Wacs Manpage Collection

First Edition

for WACS 0.8.5

B "Beaky" King
Published 12th March 2010

Wacs Manpage Collection

by B "Beaky" King

for WACS 0.8.5

Abstract

WACS is a Web-based Adult Content Server environment built using the LAMP (Linux, Apache, MySQL and Perl/Php) infrastructure. It provides the database, toolchest and API infrastructure to build a sophisticated indexed archive of adult material for either personal or commercial use. It is a Free Software/Open Source package released under the GNU Public License Version 3 (GPLv3) and supports use of Oracle in addition to MySQL.

This document contains a collection of the command line tools provided with WACS and the various functions they perform.

The WACS source code and other documentation and support tools can all be found at the WACS website at Sourceforge [http://wacsip.sourceforge.net/]. Releases are also uploaded to the Launchpad [http://launchpad.net/wacs/] site for additional resilience. There is a demonstration web site for WACS at PinkMetallic.com [http://www.pinkmetallic.com/] where you can see WACS in action and download the sample sets featured in this document - *CAUTION: contains adult material*. The main developers can be contacted at wacs@beaky.name [mailto:wacs@beaky.name] - messages sent to this address will be treated as confidential. Any email addresses, names, URLs or other identification information will be removed before any publication of the contents on bug lists, feature requests, mailing lists etc. Commercial add-ons and support options can be purchased from Bevtec Communications Ltd, see their website at Bevtec Communications [http://www.bevteccom.co.uk/].

Table of Contents

Part I. Command Line Tools for WACS Users

The commands listed here cover those tools that are designed to be of use to users of the WACS system rather than those targetted to system administration tasks.

rungq(1)

Name

rungq — run an image viewer with a local slide show of Wacs image sets

runmp — run a movie player with a local list of Wacs video clips

Synopsis

```
rungq [modifiers] argument
```

```
runmp [modifiers] argument
```

Arguments

`tagnumber`	specifies that the *tag set* (aka Saved Search) of the number given should be used. This can be abreviated to **t** and the number or even just the tag number on it's own.
`connnumber`	specifies that the *connections set* of the number given should be used. This can be abreviated to **c** and the connection set number.
`modelnumber`	specifies that the complete portfolio of the model specified should be used. This can be abreviated to **mod** or even **m** and the model's number.

Modifiers

`--cat=`	specifies that only a single category of sets should be shown from those that would otherwise match - this uses the standard values for `scatflag`. The values are as follows: F for Straight, G for Group Orgy, M for Masturbation, L for Lesbian, S for Solo and T for Toys. Thus `--cat=T` will show only toy sets.
`--excl=`	specifies what categories of sets should be excluded - the default is Backstage and Duplicate. These are provided as a comma separated list. To show only solo sets you would specify `--excl=F,G,L,B,D`. To cancel the default exclusions, specify a blank exclusion list: `--excl=`.
`--skip=`	specifies how many of the sets that would otherwise have matched should be skipped. This allows you to restart partway through a particular set of matching sets/clips, eg `--skip=15` starts at the 16th matching set.

Description

While the Wacs system is primarily web-based, through tools like **rungq** and **runmp**, it is possible to access both the underlying database and image files/video clips in the WACS archive directly within the Linix/Unix shell environment. This offers some advantages in terms of speed and flexibility of display over accessing it through a web-browser but of course looses both the ubiquity and just-works aspects of using the web interface. Using **rungq** or **runmp** is very much an optional alternative with restricted functionality to using the full web interface.

In order to use **rungq** or **runmp** you will need to have the appropriate access at the operating system level to the database and to the files themselves; this is normally achieved through being a member of the Linux/ Unix group `wacs`. It is also possible to make this work on a workstation or laptop computer on the local network through a combination of NFS and network transport of the SQL requests to the database. More information on how to configure these is given in the Configuration Manual for the WACS system.

In essence what **rungq** does is retrieve the full pathnames of the directories containing the image sets specified and invokes an image viewer on those. It opens the first image at which point it is normal to fullscreen the viewer and either page through the images using the space bar or left mouse click, or set a slideshow going with a fixed interval between images. **rungq** can *play* in this way saved searches (aka tag sets), connections or a specific models complete portfolio. Various modifiers described above can make further sub-selections based on set type or skip over a number of the entries if these have been recently viewed. The **runmp** command works in a very similiar way but passes the full path names of the video clip files instead.

rungq conventionally uses **gqview** because this viewer will let the user browse through a slideshow of image directories specified in this way. This behaviour was added to **gqview** at our request by the original author - many if not most other image viewers require explicit specification of the image files themselves or open all the files at once (eog used to do this although by GNOME 2.26 it seems to work correctly).

runmp uses the **mplayer** but could easily use a different movie player if that is prefered.

Examples

- `rungq t23` - show the image sets in saved search number 23

- `rungq --cat=T t23` - show the image sets of category *T*oys from saved search 23

- `rungq --excl=B,D,F,G c7` - show the image sets from connection set number 7 skipping Backstage, Duplicate and Straight Sex sets.

- `rungq --skip=15 mod17` - show the portfolio of model number 17, starting from the 16th set.

- `runmp --cat=L m23` - show the Lesbian movies featuring model number 23.

Part II. Data Migration Tools

The tools covered in this section are related to those tools that allow the migration of data between various WACS installations.

wacsexport(1)
wacsimport(1)
wacsxmlout(1)
wacsxmlin(1)

Name

wacsexport — export a WACS model record as an XML file

Synopsis

wacsexport [--noimages] *model_number*

Argument

model_number   the number of the model you wish to export details of.

Modifiers

--noimages   specifies that images should not be included. This is used to exclude copyrighted images when exporting model information for public distribution (ie posting on a forum).

Description

The **wacsexport** command exports all of the information the current WACS installation knows about the specified model as an XML file called by her name and model number. Thus exporting Kaz B's model record from PinkMetallic.com would create an XML file called `KazB-3.xml`. This includes not only the basic model record but also all of her identity map (idmap) records and all related download records. Additionally it also includes any headshot or other icon pictures it has of her (unless told not to via `--noimages`) as Base-64 encoded elements within the XML file.

The resulting XML file is suitable for emailing, etc as it is "ASCII armoured" for the protection of binary data (ie images). It can be imported into another WACS installation using either the **wacsimport** command or the upload function of the **wacsmodelmgr** (WACS Model Manager> web application. The transfer of sets themselves and their associated descriptive data (metadata) is handled seperately by the **wacsxmlout** and **wacsxmlin** commands.

Examples

`wacsexport  3` would create a file called KazB-3.xml if Kaz B is model 3 on the current WACS installation.

See Also

wacsimport(1), wacsxmlin(1), wacsxmlout(1)

Name

wacsimport — imports details of a model from a WACS XML export file

Synopsis

```
wacsimport Model_XML_File
```

Description

Data about models can be exchanged between WACS installations by way of a number of XML file formats, one of which is that which describes models. The **wacsimport** command is provided to load a recieved WACS Model XML file into the WACS system; this includes the model's basic details, her known identities, her known sets and both the large and small headshot icons.

The **wacsimport** command tries to determine if the model described by the XML file is already known to the current WACS installation. If she isn't, it will create a new record for her, install any headshot icons enclosed in the XML file, add all her known identities and all known set information. If she is, it will update the existing record with any *new* information included within the XML file. This will include any new identities, known set information (aka download records) and any fields where there is no data already.

For instance, if her *hometown* is blank in our local data about her, but present in the XML file, it will be updated - existing values will not be overwritten. There is one exception to this in that if an identity is marked as Active or Dormant in the database and marked as Gone in the XML file, it will be marked as Gone in the database. This is because it's a data item unlikely to be factually inaccurate and is very likely to change over time.

Some sample XML model files are provided in the WACS distribution and additional ones will be made available through our demonstration site www.pinkmetallic.com as they become available.

See Also

wacsimport(1), wacsxmlin(1), wacsxmlout(1)

Name

wacsxmlout — exports set details as a WACS XML export file

Synopsis

```
wacsxmlout [--noimages] set_number
```

Description

The **wacsxmlout** command is a tool to create an XML export file containing information on a set to allow it to be transported between different WACS installations. **wacsxmlout** is invoked with a set number (ie set 123) and will create a file named like `set123.xml`. The XML file **wacsxmlout** creates contains the set information, all associated download records, associations and idmap records for all of the models featured in the set. Additionally, unless `--noimages` was specified, it will contain the thumbnail icon, official icon and any additional icons related to the set.

wacsxmlout *DOES NOT* export the set itself and will need to be recombined with the appropriate zip or movie file when it's loaded into the destination WACS system. These need to be downloaded seperately.

Sample XML files can be downloaded from the resources section on our demonstration site www.pinkmetallic.com as they become available.

See Also

wacsimport(1), wacsxmlin(1), wacsxmlout(1)

Name

wacsxmlin — imports set details from a WACS XML export file

Synopsis

```
wacsxmlin [modifiers] XML_file_name
```

Modifiers

--default
specifies that the set should be stored in the calculated default location for this WACS server instead of it's location on the original server.

--clone
specifies that this should be an *EXACT* copy of the original - same location, set number, etc. Useful in creating clone web servers for resilience, etc.

--asnew
tells **wacsxmlin** to mark this as a newly created set so it appears in the new sets/videos index.

Description

The **wacsxmlin** command is a tool to import a WACS set export XML file containing information about a set to allow it to be transported between different WACS installations. **wacsxmlin** is invoked with the name of the set XML file created previously by the **wacsxmlout** command on another WACS installation. For a set numbered 123 on the originating site, the inputs to **wacsxmlin** would be `set123.xml` and `set123.zip`. This set will normally recieve a new set number on importation into the new WACS installation.

Sine the XML file **wacsxmlin** reads contains the set information, all associated download records, associations and idmap records for all of the models featured in the set, **wacsxmlin** does it's best efforts to find and associate the set being added to those models who feature in it on the new WACS installation. If the export file contained them, **wacsxmlin** will unpack the thumbnail icon, official icon and any additional icons related to the set.

Sample XML files can be downloaded from the resources section on our demonstration site www.pinkmetallic.com as they become available and can be used to import our sets into your own installation. Since these sets are released under the Creative Commons license, you are free to import them into your WACS server and even include them on a commercial site if you wish to, providing the PinkMetallic.com branding remains intact. We can provide proof of age documentation to US-based sites in order to conform to 18 USC 2257 for a small fee.

See Also

wacsimport(1), wacsexport(1), wacsxmlout(1)