</default>
    <entry>
      <type>foreach</type>
      <variable>isarea</variable>
      <title>#OPTDESC# - images</title>
      <target>#CGIURL#wacsshow/sarea=#OPTIONS#</target>
    </entry>
    <entry>
      <type>foreach</type>
      <variable>vsarea</variable>
      <title>#OPTDESC# - videos</title>
      <target>#CGIURL#wacsvidshow/sarea=#OPTIONS#</target>
    </entry>
  </fpmenu3>
</menu>
```

```
</fpmenu3>  
</menu>
```

Once this file is in place, log in to Wacs and try out the menus - the separate collections-versus-category menus should work as before. Please do be aware however that the backwards compatibility features may not remain in the Wacs system permanently.

Running Multiple WACS instances

Background

One of the questions we have been asked about WACS is whether it is possible to run multiple separate instances of WACS on a single server. The answer is yes, and this document sets out how you configure this to work. The one limitation at present is that all the instances need to run the same release of Wacs because the standard WACS modules (that do all the hard work) are installed globally on the host server.

How To Configure Multiple WACS Instances

Step 1: Add a SetEnv Directive To Your httpd.conf file

The first step is to alter your Apache 2 `httpd.conf` file: you need to edit the section that contains the virtual server definition for the machine that you wish to run a new virtual instance on. Alter the virtual server definition to include the following extra configuration line:

```
SetEnv          WACS_CONFIG /vol/www/data/beaky-wacs/wacs.d
```

The `httpd.conf` file is usually to be found in `/etc/httpd/conf` (`/etc/apache2/conf` on Ubuntu) and is expected to already contain a number of `<VirtualHost>` sections. Conventionally the `VirtualHost` definitions are near the end of the file. Find the one for the new server (second instance or greater) and add the line shown above, modified to the location you actually wish to store the new virtual host WACS configuration in.

Step 2: Creating The Virtual Site Configuration File

Each WACS installation needs it's own configuration file pointing to the document and icon trees for that installation. This configuration file therefore needs to be customised for each virtual site created. The best starting point is probably to copy either the default system file created by installing the packages or using `easyinstall` in `/etc/wacs.d/wacs.cfg` or by copying the example file from the `conf` subdirectory of the WACS distribution.



Tip

If you've installed using a package install, the standard wacs infrastructure components (glyphs, style files, javascript scripts) will appear in `/wacs` on every virtual server. If you don't want to make use of these in one of the virtual servers, you can set it to use a private tree through the `wacsurl` directive in the `wacs.cfg` file. See Chapter 9, *Server Details* for more information.

Make sure you have created the directory named in the Apache 2 `httpd.conf` file (see above) and copy this new file into it. Then work through the settings given in the `wacs.cfg` file and modify them as necessary. Note that at a very minimum you will want different database login credentials and your image

archive tree must be different. Details of what these settings are and what they should be set to can be found in other chapters of this configuration guide.

Step 3: Adding The Menu File

The way that WACS works, any configuration file not found in the specified location will also be checked for in the standard locations, so when trying to have multiple virtual hosts, you have to make sure that the key files do exist in the specified configuration directory. This means if you don't create a `menu.cfg` file for each configuration, the default one in `/etc/wacs.d/wacs.cfg`, which may be totally inappropriate to that virtual site, will be used. In other circumstances, you may be able to use this to your advantage, but you need to be aware of it. At a very minimum, creating your new `menu.cfg` file as:

```
<?xml version='1.0' standalone='yes'?>
<menu>
</menu>
```

This will ensure that the virtual site has standard WACS menus.

Step 4: The Access Control List

As with the menu configuration file, the static access control list `/etc/wacs.d/wacs.acl` will also be common across all virtual hosts unless you create a local one in the specified new config location. This is more commonly what you might actually want because it allows your system support and help desk staff the same level of automatic access to all virtual servers hosted on the system. If you do want to override this, you again create a local one within the configuration file area for the virtual host in question.

Step 5: Creating The Virtual Host Database

Since each installation has its own database, you will obviously need to create a new database table for each instance, and to do this you will need to follow the manual installation procedure, detailed in the WACS installation guide from step 9 (Creating the database). Normally you would create a different database account for each instance and create the necessary database tables within that account and then populate them.

When using Oracle it is possible to have a single models database instance and access it via a Named View from each virtual server using an `idmap` to determine if a given model should appear in a given virtual server's catalogue. Through the ability to specify the database schema name to use in the config file, you can also import certain tables from a central repository account using the `account.schema` notation. If you didn't understand the last two sentences, don't worry, they're pretty complex relational database concepts and you almost certainly won't need to use them! If you did, you'll hopefully be beginning to grasp that WACS is fully capable of giving service in an enterprise level environment.

Step 6: Populating The Virtual WACS Server

Once the steps above have been completed you can then populate the server using either the web based tools, or the command line based ones. Do note that if you're using the command line tools, including the `wacsimport` and `wacsxmlin` tools, you will need to specify the environment variable `WACS_CONFIG` to ensure that the tools are working on the correct installation. By default, it'll always go to the one configured in `/etc/wacs.d/wacs.cfg`. As with the Apache configuration, setting the environment variable changes the environment in which the tools act. The commands would be along the lines of:

- `sh/bash shells:`

```
export WACS_CONFIG=/vol/www/data/beaky-wacs/wacs.d
```

- csh/tcsh shells:

```
setenv WACS_CONFIG /vol/www/data/beaky-wacs/wacs.d
```

With this done the tools will then act on the file tree and database of the desired virtual server.

Part II. WACS Configuration: Reference

This section of the WACS Configuration Guide aims to document the configuration options available and the legal values and uses for them.

Main WACS Config: wacs.cfg

Chapter 7, *Database Environment*

Chapter 8, *Database Tables*

Chapter 9, *Server Details*

Chapter 10, *FSLoc: File System Locations*

Chapter 11, *Security*

Chapter 12, *Download*

Chapter 13, *Colours*

Chapter 14, *Layout*

Chapter 15, *Precedence*

Chapter 16, *Debug*

Other WACS Config Files

Chapter 17, *WACS Configuration: Menus*

WACS Web Applications

Chapter 18, *WACS Web Apps: Presentation*

Chapter 19, *WACS Web Apps: Indices*

Chapter 20, *WACS Web Apps: Models*

Chapter 21, *WACS Web Apps: Tag*

Chapter 22, *WACS Web Apps: Search*

Chapter 23, *WACS Web Infrastructure: Retrieval*

Chapter 24, *WACS Web Infrastructure: Tags*

Chapter 7. Database Environment

Overview: Database

Wacs makes very extensive use of a relational database backend in organising and storing data about its collection of images and video clips. Its primary method of access to this data is via the perl generic database access driver, **DBI**, and its respective lower layer drivers for specific databases such as **DBD::Oracle** for Oracle 10 and **DBD::MySQL** for MySQL 5. The matching routines for PHP, from the PEAR collection, are used to provide the same functionality for WacsPHP apps.

At no point in the code is a different SQL request made upon the basis of the underlying database being used, so the code is theoretically portable to any other relational database with an appropriately complete implementation of SQL. That said, without making additional arrangements with the authors, only MySQL and Oracle are supported and routinely tested. If porting to an alternative database platform, the most likely issues are to do with the representations of dates within the system.

wacs.cfg: Database Environment Section

In order to correctly communicate with the database engine, whichever one that may be, a number of parameters have to be provided to the underlying mechanism. These include the database account name to use, the password associated with that account, the network location of the database server and any additional support environment variables that are needed to make it work. Normally those free databases (such as MySQL) that are shipped with the operating system itself are pre-configured to know the location of all their component parts. For commercial packages like ORACLE, this is not so clear cut as on different distributions, the Oracle applications and libraries might be in `/usr/local/oracle`, or `/opt/oracle` or any number of installation-specific locations. In these cases the database routines include an environment variable, in the case of Oracle called simply `ORACLE_HOME`, to locate where the necessary components are within the filesystem. WACS includes the ability to establish any one arbitrary environment variable with a suitable value before the database routines are invoked.

The table below details the configuration elements provided by the WACS configuration system and how to use them:

Table 7.1. Configuration: Database Section

| Variable | Description | Typical Value |
|--------------------|--|--|
| dbienvvar | Commerical databases such as Oracle 10g usually require the presence of an environment variable to tell their libraries where to find configuration information, libraries, etc. This is the name of that variable; the companion attribute, <code>dbienvvalue</code> , is what value that variable should have. | <code>ORACLE_HOME</code> |
| dbienvvalue | The value that <code>dbienvvar</code> should be set to; typically the location of the installed database library and configuration files for the database. | <code>/usr/local/oracle/V10.1.0</code> |
| dbiconnect | This is the connect string that needs to be passed to the perl DBI routines to specify the database type and instance to be connected to. For Oracle it'll be | <code>dbi:Oracle:orasrv</code> |

| Variable | Description | Typical Value |
|---------------------|--|---|
| | something like <code>dbi:Oracle:orasrv</code> where <code>orasrv</code> is the SID of the database. For MySQL it'll be something like <code>dbi:mysql:wacs:mysqlserver</code> where <code>wacs</code> is the database name and <code>mysqlserver</code> is the name of you database host. | |
| phpdbconnect | This is the connect string that needs to be passed to the PHP DB routines (php5) to specify the database type and instance to be connected to. For Oracle it'll be something like <code>oci8://wacs:wacs@orahost/orasrv1</code> where <code>orahost</code> is the host name of the Oracle server and <code>orasrv1</code> is it's instance name. For MySQL it'll be something like <code>mysql://wacs:wacs@mysqlhost/wacs</code> where the first <code>wacs</code> is the username, the second the password, then the <code>@</code> symbol followed by the hostname the database runs on and the finally the database name. | <code>mysql://wacs:wacs@mysqlhost/wacs</code> |
| dateformat | This is the format of date used by the underlying database - initially this will be either <code>DD-MON-YYYY</code> (the Oracle format) or <code>YYYY-MM-DD</code> (the MySQL format). | <code>YYYY-MM-DD</code> |
| dbname | The database name - <i>Platform Dependant</i> . For MySQL 5.x, this is the database name; for Oracle 10 and higher this is the server name as specified in the OCI connect string. | |
| dbuser | The username to log into the database with. | <code>wacs</code> |
| dbpass | The password to use in conjunction with the account name given in given in dbuser above to log into the database with. | <code>wacs</code> |

Database Engine Notes

Database: MySQL 5

The current development environment for the MySQL version of the WACS system is based upon MySQL release 5.0.45 as provided with Fedora 8 Linux. Some of the more advanced features in Wacs require nested selects to be supported, a feature that was introduced to MySQL after the conventional 4.x series releases. Anything with version 5 should be OK and it's possible that it will work but with some limitations on version 4.

Database: Oracle

The current development environment for the Oracle version of the WACS system is based on the x86_64 version V11.1.0.6, but retaining support for Oracle 10; the upgrade to Oracle 11 happened at Wacs release 0.8.4. We are not aware of any features that Wacs requires that don't work in Oracle 9, 10 or 11. There might be some problems on Oracle 8 or older due to issues on getting table structure meta-data for the perl DBD driver.

Chapter 8. Database Tables

Overview: Tables

This section of the configuration deals with the name of each table within the database. By specifying this in the configuration file, it is possible for us to do a number of things to improve flexibility and security. For instance, it becomes possible for certain tables to be owned by a user other than the account the WACS database layer logs in as. This is useful in controlling what may be done to records in the database and by whom.

wacs.cfg: Database Section

Table 8.1. wacs.cfg: Tables

| Variable | Description | Typical Value |
|---------------------|--|---------------------------|
| models | name of the models tables. This is indirected to allow for databases to be owned by a different user than the wacs system runs as or for porting considerations to other database platforms. In Oracle you might call it <code>dbowner.models</code> rather than the normal simple <code>models</code> . | <code>models</code> |
| idmap | name of the idmapping table - this is the one that links a local model to a model name/number for a specific site. | <code>idmap</code> |
| assoc | name of the associations table, ie the one that links a model to a given set. A single set can have many associations, if it has a lot of known models in it. | <code>assoc</code> |
| sets | name of the sets table - the <i>main</i> database table of the WACS system. Very little will work if the wacs programs can't find this one! | <code>sets</code> |
| download | name of the download table. This is used to identify the source information for a set. Depending on status can either be a task list, or a check list of what we already have. | <code>download</code> |
| photographer | name of the photographer table. Where the list of known photographers is. | <code>photographer</code> |
| tag | name of the tag table. This is the search results and bookmarking mechanism's workspace. | <code>tag</code> |
| vendor | name of the vendor table. This contains information on various source web sites, how to parse them and download from them and how to link to them. | <code>vendor</code> |
| conn | name of the connections table. This is the new mechanism for arbitrary connections between sets, models and photographers - the key difference between tags and connections, is that the WACS applications will flag up when a set is a member of a connection. | <code>conn</code> |
| keyword | name of the keywords table. This is used for guessing attributes and locations from key words found in the description | <code>keyword</code> |

Chapter 9. Server Details

Overview: Server

The server section contains URLs for your web server. Note that *EVERY URL* stem given here is a directory and includes the final forward slash (/). The programs do make the assumption they only need to append the program/file name to these URL stems.

wacs.cfg: Server Section

Table 9.1. Configuration: Server

| Variable | Description | Typical Value |
|------------------|---|--|
| baseurl | The "base" URL of your web server, ie the top level page | http:// www.example.com/ |
| cgiurl | The URL of your cgi-bin directory on your web server. If you intend to use subdirectories for Wacs within the cgi-bin directory, include that here. | http:// www.example.com/cgi- bin/ |
| iconurl | This where the attribute and rating icons are to be found on your web server. These are the glyphs and artwork components. Not to be confused with the model "headshot" icons, which are expected to be in the icons sub-directory of the Wacs document tree. | http:// www.example.com/wacs/ glyphs/ |
| siteurl | This is the URL for the site-specific web accessible files go - this will include model headshots, additional icons and any other site specific help files. In many cases this may be set to the same as wacsurl below, but it should be changed if you either want to move your local web components out of the /usr/share/wacs area or are hosting multiple sites on the same server. | http:// www.example.com/ mysite/ |
| wacsurl | This is the URL for the Wacs document and stylesheet tree itself, ie where the htmlbones index.php and wacs css stylesheets go. | http:// www.example.com/wacs/ |
| directurl | This is the URL for getting to the set icons <i>*ONLY*</i> if the <code>usedirect</code> option is to be supported. <code>Usedirect</code> causes the web pages generated by wacs applications to include an actual URL for the set icons rather than a reference to the wacsimg cgi program. | http:// www.example.com/wacs/ iconcache/ |
| wacsmain | This is the URL for the Wacs main menu which would typically be either a custom front page, for instance <code>index.php</code> in the wacsdirectory or the Wacs front page program wacsfp . NB: this is a complete URL and does not have to end in a slash for appending options. | http:// www.example.com/cgi- bin/wacsfp |

| Variable | Description | Typical Value |
|-----------------|--|--|
| wacsallm | This is the URL that the All Models link should go to. On a site with a small number of models defined, this will probably be just wacsmodelthumbs on it's own; as more models are added, that may just take too long and you may want to set it to be an Alphabetic search like wacsmodelthumbs/A=A or some other criteria. | http:// www.example.com/cgi- bin/wacsmodelthumbs/ A=A |

Chapter 10. FSLoc: File System Locations

Overview: FSLoc

These values direct Wacs as to where to find the files it should be serving out. Note that in a normal setup these would not be within the web server tree at all, thus making Wacs the gateway to web access of them. Accessing them directly rather than through Wacs could mean loss of all access and login control, and that the URLs would contain more obvious paths giving a better clue to their nature.

wacs.cfg: fsloc section

Table 10.1. FSLoc: File Storage Locations

| Variable | Description | Typical Value |
|----------|---|-------------------|
| images | Location within the filesystem of the top of the images tree from which Wacs is to operate. | /home/wacs/images |
| videos | Location within the filesystem of the top of the videos tree. | /home/wacs/videos |

Table 10.2. FSLoc: Icon & Glyphs

| Variable | Description | Typical Value |
|----------|---|-----------------------------|
| glyphs | Location of the glyphs (icons and artwork) within the file system. | /var/www/html/wacs/glyphs |
| seticons | Where the small icons for each set generated by generate are to be placed, and thus where the wacs programs can fetch them from. | /home/wacs/data/iconcache |
| officons | Location where the official icons provided by the source site are stored; these are not necessarily even close to the thumbnails we generate from the image sets themselves and we may well not have them for all sets, depending upon where the sets came from. These will only be used if their location is specified in the sofficon fields in the sets table. | /home/wacs/data/seticons |
| vidicons | Where the icons for video files can be found. | /home/wacs/data/vidicons |
| dvdicons | Where the icons for DVD covers can be found (for a future project where Wacs knows about appearances by your favourite models on DVDs in your collection). | /home/wacs/data/dvdicons |
| addicons | Where the additional icons can be found - this is prepended to the value of saddicons from the sets table if it is set and the directory pointed to is assumed to contain only additional icons for this set. | /var/www/html/wacs/addicons |

| Variable | Description | Typical Value |
|-------------|---|-----------------------------|
| modicons | Where the normal-sized headshot icons for the models are to be found. These are usually around 120 x 156 pixel jpeg images. | /var/www/html/wacs/icons |
| modbigicons | Certain Wacs applications (like wacsmodelpage) will use a larger icon for the model if present, these are typically somewhere around 260 x 330 pixel jpeg images but can vary quite considerably. This is the location of these larger icons. | /var/www/html/wacs/bigicons |

Table 10.3. FSLoc: Applications & Documents

| Variable | Description | Typical Value |
|------------|---|-------------------------|
| thumbcache | When wacsthumbs creates a set of thumbnail images for a set, these are cached to speed display on a future occasion. This is the location where this cache is stored. | /home/wacs/cache/thumbs |
| minicache | When wacsthumbs creates a set of miniature thumbnail images for a set, these are cached to speed display on a future occasion. This is the location where this cache for these extra-small version thumbnails are stored. | /home/wacs/cache/thumbs |
| scalecache | Where rescaled versions of the various set icons are cached after the resized version has been created. This is the location where the resized versions are stored. | /home/wacs/cache/scaled |
| download | Specifies the directory into which new downloads are to be placed. This is used by the automatic download tools. | /home/wacs/download |
| archive | Specifies the directory into which the downloaded zip files are placed after they've been unpacked. You are expected to either write them off to removable media (CD-R, DVD-R, etc) or just delete them. If it is set to a single hyphen "-", Archive will not be offered as an option in the unpack manager. | /home/wacs/archive/zips |
| leasedir | This specifies the directory in which the leases file will be created. If this does not exist, the lease file (name as specified below in the security section) will be created in /tmp. | /var/run/wacs |
| tooldir | This is the location where the command line tools like refresh, chkmodel, etc can be found. | /usr/local/bin |
| imgtooldir | This is the location of the NetPBM tools like pnmtjpeg . If left blank (the default), the standard search path will be used. | /usr/bin/Pbm |
| cgibindir | This is the location of the cgi-bin directory, ie where all the wacs cgi programs should be installed. | /var/www/cgi-bin |
| htmldir | This the location of the wacs HTML documents tree, and is used for installing glyphs, etc. | /var/www/html/wacs |
| wacsdir | This is the location of the Wacs resources directory; in packaged installations and some others, many of the above will be in sub-directories of this one. The initial use for this attribute is for the wacssetup command to be able to locate | /usr/share/wacs |

| Variable | Description | Typical Value |
|-----------------|--|----------------------|
| | the creation directory but it was made generic to allow for other uses in future. New in Wacs 0.8.4. | |

Chapter 11. Security

Overview: Security

This section contains the attributes used by the security mechanism to determine whether a given user should be allowed to log in and if so, how long their lease should last for.

wacs.cfg: security

Table 11.1. Security Section: wacs.cfg

| Variable | Description | Typical Value |
|-------------|--|---------------|
| authmode | How to attempt authentication: initially this is "host" which means use whatever the web server uses itself but there will be a future extension to allow use of other authentication sources such as a database table of users. | host |
| barredusers | A comma separated list of those valid users of the server system who are not allowed to log into WACS. Most likely this might be the kids or possibly spouses who would take exception to the material - the consequences of putting people in this list are on your head! | wilma,barney |
| powerusers | A list of people who should be granted power user status when they login through wacslogin. This enables manipulation of tag lists and in due course, ratings, etc. | dino |
| adminusers | The highest grade of access - administrators allowed to do anything in the Wacs system tools. | fred,root |
| admingroup | The Unix group owner of the wacs image tree - admin users as listed above, plus the web server owner (www or apache) need to all be members of this group. | wacs |
| leasetime | The period in hours (number with an h after it) or minutes (number with an m after it) that a lease should be granted for to a given IP address; this could be fairly long lived if it's mainly a friends ADSL line which may change periodically (say 24 or 48), or fairly short if you worry about a subsequent user on the same PC as you were surfing from (say 1 or 2). | 24h |
| leasefile | The name of the leases file. This will be in the location specified by the leasedir option in the fsloc (file system locations) section, unless that doesn't exist, in which case it'll be created in /tmp. This file is in the same format as the permanent access control list in the Wacs configuration directory. | leases.acl |

Chapter 12. Download

Overview: Download

The download section covers parameters for the automatic download tools, such as refresh and chkmodel.

wacs.cfg: Download

Table 12.1. Download Section: wacs.cfg

| Variable | Description | Typical Value |
|------------|---|---------------|
| actithresh | Recheck threshold for models marked as active. This controls how long the refresh command will wait between checks, given in days. Even if set to zero, it will not recheck models already checked today. | 2 |
| dormthresh | Recheck threshold for models marked as dormant. This controls how long the refresh command will wait between checks, given in days. Even if set to zero, it will not recheck models already checked today. | 5 |
| idlethresh | The number of days of no new sets before a model will be marked as dormant by refresh. Obviously you want to keep checking for a while, but after say three months (90), it's unlikely that the website has more sets of this model that they're merely yet to release. Of course the model can return to active when she does a new photoshoot for a site, and a periodic check of the dormant models should pick up on that fact. The moment a new set appears, the model is marked back as active automatically. | 90 |
| checkcount | The number of models to check each time that the refresh command is run. This should be tuned to ensure that a nightly check doesn't revisit the same model too frequently, say once every two weeks or so. | 10 |

Chapter 13. Colours

Overview: Colours

The colours section gives Wacs some general hints on which colours to use when. Not all of the Wacs programs yet pay attention to these settings but it is a future direction that they should! And yes, I know it's not spelt the American way, but hey the language is called English, right? So you get to learn to spell it the correct English way for a change! After all, we have to spell it the American way in every web page we write.

wacs.cfg: Colours Section

Table 13.1. Colours Section: wacs.cfg

| Variable | Description | Typical Value |
|--------------------|---|---------------|
| foreground | The foreground colour of web pages in HTML standard six-octet notation. | #000000 |
| background | The background colour of web pages in HTML standard six-octet notation. | #FFE8FF |
| setbackground | The background colour of an image set entry in various pages; note in particular that this colour hint can be the only visual differentiator between images and videos in the likes of wacsmphumbs and wacsimglist - think about this before making it the same as the other two background settings below. | #FFB0FF |
| vidbackground | The background colour of a video set entry in various pages. See note above. | #FFFFB0 |
| dvdbackground | The background colour of a DVD set entry in various pages. See note above. | #B0FFB0 |
| mastheadforeground | The foreground colour of the masthead on model pages and the front page | #F0D8F0 |
| mastheadbackground | The background colour of the masthead on model pages and the front page | #000040 |
| mastheadhighlight | A suitable alternative foreground colour to use in highlighting over the top of the mastheadbackground colour (new in WACS 0.8.2) | #F0B8B8 |
| divider | The colour of the section divider in various indexes. | #FFA0A0 |
| indexcolours | This controls whether models names in wacsmodelthumbs should be colour coded by rating, status, etc or not. It should be set to either Y or N depending upon whether you want the colour coding active. | Y |

Chapter 14. Layout

Overview: Layout

This controls the number of columns of set icons that Wacs will try to display and how big those icons will be. Other variables to control more aspects of layout, including model index layouts will no doubt follow. A few of these values are still hard coded in the scripts, but this should change over time.

wacs.cfg: Layout Section

Table 14.1. Layout Section: wacs.cfg

| Variable | Description | Typical Value |
|----------------|---|---------------|
| vendormode | Whether to enable the vendor mode pull down menus on the various javascript menus. Should be "Y" or "N". Please see the AppNote section for information on how to reproduce the old-style behaviour on WACS 0.8.x and later. | Y |
| style | What mode to use for layout - modelattr is for layout by hair and breast size, gallery is for topic-based galleries. | gallery |
| thumbsize | The maximum size of thumbnails for a set. Obviously since most images are not square, this will typically be the larger of whichever two dimensions the icon has. | 220 |
| thumbcount | Number of set thumb nails to be displayed going across the web page. | 4 |
| thumbrows | Number of set thumb nails rows to be displayed in paged-mode operation, ie wacsindex/page . | 3 |
| modelcount | Number of model thumb nails to be displayed going across the web page (based on using small size icons, ie 120x156). | 7 |
| minisize | The size of miniature icons for a set. These are typically dynamically resized by wacsimg on request. | 110 |
| setspergallery | The number of sets to be placed in each numbered gallery before moving on to the next available gallery number. ^a | 20 |
| units | The units of measurement to be displayed (database is all metric values); can be either <i>metric</i> or <i>imperial</i> . ^b | metric |
| mpdividers | Controls whether or not the dividing bars between media types should be included when drawing the model page. mpdividers can be set to either Y or N. ^c | Y |
| newperiod | Controls the period for which a model, image set or video is considered to be new. This affects what is shown by wacsnewmodels , wacsnewsets and wacsnewvideo . The period is given in number of days - certain common combinations are recognised and given text names: 6, 7 or 8 means week; 14 or 15 means fortnight; 28, 29, 30 or 31 means month; and 90, 91 or 92 means quarter. | 31 |

^aonly applicable to WACS 0.8.1 and higher

^bonly applicable to WACS 0.8.2 and higher

^conly applicable to WACS 0.8.4 and higher

^donly applicable to WACS 0.8.5 and higher

Chapter 15. Precedence

Overview: Precedence

This is a more complex section since most of the precedences are determined by how you've set up *areas* (aka *sarea*) on your image collection. The `default` section should always be present, but the others are named for the areas you've chosen. The algorithm is: look for a precedence setting of either `models` or `sets` for the given *sarea*, and failing that, return the default value. So if you have *sareas* of `favourites`, `amkingdom` and `usenet` and want indexes to show models thumbnails in `favourites` and `amkingdom` but set thumbnails in `usenet`, you would make this section read:

Example 15.1. Sample Precedence Section

```
<precedence>
  <default>models</default>
  <usenet>sets</usenet>
</precedence>
```

wacs.cfg: Precedence

Table 15.1. Precedence Section: `wacs.cfg`

| Variable | Description | Typical Value |
|-------------------------|--|---------------------|
| <code>default</code> | What the "normal" precedence should be, <code>models</code> or <code>sets</code> . | <code>models</code> |
| <code>sarea_name</code> | Override the precedence for the specified <code>sarea_name</code> to be the value specified (usually opposite of the default). | <code>sets</code> |

Chapter 16. Debug

Overview: Debug

This section controls the debugging message level for the various wacs modules and applications, making it possible to increase the amount of diagnostic output selectively to aid in debugging code or to provide information for putting in useful bug reports. At it's simplest, this section would read:

Example 16.1. Sample of Global Debug Setting

```
<debug>
  <global>4</global>
</debug>
```

This will set the global debug level accordingly (once the `read_conf` routine has been executed by the application of course). For more fine-grained control, the debug level in modules can be specified with `module_wacs` or for an application with `app_wacsimport`. That is `module_` and then it's name in lower case for modules, and `app_` followed by the application name (again in lower case) for applications; `tool_` for command line tools, and `util_` for programs normally run by other things (ie the database populators). Thus if you were working on code for `wacsimport`, you might choose to make this section read:

Example 16.2. Sample Of Specifically Targetted Debug

```
<debug>
  <global>0</global>
  <module_wacs>1</module_wacs>
  <app_wacsimport>5</app_wacsimport>
</debug>
```

These settings would provide some basic extra debugging for the `wacs` module (key pathnames, etc) and then pretty extreme output for the `wacsimport` application.

Chapter 17. WACS Configuration: Menus

WACS Menu System Overview

One of the cornerstones of the WACS philosophy is that there are a number of free standing applications, each of which has a number of optional behaviour modifiers, which combine together to build the final web environment. This means that the entire behaviour of the WACS system can be modified by how these various parts interact with each other, and how they interconnect. You can configure your WACS server to group together the links in the way you want them to appear, interacting both with the provided WACS applications and with any of your own you wish to add, written in either Perl or PHP.

Each WACS web application has a title bar with a number of menus shown on it; these menus are derived from entries in a single menu file or in their absence from some basic default menus. All WACS applications then add the appropriate menu to the title bar with either the default options, or the customised menu depending on what is present in the menu configuration file. By convention, the right-hand most menu on a WACS application is one called navigation, which provides links to the Main menu (front page), recent additions pages, the alphabetical list of models and the saved searches.

Each menu has a distinct name, which can be either one of the standard ones listed below or any other menu you care to define in the `menu.cfg` file. If a given menu is defined both as a default and in the `menu.cfg` file, the one defined in the `menu.cfg` file takes precedence.

Basic Default Menus

There are currently seven general default menus which can always be assumed to exist in a WACS configuration - these menus are available in both the Perl and PHP versions of the API and are used by most of the standard Wacs tools.

Table 17.1. The Default Menus

| Name | Title | Description |
|---------------------|---------------|---|
| navigation | Navigation | Standard navigation menu: main menu, new additions, alphabetic model index, and saved searches. |
| search | Search | Searching menu: saved searches, image search, video clip search, and model search. |
| favourites | Favourites | Models can be classed by a number of categories of Favourite: Cuties, Solo, Straight, Lesbian and Current Featured - this provides links to these categories. |
| newrelated | Related Lists | Links to relevant things related to new items: the new videos page, the new images page, and the same pages for the previous week. |
| alternatives | Alternatives | Alternative views of the same model details. |
| tagsalt | Alternatives | Alternative activities for a pre-existing tag set. |
| admin | Admin | Menu of collection administration tools |

Menu XML tokens

Basic Menu XML Types

Table 17.2. The Menu XML Types

| Name | Description |
|----------------|--|
| static | A single menu entry with a defined description and destination. |
| foreach | A menu entry which will repeat for each value in the specified passed record structure (hashref in perl). Please see the entry on the menu functions in the programming manual, reference section, for Wacs UI |
| divider | Puts in a horizontal rule at this point in the menu |
| heading | Puts in a bold heading in the menu. Note that this is allowed to have a link on it if you want that but it will not have substitutions applied. |

Sample Menu

Shown below is a very simple sample menu.

Example 17.1. Sample Menu

```
<?xml version='1.0' standalone='yes'?>
<menu>
  <main>
    <menutitle>Main Menu</menutitle>
    <default>#WACSALLM#</default>
    <entry>
      <type>static</type>
      <title>All Models Index</title>
      <target>#WACSALLM#</target>
    </entry>
    <entry>
      <type>static</type>
      <title>PHP Demo</title>
      <target>#BASEURL#simple/index.php</target>
    </entry>
  </main>
</menu>
```

Menu Handlers

In addition to the core menu system described above, the menu files can also optionally contain a virtual sub-menu of handler directives - these inform applications using this particular menu of the preferred applications for handling certain types of objects. Simply put, it allows the regular Wacs applications to better integrate with the rest of your environment by using your applications instead of the standard ones. This was originally developed as a way of re-using the wacsindex paged icon display within the Wacs-PHP simple skin.

Taking the example of `wacsindex`, or rather it's configurable alter-ego, `wacspage`, this meant that we wanted the link on model's names to jump out to the Wacs-Php Simple Skin version of the `modelpage` (`girlie.php`) rather than using the conventional `wacsmptthumbs`. We also wanted to suppress the links on the Photographer name as we have no photographer handler in the Wacs-Php simple skin yet.

The Special *handler* Menu

Diving straight in, here is the handler menu from the `simple.cfg` file distributed with the Wacs-Php Simple Skin.

Example 17.2. Sample handler Menu

```
<handlers>
  <models>#BASEURL#simple/girlie.php?modelno=#OPTIONS#<models>
  <photographer>#</photographer>
</handlers>
```

The first of the active lines specifies that links on model names should link to the model page from the Wacs-Php simple skin, `girlie.php`. The second one merely suppresses the link to photographers, as the environment concerned does not include a displayer for that data type. If you want to test for a null return, check for the reply being just the hash symbol.

Chapter 18. WACS Web Apps: Presentation

Presentation Applications

- wacsnewmodels
- wacsnewsets
- wacsnewvideo
- wacsshow
- wacsvidshow
- wacsideshow
- wacsfp

wacsnewmodels

wacsnewmodels lists models recently added to the database - it extends it's definition of newly added dynamically from two weeks up to two years in two week increments until it has found more than six new models.

wacsnewsets

wacsnewsets lists sets recently added to the database. Optionally takes an argument of a date from which to work back - this function is used by the previous week link at the bottom of the normal page but can actually take any arbitrary date. To do this for the week backwards from the 21st September 2006, this would be called with **wacsnewsets/21-SEP-2006**. [NB: this is currently assumed to be native date format of the database, ie 21-SEP-2006 for Oracle or 2006-09-21 for MySQL; one would hope that other SQL implementations would at least accept dates in one of these formats.]

It also includes a facility to import all of the new sets for the week concerned into a tag set for subsequent perusal. To do this, you need to select a tag set number which has not been used before, unless you wish to append the new items to this tag set. If you select tag set 599, to copy all new sets this week into this you call wacsnewsets with **wacsnewsets/T599**. It is also possible to create a tag group of those sets that were new at a previous date; for the 8th September 2006 into tage set 597, you would call wacsnewsets with **wacsnewsets/T597=08-SEP-2006**.

wacsnewvideo

wacsnewvideo is same code and features as wacsnewsets, just with the control variable changed so it features video clips instead of image sets. All the same features and options are supported.

wacsshow

wacsshow makes a pretty index of the categories for a given sarea including a few items from each area taken at random. For example to browse the section redheads, you would invoke wacsshow with **wacsshow/sarea=redheads**. It can also take additional arguments including **order**, **sample** and **limit**. The order argument can be either **order=F** for a conventional forward sort, ie A-Z or 0-9, or **order=R** for a reverse sort, ie Z-A or 9-0. The primary point of this is to use for galleries where it can be used to put the highest numbered, ie latest, galleries first on the list. The sample argument selects how it should select the highlighted images; the options are **sample=R** for a random sample, **sample=L** for the latest additions to be shown, or **sample=H** for the highest rated sets to be shown.



Note

sample=H is silently ignored for models at this point because it would require excessive additional database activity to calculate.

The limit argument is used to reduce the number of subareas featured takes the form of **limit=5** which would limit wacsshow to five areas. You can combine these to form something like: **wacsshow/sarea=redheads:order=R:sample=H:limit=5** which would show the latest five galleries in the areas redheads, select the sample images from the very highest rated sets and show just five galleries.

Wacsshow's behaviour is significantly controlled by the precedence section of the WACS configuration file. In this a default precedence can be set to either models or sets; it can also be overridden per specific sarea. So you might set your default precedence to sets, but decide that for the sarea redheads you would override this to models. Thus when looking at some other area, such as maybe "usenet" for general stuff downloaded off the net where each set is not necessarily linked with any model, it shows you a selection from the sets within that section. However when you look in "redheads", it will instead show you the model icons of those models with sets featured under this section. For more information on configuring this behaviour, please see the bit about configuring precedence in the config section.

wacsvidshow

wacsvidshow - the video version of wacsshow - options as above.

wacsslideshow

wacsslideshow runs a web-based slide show of a set. Sometimes more convenient than downloading a zip is to watch a slideshow online. This functionality is provided by wacsslideshow, to invoke basically just call with the set number, for instance for set123 call it with **wacsslideshow/set123**. It can also support the display of a tag set or connection set in a similar fashion, merely invoke with tag and the relevant (set-based) tag number, for instance to see all sets in tag504 call it with **wacsslideshow/tag504**. For a connection, invoke it with conn and the connection number, as in **wacsslideshow/conn10** to show connection 10 as a slide show. To do a slideshow of all a model's sets, you can use either mod or model followed by the model number, eg call it with **wacsslideshow/model1234** to show all of model no 1234's sets.

wacsfp

wacsfp is the standard front page of the WACS system. It doesn't take any options at this time, but offers dynamically generated, context sensitive pulldown menus on top of a selection of random suggestions and useful links allowing a wide range of access around the entire Wacs system.

Chapter 19. WACS Web Apps: Indices

Index Applications

- wacsimgcats
- wacsimglist
- wacsvidcats
- wacsvidlist
- wacsphotcats
- wacslistphotog
- wacsstats

wacsimgcats

wacsimgcats generates a concise index of the images. Without an argument does everything. Can also take an sarea as an argument, so to look at only the area called "redheads", call with **wacsimgcats/sarea=redheads**

wacsimglist

wacsimglist produces thumbnails of the images in the specified location - normally takes a complete sarea and scategory listing as arguments. Thus for a redhead with small breasts called Sarah, you would invoke it with **wacsimglist/sarea=redheads:scategory=smalltits/Sarah**. Optionally, it will also support a photographer based narrowing of the list, so if you wanted Sarah's sets only by photographer Fred, you would invoke it with **wacsimglist/sarea=redheads:scategory=smalltits/Sarah:sphotog=Fred**. **wacsimglist** can also cope with a wildcard at the end of the category field, so you could invoke it with **wacsimglist/sarea=redheads:scategory=smalltits/+** to show you all redheads with small tits. You can further narrow it down using the sphotog selection criteria as well, so that to see all Fred's sets of small titted redheads, you can do: **wacsimglist/sarea=redheads:scategory=smalltits/+:sphotog=Fred**. It will also accept slocation as either a replacement for sarea or as a modifier on any other valid selection. This is invoked by wacsimgcats and wacsshow, and calls wacsindex to display any set selected. Note that if the /+ is appended to the end, the output will include it's last component as a title; if it's left off, no title will be added.

wacsimglist also supports use of the conn (connections) table - if you invoke it with **wacsimglist/cgroup=123** it will display the members of cgroup 123. This can also be modified with the sphotog= option as above. **wacsimglist** similarly supports displaying tag sets via the same mechanism, ie **wacsimglist/tgroup=123**.

wacsvidcats

wacsvidcats is the video version of wacsimgcats. Uses exactly the same script with a control variable changed in it.

wacsvidlist

wacsvidlist is video version of wacsimglist. Uses exactly the same script with a control variable changed in it. Supports photographer searches in exactly the same way.

wacsphotcats

wacsphotcats is the photographer catalogue, it's again based upon wacsimgcats with a control variable changed. To search for a given photographer's sets, for example, SWE (Sweet Photography) call with **wacsphotcats/SWE**.

wacslistphotog

wacslistphotog produces a list of photographers, ranked by rating, with a little bit of information about them. Each one links to wacsphotcats to display that photographer's work. It takes no options at this point.

wacsstats

wacsstats produces a number of different statistics about the current wacs collection - in particular, it does a whole range of top 25s - models with the most sets, videos, images, biggest sets, etc. Not particularly essential, but great fun and a different angle from which to explore the Wacs collecton.

Chapter 20. WACS Web Apps: Models

Models Applications



Warning

This is NOT an introduction to these web applications in a general sense - that is provided in considerable detail in the Wacs User Guide. This section is about how you modify the behaviour of these commands, typically when building your own menus using them by specifying options in the URL.

- wacsmodelthumbs
- wacsmodelindex
- wacsmodelpage
- wacsmphumbs
- wacsmpmi

wacsmodelthumbs

wacsmodelthumbs displays thumbnails of models in the database. By default, this will display all the models in the database - this can be a lot of entries so there are many modifiers available to limit the results to something useful.

First up there are the flagged models - currently this consists of six categories: Cuties (C) defined as good looking models whose sets are not particularly explicit, Solo (S) models mostly selected for good solo sets, Lesbian (L) those who have particularly good lesbian sets, Straight (F) models who have particularly good straight sets, Featured Models (M) for those who are currently being highlighted for one reason or another and Placeholders (P) used to hide pseudo-model entries that may be convenient for download purposes (for instance a group combination) but aren't real identifiable single models. For each of these, just provide the appropriate capital letter when calling **wacsmodelthumbs**, for instance use **wacsmodelthumbs/C** for Cuties, and **wacsmodelthumbs/L** for Lesbian sets.

Next up is the vendor search mechanism. Here it is assumed that you've utilised the idmap facility and have allocated an appropriate abbreviation to each source of material. So if for instance you've marked models from Sapphic Erotica (www.sapphicerotica.com) with an IDMap flag of SE, the following examples would apply. To see all models with content from Sapphic Erotica, you would call **wacsmodelthumbs** with **wacsmodelthumbs/V=SE** where V stands for vendor. You can further select just those models to whom you've given a top rating, ie a mrating value of 4 or 5, using the Q (for Quality) selector. Thus **wacsmodelthumbs/Q=SE** will select only your top rated models from vendor SE. Another vendor based selector is connected to the download mechanism - N - selects models who have had new sets added within the last three months (as determined by the refresh and chkmodel commands on their last run). Thus **wacsmodelthumbs/N=AMK** will select only those models updated within the last three months. A companion to this is the D (for Dormant) selector: this selects those models who have not been active within the last three months. Thus **wacsmodelthumbs/D=KPC** will select all Karup's PC (KPC) models who have not had new sets added in the last three months.

The final two vendor search options are designed to help with maintenance of the system and shows those models from a given vendor who have not yet been rated or who have incomplete identification to be selected. The U (for Unclassified) selector finds models with sets from this vendor who have no specified rating, it is used like this: **wacsmodelthumbs/U=SE**. The X (for Incomplete) selector finds models flagged as coming from this site but where we have no key or last checked information for them. You can look for models with incomplete ids with **wacsmodelthumbs/X=KHA** which will find models from Karup's Hometown Amateurs (KHA) with incomplete information.

Next is the Alpha-search mechanism, used typically for doing pages for a given letter of the alphabet, but quite capable of being even more selective if required. For an index of models with names beginning with H, you would call wacsmodelthumbs with **wacsmodelthumbs/A=H**. When using the Alpha search mode (A=), there will be a row of links for each letter of the alphabet along the top of the page, just below the normal hot item links. If you want to search for all models beginning with Bet for example, you can do **wacsmodelthumbs/A=Be** and it'll find Beth, Betsy, Betty, etc. It will also find models with an alias of Betty or an identity record for a site with a name of Betty¹. If for example you have the prolific Hungarian model Zafira in your system, Zafira will appear in a search for Betty as she's called Betty 2 on Karup's PC. This can be a little confusing at first, but is actually very useful when you've figured out what is actually going on.



Note

The search for aliases and idmap names is only enabled if you give two or more characters in the search, the **wacsmodelthumbs/A=B** will not include Zafira as that is trying to produce a straight forward alphabetic index. Search for **wacsmodelthumbs/A=Be** and alias and identity map names will be included, and in this example Zafira will be included in the results.

Finally you can get wacsmodelthumbs to display an index of models in a model based tag set by using the T option. For example, to display the members of tag set 595 (which must be a tag set of models, not sets), you would call wacsmodelthumbs with **wacsmodelthumbs/T595**. In summary, the options are:

Table 20.1. URL Options to WacsModelThumbs

| Invocation | Meaning |
|-----------------|--|
| A=arg | Alpha search for argument <i>arg</i> . If you give two characters or more, aliases and identity map names will be included in the results ^a . |
| C | Models flagged as Favourite Cuties |
| D=vendor | Models flagged as Dormant from the specified <i>vendor</i> (WACS 0.8.4 and later) |
| F | Models flagged as Favourite Straight |
| L | Models flagged as Lesbian Favourites |
| M | Models flagged as Current Featured Models |
| N=vendor | Models flagged as having had new sets recently from <i>vendor</i> |
| P | Models flagged as Placeholders, normally ignored |
| Q=vendor | Models with a top rating score from <i>vendor</i> |
| S | Models flagged as Favourite Solo Models |
| Tnumber | Models listed in tag number <i>number</i> |
| U=vendor | Models with no rating score with sets from <i>vendor</i> |
| V=vendor | Models with sets from <i>vendor</i> |

¹This feature was introduced in WACS 0.8.4, prior to that similar alternative name searching was only available through **wacsmodelmgr**

| Invocation | Meaning |
|-----------------|---|
| X=vendor | Models with incomplete identity information from <i>vendor</i> (WACS 0.8.4 and later) |

³Feature available in Wacs 0.8.4 and above

wacsmodelthumbs is capable of colour-coding the names of the various models to show key aspects of their status and ratings - this is enabled or disabled by the **indexcolours** directive in the configuration file. When enabled, (indexcolours = Y), it will colour model's names as follows: green - rating of 4 or 5 (ie a high rating), amber - rating of 3 (a medium rating), magenta rating of 1 or 2 (a low rating). Models with no sets yet unpacked will be coloured blue. Black signifies all others - typically some sets but no rating.

wacsmodelindex

wacsmodelindex displays a detailed index of models. This displays a far more comprehensive index of each matching model's details and includes most of the information in the database reformatted into sensible sentences (mostly!). It gives details on age, country of origin, shows any notes, and lists the number of outstanding sets/videos yet to be downloaded and already downloaded but not catalogued. It's a highly extended version of wacsmodelthumbs and accepts all of the same optional selection parameters listed above.

wacsmodelpage

wacsmodelpage displays a model's portfolio. This is the main page for looking at model's work; it has a heading section at the top with the big icon and a selection of attributes about her and navigation bar. Below it has a box of information including the icon, attributes and size for each set and video of hers that we have. Normally wacsmodelpage is invoked with our model number for a model, for instance **wacsmodelpage/147** would retrieve the page for our model 147. However, if the model number concerned is not found, the key given is also tried for any idmap keys that might match. This is not very useful for sites which started their own model numbers from a low number, but works very well for keys from sites such as those from the AMKingdom group who have a really good alpha-numeric reference scheme. In this scheme, a model called Karlie is kar042, a model called Sabrina is sab001, and so on the numbers incrementing sequentially for each new model with that three letter prefix. It also works for sites such as how Karup's PC used to be ² where the model key is her name followed by a number, thus for a model with key bailey_(2) you can use that (complete with brackets of course).

There are currently seven groups of modifiers on the behaviour of **wacsmodelpage**. By default, if a `prefexcl` attribute exists within the access control list file (see Chapter 11, *Security*) for the current host/user combination, this will be applied to the search of the models portfolio. When this mode is enabled a warning and link is added in the page heading. To disable the preference exclusions (`prefexcl`) mode for model 123, you call wacsmodelpage with a /N appended to the URL, as in **wacsmodelpage/123/N**. This functionality is all suppressed if the `prefexcl` flag is not set for the current user. The second modifier allows the copying of all of this model's sets to a tag index, to add model 123's sets to tag set 456, you call wacsmodelpage with **wacsmodelpage/123/T456**. All of these options work for idmap based key searches too, so if you want Erika's (from AMK) sets added to the tag set, you can do **wacsmodelpage/eri046/T456**. You can also specify that a tag set should be created with the next available tag set number by using /Tnext as in **wacsmodelpage/eri046/Tnext**. In addition to the T option which tags both Image and Video sets, there is also an I option which tags only the image sets, as in **wacsmodelpage/123/I456** or **wacsmodelpage/123/Inext.#** The V option has the corresponding effect but for video files instead of image sets. The third option allows you to display only those sets by a specific photographer, this is done with the /

²With a recent redesign, Karup's PC shifted to using a numeric model number like many other sites so this no longer works except on models who still have the old style identity information as well

P= option. This might be used as in **wacsmodelpage/123/P=BMB** to select only those sets by model no 123 that were shot by photographer BMB (this cancels out the preference exclusion feature, intentionally). The fourth option allows you to display only those sets featuring a lesbian liason between the main model and a second specified model, so if for instance you wanted to see those sets where model 9 goes with model 15, you would do **wacsmodelpage/9/L=15**. The fifth option R (for Room) is to select sets and videos only in a specified location, done with the /R= option. If you want to select sets in the bathroom only, you would do **wacsmodelpage/204/R=Bathroom** . Note that the /R= option DOES NOT override preference exclusions. The sixth option A (for Attire) is to select sets and videos where the model is wearing the specified type of Attire. For instance if you want only sets featuring Swimwear, you would do **wacsmodelpage/147/A=Swimwear**; for school-themed clothing like cheerleader or schoolgirl you would use **wacsmodelpage/147/A=Schoolwear**. The seventh option F (for Flag) is to select sets and videos only with the specified set type, done with the /F= option. If you want to select Lesbian sets only, you would do **wacsmodelpage/1080/F=L** ; for toys, you would use **wacsmodelpage/1080/F=T**. This option does override the preference exclusions; if you ask for a type you've excluded it would be pointless not to show it to you.



Note

If the preference exclusions (prefexcl) feature is active, it will be applied to the tag set creation, so only the preferred types of sets will be tagged. At the moment the modifiers are mutually exclusive.

wacsmphumbs

wacsmphumbs displays a model's portfolio in a smaller, more condensed table layout. It is actually wacsmodelpage with a different global option set; all the same features are included. Preference exclusions, the /N override and the /T*tagno*, /P=*photog*, and /L=*modelno* and other modifiers are all fully supported.

wacsmpmiini

wacsmpmiini displays really small thumbnails for those models with huge numbers of sets. As with wacsmphumbs this is again wacsmodelpage with a different global option set, so the same options and features are supported.

Chapter 21. WACS Web Apps: Tag

Tag Applications

- wacstagsets
- wacslisttags

wacstagsets

wacstagsets displays a set-based tag set. To invoke for tagset 456, call with **wacstagsets/456**. Note that one of the options to wacsmodelindex above provides equivalent functionality for model-based tag sets.

Since wacstagsets provides a link to wacsdeltag for each individual set it displays, it provides a very rudimentary way of editing set-based tag sets, in so much as you can create a tag set through wacsnewsets or the search functions, and then trim the entries you do not want using wacstagsets.

wacslisttags

wacslisttags lists current known tag sets. Wacslisttags provides the main way into using the tag system - it provides an index of the currently defined tag sets and calls wacstagsets or wacsmodelindex appropriately. It does not take any options at this point. Only an admin level user can see all people's tags, other users are restricted to those they have created, are owned by 'wacs' or have been set as public. WacsListTags now includes an option for set based tags to call the wacsslideshow program to see a slideshow of the whole tag set. Wacslisttags now includes a link on the tag number to a short form index (via wacsimglist) and to the more detailed wacstagsets on the title of the tag set. It also shows administrators the owner of the tag set concerned.

Chapter 22. WACS Web Apps: Search

Search Applications

- wacsimgselect
- wacsvidselect
- wacsmodelselect

wacsimgselect

wacsimgselect searches for photosets by attribute. This application puts up a form with lots of options based upon source, attributes, photographer and rating; when this form is submitted a new tag set is created with the results of the search. It doesn't take any special options, all interaction is via the web form.

wacsvidselect

wacsvidselect searches for videos by attribute. This is a version of wacsimgselect modified to work with videos.

wacsmodelselect

wacsmodelselect searches for models by attribute. This application puts up a form that allows you to search for models by their attributes. When the form is submitted it creates a model tag set of all the matching records.

Chapter 23. WACS Web Infrastructure: Retrieval

Retrieval Applications

- wacsfull
- wacsframe
- wacsimg
- wacsindex
- wacspage
- wacsthumb
- wacszip
- wacsusc2257

wacsfull

wacsfull sends the full size image from a set - for image 45 of set 123, call with **wacsfull/set123-45.jpg**. If called with **scale** instead of **set**, it will rescale and shrink the image as specified in the current user preference settings. To use the scaling option, call with **wacsfull/scale123-45.jpg**.

wacsframe

wacsframe puts a simple HTML page wrapper around an image provided by **wacsfull** to provide a better browsing experience. It is typically called by **wacsindex** with four parameters separated by = (Equals signs). The first parameter is the stem of the **wacsfull** command (**set1234** or **scale1234**) depending on the desired base image; the second parameter is the image number within the set to display, the third parameter is the number of images in the whole set, the final parameter is a list of models involved. An example invocation asking for image 12 of a 76 image set called 456 featuring Sarah, our model no 123, scaled to the current scaling factor would be: **wacsframe/scale456=12=76=Sarah-123**. Multiple models can be called using commas , as in [...]=**Sarah-123,Jane+P-67** thus adding links to Sarah, model no 123, and Jane P, model no 67. The complexity of this URL completely avoids the need for this script to talk to the WACS database or configuration system at all.

wacsimg

wacsimg sends the icon for the specified set - for set 123, to get the icon call with **wacsimg/set123.jpg**. To force it to produce only official icons, call with **wacsimg/icon123.jpg**. For scaling to the standard icon size, as specified in the configuration file layout section option "thumbsize", use **wacsimg/std123.jpg**. To get a miniature icon as specified by the "minimize" option, use **wacsimg/mini123.jpg**

wacsindex

wacsindex provides details of, and thumbnails of the specified set - for set 123, call with **wacsindex/set123.html**. Note that this has now been enhanced to include the official icon if available, and attempts to make some kind of page for videos as well. It will also include any set description (sdesc) that has been added for that set. It will also show any additional icons that have been associated via the additional icons (saddicon) field. If called with **wacsindex/page123.html**, it will produce a pretty format multi-page index instead of the standard full set page. Subsequent pages are called by **wacsindex/page123-2.html** and so on. If called with **wacsindex/info123.html**, it will produce detailed information for the set along with a group of eight icons from even spaced locations through the set giving some idea of the action.

wacspage

wacspage is actually a clone of **wacsindex** but is configured to operate only in the *prettyist* mode, the page mode. The trick is that the prefix on the set number is menu configuration name to make use of. So if you want to use the **wacsindex** within a custom web site, you can do this with wacspage. There is a worked example of this as part of the Wacs-PHP simple skin making use of the `simple.cfg` where you call **wacspage** with `wacspage/simple234.html`.

wacsthumb

wacsthumb provides the specified thumbnail image from the set - for the thumbnail image for image 45 of set 123, call with **wacsthumb/set123-45.jpg**. Additionally from Wacs 0.8.5, wacsthumb can be used to retrieve, scale and cache additional icons from the directory specified in the set. The additional icons can be retrieved at normal size (which may include upscaling them) using **wacsthumb/add123-4.jpg** or at mini icon size using **wacsthumb/addmini123-4.jpg**. Also added in Wacs 0.8.5 is the ability to provide .png files as an alternative to .jpg, merely by requesting **wacsthumb/set123-45.png**. The final new ability added in Wacs 0.8.5 is the ability to provide mini versions of the thumbnails through **wacsthumb/mini123-45.jpg**.

wacszip

wacszip is used to retrieve an archive file of actual content. For an image set, it returns a zip file; for a video set the video file. It can be called with any of the supported extensions (.zip, .wmv, .mpg or .mov) and will return the content regardless. Generally, getting the extension right is vital to the browser doing the right thing, but wacszip doesn't care! Examples of use would be **wacszip/set123.zip** and **wacszip/set456.mpg**.

wacsusc2257

wacsusc2257 displays the 18 USC 2257 declaration for the set named, or if that fails a generic summary of upstream source data. Example of use would be **wacsusc2257/123** this should be used on all pages showing actual image content.

Chapter 24. WACS Web Infrastructure: Tags

Tag Infrastructure Applications

- wacsaddtag
- wacsdeltag
- wacsmarktag

wacsaddtag

wacsaddtag adds a model or set to a tag. In order to simplify the process of tag handling and make it modular and distinct from each of the retrieval tools that might wish to use it, tags are actually manipulated using completely separate cgi-bin programs. These are called with the necessary parameters to do the tag manipulation requested and return a PNG image on completion - a tick for success, a cross for failure. Thus addtag is called repeatedly to add each entry into a tag group. At the simplest level, there are two default temporary tag sets, tag set no 1 for sets, and tag set no 2 for models. To add set 123 to the default tag set for sets, you would call wacsaddtag with **wacsaddtag/S123**. To add model 12 to the default tag set for models, call it with **wacsaddtag/M12**.

To provide a better interface for the search engine to use, wacsaddtag also has two more complex options, A and G. These each take three arguments separated by equals signs; A is for sets and takes desired tag number, set to add to it, and description of this tag; G is the equivalent for models and takes tag number, model number and tag description. Thus to add set 123 to tag set 456 and call it Cuties Outdoors, you would call wacsaddtag with **wacsaddtag/A456=123=Cuties+Outdoors**. For models, say you want to add Sarah (model 63) to Cute Redheads (tag 789), you call wacsaddtag with **wacsaddtag/G789=63=Cute+Redheads**.

wacsdeltag

wacsdeltag removes the specified item from the specified tag set. To remove set 123 from tag set 456, you would call wacsdeltag with **wacsdeltag/S456=123** - note that the tag number always follows the key letter. To delete all entries in tag set 456, you would call it with **wacsdeltag/S456=all**. Wacsdeltag also does the same with models, so to delete Sarah (model 63) from tag set 789, you would use **wacsdeltag/M789=63**. Once again, if you put all, you can delete that entire model-based tag set, as in **wacsdeltag/M789=all**.

wacsmarktag

wacsmarktag marks a specific tag set member as already viewed. To mark a set as seen call wacsmarktag with **wacsmarktag/S456=123**. It also supports model-based tag sets, as in **wacsmarktag/M789=63**. To clear the marked-as-viewed state on a complete tag set, you would do **wacsmarktag/S456=reset** and all records will be set back to unmarked. This only allows updating of temporary (search result) tag sets. It is also possible to use wacsmarktag to archive older search results to drop them from the visible list - they

can still be viewed if called explicitly. For instance, to archive tag 34 (containing sets), you would use **wacsmarktag/Z34=all**. For model-based tag sets, it would be **wacsmarktag/Y20=all**.

Index

A

- actithresh
 - Keyword (download section), 32
- addicons
 - Keyword (fsloc section), 28
- admingroup
 - Keyword (security section), 31
- adminusers
 - Keyword (security section), 31
- Architecture, 3
- archive
 - Keyword (fsloc section), 29
- assoc
 - Keyword (tables section), 25
- authmode
 - Keyword (security section), 31

B

- background
 - Keyword (colours section), 33
- barredusers
 - Keyword (security section), 31
- baseurl
 - Keyword (server section), 26

C

- cgibindir
 - Keyword (fsloc section), 29
- cgiurl
 - Keyword (server section), 26
- checkcount
 - Keyword (download section), 32
- Colours
 - Choosing Good Ones, 4
- colours
 - Changing..., 11
- Configuration
 - Database, 23
 - Download, 32
 - File System Locations, 28
 - Operating System, 6
 - Permanent, 4
 - Security, 31
 - Server, 26
 - Transient, 5
- conn
 - Keyword (tables section), 25

D

- Database

- MySQL, 24
- Oracle, 24
- Overview, 23
- dateformat
 - Keyword (database section), 23
- dbiconnect
 - Keyword (database section), 23
- dbienvvalue
 - Keyword (database section), 23
- dbienvvar
 - Keyword (database section), 23
- dbname
 - Keyword (database section), 23
- dbpass
 - Keyword (database section), 23
- dbuser
 - Keyword (database section), 23
- debugging, 37
- default
 - Keyword (precedence section), 36
- directurl
 - Keyword (server section), 26
- divider, 39
 - Keyword (colours section), 33
- dormthresh
 - Keyword (download section), 32
- download
 - Keyword (fsloc section), 29
 - Keyword (tables section), 25
- dvdbackground
 - Keyword (colours section), 33
- dvdicons
 - Keyword (fsloc section), 28

F

- foreach, 39
- foreground
 - Keyword (colours section), 33

G

- gimp, 11
- glyphs
 - Keyword (fsloc section), 28

H

- heading, 39
- Hostname, 6
- htmldir
 - Keyword (fsloc section), 29

I

- iconurl
 - Keyword (server section), 26

idlethresh
 Keyword (download section), 32
idmap
 Keyword (tables section), 25
images
 Keyword (fsloc section), 28
imgtooldir
 Keyword (fsloc section), 29
indexcolours
 Keyword (colours section), 33
ISP Requests, 7

K

keyword
 Keyword (tables section), 25

L

leasedir
 Keyword (fsloc section), 29
leasefile
 Keyword (security section), 31
leases.acl, 5
leasetime
 Keyword (security section), 31
Logical Structure, 4

M

mastheadbackground
 Keyword (colours section), 33
mastheadforeground
 Keyword (colours section), 33
mastheadhighlight
 Keyword (colours section), 33
Menu Types, 39
menu.cfg, 4
Menus
 Overview, 38
 Sample, 39
minicache
 Keyword (fsloc section), 29
minisize
 Keyword (layout section), 34
modbigicons
 Keyword (fsloc section), 28
modelcount
 Keyword (layout section), 34
models
 Keyword (tables section), 25
modicons
 Keyword (fsloc section), 28
mpdividers
 Keyword (layout section), 34
MySQL, 24

N

newperiod
 Keyword (layout section), 34

O

officons
 Keyword (fsloc section), 28
Oracle, 24
OS Configuration, 6

P

photographer
 Keyword (tables section), 25
phpdbconnect
 Keyword (database section), 23
Port Forwarding, 8
powerusers
 Keyword (security section), 31

S

scalecache
 Keyword (fsloc section), 29
Server
 Overview, 26
setbackground
 Keyword (colours section), 33
seticons
 Keyword (fsloc section), 28
sets
 Keyword (tables section), 25
setspergallery
 Keyword (layout section), 34
siteurl
 Keyword (server section), 26
static, 39
Static IP Address, 6
style
 Keyword (layout section), 34

T

tag
 Keyword (tables section), 25
thumbcache
 Keyword (fsloc section), 29
thumbcount
 Keyword (layout section), 34
thumbrows
 Keyword (layout section), 34
thumbsize
 Keyword (layout section), 34
tooldir
 Keyword (fsloc section), 29

U

units

Keyword (layout section), 34

V

vendor

Keyword (tables section), 25

vendormode

Keyword (layout section), 34

vidbackground

Keyword (colours section), 33

videos

Keyword (fsloc section), 28

vidicons

Keyword (fsloc section), 28

W

wacs.acl, 4

wacs.cfg, 4

database, 23

download, 32

fsloc, 28

security, 31

server, 26

tables, 25

wacsaddtag, 54

wacsallm

Keyword (server section), 26

wacsdeltag, 54

wacsdir

Keyword (fsloc section), 29

wacsfp, 42

wacsframe, 52

wacsfull, 52

wacsimg, 52

wacsimgcats, 44

wacsimgselect, 51

wacsindex, 53

wacslisttags, 50

wacsmain

Keyword (server section), 26

wacsmarktag, 54

wacsmodelindex, 48

wacsmodelselect, 51

wacsmodelthumbs, 46

wacsmpmini, 49

wacsmpthumbs, 49

wacsnewmodels, 41

wacsnewsets, 41

wacsnewvideo, 41

wacsphotcats, 45

wacsshow, 41

wacsideshow, 42

wacstagsets, 50

wacsthumb, 53

wacsurl

Keyword (server section), 26

wacsusc2257, 53

wacsvidcats, 44

wacsvidlist, 45

wacsvidselect, 51

wacsvidshow, 42

wacszip, 53

X

XML

Quick Intro, 3