

Remy Bien
Sebastiaan Groot
Wouter Miltenburg
Koen Veelenturf
Friday 24 January 2014

yaNMS System Management Documentation



Thank you for using yaNMS and welcome to the yaNMS community. We will use our GitHub page¹ for new updates and documentation, so please check if there is a new version of this document available at our GitHub page so you are sure you are reading up-to-date information.

This product is open source and because of this the community has only one simple rule: “This product is, and will be, always open source. When one wants to modify the product he or she is free to do so, but he or she must also publish these modification so that the community can profit from it.” For more information please read our license model².

¹ Our GitHub page can be found using the following link: github.com/yanms

² Our license model can be found using the following link: https://github.com/yanms/nms_code/tree/master/LICENSE

Table of contents

Table of contents	2
1. Devices	3
1.1 Add Device	3
1.2 List and Manage Devices	3
1.3 Manage Devices	4
1.4 Modify Device	5
1.5 Device History	6
2. Application Administration	7
2.1 ACL	7
2.1.1 ACL: Groups	8
2.1.2 ACL: Users	8
2.1.3 ACL: Devices	9
2.2 History	9
3. Manage Generic Devices	10
3.1 Generic Device	10
3.1.1 Generic Device Template	10
3.1.2 OS per generic device template	11
3.2 Update data listed in “Manage generic device” section	11
3.3 Delete data listed in “Manage generic device” section	11
4. Installation yaNMS	12
4.1 Apache Configuration	12
4.2 Serving Files	13

1. Devices

In the left-side menu, the middle button directs the user to the Devices page. From here, devices can be added, generic devices can be managed and added devices can be managed. For generic device management, see chapter 2.

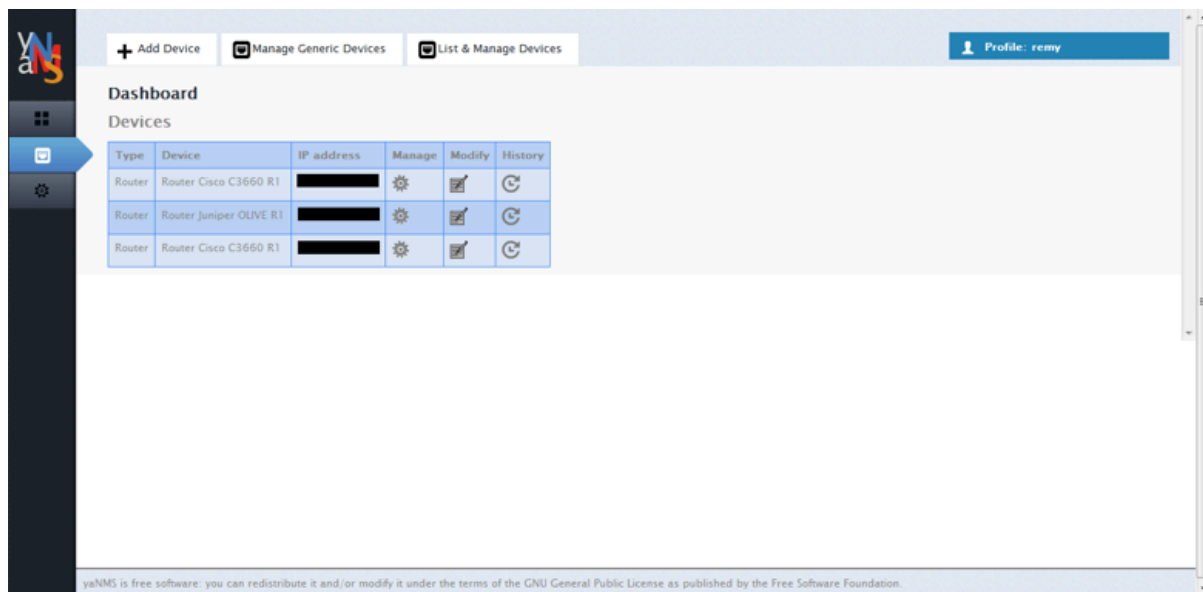
1.1 Add Device

Adding a device can be done with a few quick steps, once a device template (generic device) has been added. As seen in the figure below, the user first selects a device type and vendor, after which the device model list gets filled with matching models. After selecting the operating system and preferred protocol, all of the main information about the device is filled in. Under “Network settings”, the IP version, address and port are to be entered. Under “Login settings”, credentials used to log into the device have to be specified. The username and password are used to log in to the device using the specified preferred protocol. The privileged password is used, for some devices, to enter a privileged or elevated mode to execute certain commands. The last option, “Add to group”, is used to specify the device group in which the device will be placed initially.

The screenshot shows the 'Add Device' form in the yaNMS interface. The top navigation bar includes 'Add Device', 'Manage Generic Devices', and 'List & Manage Devices'. The user profile 'remy' is visible in the top right. The left sidebar contains a menu with a gear icon. The main form area is titled 'Dashboard' and 'Add Device'. It contains several sections: 'Type' (Router), 'Vendor' (Cisco), 'Model' (C3660 R1), 'OS' (Router Cisco C3660 R1 default - IOS 12.4-15), and 'Preferred protocol' (SSH2). Below these are 'Network settings' (IPv4 selected, IP address: 10.0.0.1, Port: 22), 'Login Settings' (Username: <username>, Password: *****, Privileged EXEC (enable) password: *****, Add to group: dev example), and an 'Add Device' button at the bottom.

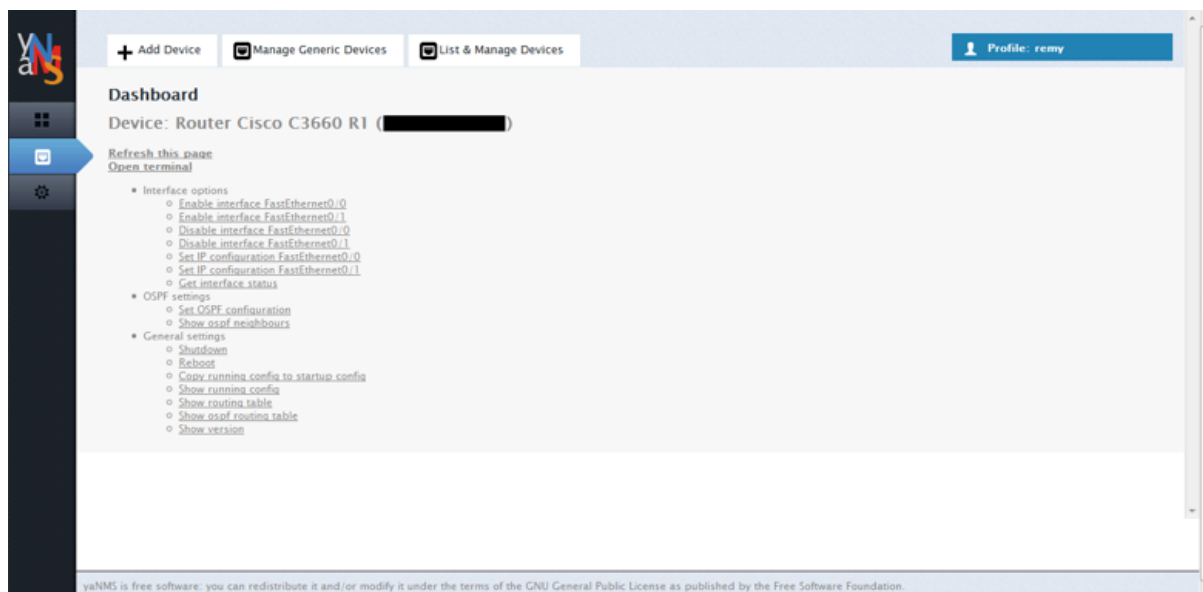
1.2 List and Manage Devices

On the “List & Manage Devices” page (see figure on next page), all the devices you have rights for are displayed in a single list. The device type, vendor and model are displayed here, as well as the IP address and buttons to the “Manage”, “Modify” and “History” page for each device.

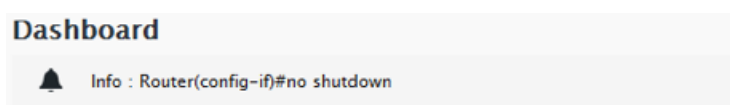


1.3 Manage Devices

After clicking on the “Manage” button on a device, connection is made with the device to obtain a list of available interfaces, after which the Manage device page is displayed (see figure below). This page contains links used to execute specific tasks on the device, as well as a “Refresh this page” and “Open terminal” button.

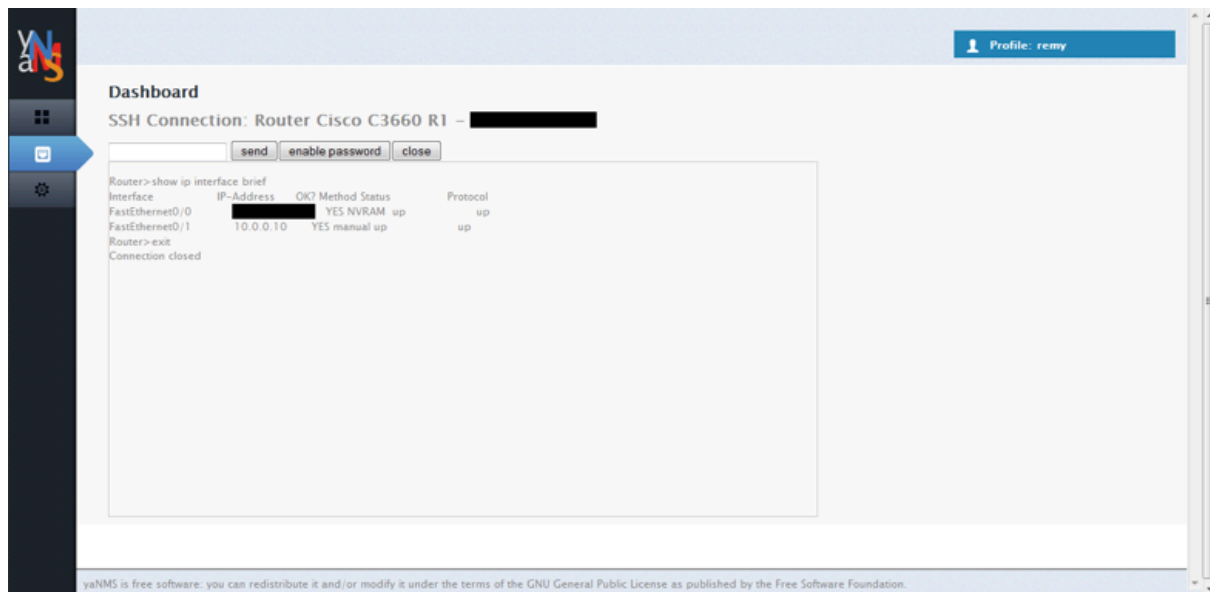


Once a task-link is pressed, a command or set of commands is sent to the device in order to perform the specified task. Once the task is completed, a message containing information returned by the device is displayed:



Clicking the “Refresh this page” button is useful when the application failed to make a connection with the device, causing all interface-specific tasks to not be displayed, as this forces the connection between the application and the device to be reestablished. It is also useful when cached information about the device, such as the XML file, has been changed, but the page isn’t displayed these changes properly.

Finally, the “Open terminal” button directs you to a browser terminal screen (see figure below), allowing direct access to the devices console. Login credentials are already known to the application, so the login process is done automatically. If a privileged-mode password is required, the button “enable password” can be used to transmit the privileged-mode password to the device, without having to type it in yourself.



1.4 Modify Device

The “Modify device” page (see figure on next page) is identical to the “Add device” page discussed earlier, but is used to modify an existing device. All but the password fields are already filled in on arrival, so only the values you want changed, as well as the passwords, need to be filled in.

Dashboard

Modify Device

Type: Router
 Vendor: Cisco
 Model: C3660 R1
 OS: Router Cisco C3660 R1 default - IOS 12.4-15
 Preferred protocol: SSH2

Network settings

☒ IPv4
☐ IPv6
 IP address: 10.0.0.1
 Port: 22

Login Settings

Username: <username>
 Password: *****
 Privileged EXEC (enable) password: *****

yaNMS is free software: you can redistribute it and/or modify it under the terms of the GNU General Public License as published by the Free Software Foundation.

1.5 Device History

On the “Device history” page (see figure below), all tasks performed on the device can be recalled. It is useful to note that not the abstracted tasks, as seen on the “Manage device” page are logged here, but the actual commands sent to the device over the specified protocol. This makes the “Device history” page useful for accountability and traceback of events, but also as a debug mechanism in case you suspect the created task abstractions contain bugs.

Dashboard

Device History

Page 1 of 2 [next](#)

History id	Action	Action type	Device	User affected	User performed task	Time
96	[dev1] exit	Manage device	Router Cisco C3660 R1 - [REDACTED]	None	remy	Jan. 24, 2014, 1:34 p.m.
95	[dev1] show ip interface brief	Manage device	Router Cisco C3660 R1 - [REDACTED]	None	remy	Jan. 24, 2014, 1:34 p.m.
94	[dev1] exit	Manage device	Router Cisco C3660 R1 - [REDACTED]	None	remy	Jan. 24, 2014, 1:33 p.m.
93	[dev1] show ip interface brief	Manage device	Router Cisco C3660 R1 - [REDACTED]	None	remy	Jan. 24, 2014, 1:33 p.m.
92	[dev1] exit	Manage device	Router Cisco C3660 R1 - [REDACTED]	None	remy	Jan. 24, 2014, 1:33 p.m.
91	[dev1] disable	Manage device	Router Cisco C3660 R1 - [REDACTED]	None	remy	Jan. 24, 2014, 1:32 p.m.
90	[dev1] end	Manage device	Router Cisco C3660 R1 - [REDACTED]	None	remy	Jan. 24, 2014, 1:32 p.m.
89	[dev1] no shutdown	Manage device	Router Cisco C3660 R1 - [REDACTED]	None	remy	Jan. 24, 2014, 1:32 p.m.
88	[dev1] interface FastEthernet0/1	Manage device	Router Cisco C3660 R1 - [REDACTED]	None	remy	Jan. 24, 2014, 1:32 p.m.
87	[dev1] configure terminal	Manage device	Router Cisco C3660 R1 - [REDACTED]	None	remy	Jan. 24, 2014, 1:32 p.m.
86	[dev1] enable	Manage device	Router Cisco C3660 R1 - [REDACTED]	None	remy	Jan. 24, 2014, 1:32 p.m.
85	[dev1] show ip interface brief	Manage device	Router Cisco C3660 R1 - [REDACTED]	None	remy	Jan. 24, 2014, 1:32 p.m.
84	[dev1] disable	Manage device	Router Cisco C3660 R1 - [REDACTED]	None	remy	Jan. 24, 2014, 1:31 p.m.
83	[dev1] end	Manage device	Router Cisco C3660 R1 - [REDACTED]	None	remy	Jan. 24, 2014, 1:31 p.m.
82	[dev1] shutdown	Manage device	Router Cisco C3660 R1 - [REDACTED]	None	remy	Jan. 24, 2014, 1:31 p.m.
81	[dev1] interface FastEthernet0/1	Manage device	Router Cisco C3660 R1 - [REDACTED]	None	remy	Jan. 24, 2014, 1:31 p.m.

2. Application Administration

All text regarding to the application administrator chapter must be inserted here.

2.1 ACL

In this chapter ACL will be described and information is given how an administrator can configure ACL to make it suitable in his / her situation.

When you visit the ACL pages you see three menus in the menubar:

- ACL Groups
- ACL Users
- ACL Devices

In the ACL Groups section you will find all groups that exist in the application. Here you will see two columns named User groups and Device groups.

In the ACL Users section you will find all users that exist in the application.

In the ACL Devices section you will find all devices that exist in the application.

Dashboard
Manage group: dev:1st floor

Devices
☒ Router CISC0 C3660 - 145.92.14.145

Device Group Rights
☒ List Devices
☒ Manage Devices
☒ Change Devices

INHERIT OPTIONS

	Users in Group
<input checked="" type="checkbox"/> dev:default	root example

INDIVIDUAL USERS

User	Added to group?
<input type="checkbox"/> root	✗
<input type="checkbox"/> example	✗

Dashboard
Manage group: usr:manager

User Rights
☐ Add user
☐ Change user
☐ Delete user
☐ List user

Group Rights
☐ Add group
☐ Change group
☐ Delete group
☐ List group and can add user/device to group

Device Rights
☐ Add device
☐ Delete device

Generic Device Rights
☐ Add generic device
☐ Change generic device
☐ Delete generic device

INHERIT OPTIONS

	Users in Group
<input checked="" type="checkbox"/> usr:staff	example
<input type="checkbox"/> usr:admin	

INDIVIDUAL USERS

User	Added to group?
<input type="checkbox"/> root	✗
<input type="checkbox"/> example	✗



When you have just installed yaNMS you need to login as root. Configure ACL as soon as possible and when you have created an administrator account change the password for your root account. Do not use root to manage devices, it is only used as a last resort when all accounts are locked out.

2.1.1 ACL: Groups

There are two different type of groups in yaNMS namely: Device groups and User groups. The following user or device rights exist in yaNMS:

- Device group:
 - Device group rights:
 - List Devices
 - Manage Devices
 - Change Devices
- User group:
 - User rights:
 - Add user
 - Change user
 - Delete user
 - List user
 - Group rights:
 - Add group
 - Change group
 - Delete group
 - List group and can add user/device to group
 - Device rights:
 - Add device
 - Delete device
 - Generic Device rights:
 - Add generic device
 - Change generic device
 - Delete generic device

„Only users with clearance can access, manage and change devices or users”

As you can see, these two groups are completely segregated. User groups are specifically for user rights and device groups are specifically for group rights. When one wants to manage a device it needs to be added to a device group. A device group contains users, devices and privileges (i.e. rights).

2.1.2 ACL: Users

When a user is added and not assigned to a group it cannot do much. Only users with clearance can access, manage and change devices or users. In order to make a new user useful, for managing devices, an administrator has to give the user specific privileges / rights. A user can be added in two ways:

1. When a webpage is visited, and when a user is not logged in, the user will see the login page. On this page the user can decide to register itself and an administrator has to activate the account before the user can access other parts of the systems.

The screenshot shows the 'Dashboard' and 'ACL MANAGE USER' section. It includes a table of existing users and a form to add a new user. Below the form is the 'Manage User: example' section, which contains a table of permissions for different user groups.

Username	Last Login	First Name	Last Name	E-mail address	Active?	Date joined
example	Jan 10, 2014, 4:01 p.m.	example	example	example@exam.at	True	Jan 10, 2014, 4:01 p.m.

Below the table is a form to add a new user with fields for Username, First Name, Last Name, E-mail address, Active User?, Password, Password Check, and a checkbox for 'Lock user from access'.

The 'Manage User: example' section shows a table of permissions for different user groups:

Group	Users	Permissions
der default	root	
der staff		Can add group Can change group Can delete group Can list (own) group Can add device Can delete device Can add gen. dev Can delete gen. dev
der admin		Can add group Can change group Can delete group Can list (own) group Can add user Can change user Can delete user Can list users Can add device Can delete device Can add gen. dev Can delete gen. dev
der Can Root		
der manager		

2. When you go to the ACL page and click on „ACL users” a button appears (Add User). When you click on this button you can add a new user.

The difference between the two methods is that in the first situation the user account is not activated (an administrator has to modify the account in order to activate it) and in the second situation the user account is activated by default.

2.1.3 ACL: Devices

Using the ACL Devices system module you can administer all your devices that are known in the application. It is possible to easily add a device to multiple groups, but you can also see the history of a device. When a device must be deleted it is possible to do so using the ACL Device system module.

2.2 History

From the moment yaNMS is installed, everything is logged and recorded in the database. When a misconfiguration happens it is possible to see which users are affected and who caused the misconfiguration. It is also possible to see when someone has logged out and also when someone logged in.

3. Manage Generic Devices

This page will describe the “Manage generic devices” system module. Users that have the appropriate clearance can access the Manage Generic Devices through Devices > Manage Generic Devices (`<ip>/devices/gen_dev/manage`).

3.1 Generic Device

A generic device, is in a sense, just a template for a specific device. For example, when you have ten routers from a specific brand that are all the same, then you don not want to provide the same information ten times. This is the reason why generic devices are introduced, providing all the information that is the same, and you can create a template that is the same for all those devices.

In essence, there are two templates that you must create whenever you want to support a new device. There is the “OS per generic Device Template” and “Generic Device Template”. This is explicitly done, because there could be a case where a device supports multiple Operating Systems and this distinction, in a device an OS template, can support such cases.

3.1.1 Generic Device Template

To add a “Generic Device Template” for a bunch of devices you must provide the following information:

- Type
 - Typename of those devices
- Vendor
 - Vendor name of those devices
- Model
 - Model name of those devices
 - Version name of those devices
- XML
 - XML-file for those devices
 - Location of the file

When all the information is provided there is only one last thing that needs to be done. Create a link/relationship between all this information and you are done. You can create this relationship in the top left corner of the “Manage Generic Devices” page of yaNMS. When a device is created, and the “type” and “vendor” fields are specified, yaNMS will automatically list the correct models according to the relationships that you have just created.

3.1.2 OS per generic device template

To add an “OS per generic device” template for a bunch of devices you must provide the following information:

- OS
 - Vendor (Foreign key to vendor that is noted earlier in this list)
 - OS type
 - Name
 - Build
 - Short info
- OS per generic device (the relation between a specific OS and a generic device)
 - Device template (i.e., generic device template)
 - OS



In order to create an “OS per generic device” template you must first create a “generic device” template.

3.2 Update data listed in “Manage generic device” section

When someone has provided the wrong information, it is possible to change this information. Just click on the button below and the user will be redirected to the correct page to change the provided information:



3.3 Delete data listed in “Manage generic device” section

When a user has clearance to delete “Manage generic device” instances, it can do so by selecting such an instance and click on the button “Delete”. yaNMS will check if a device is linked to such an instance, if this is not the case it will delete the instance from yaNMS. Whenever a user wants to delete a instance and a device is linked to this instance, yaNMS will simply refuse to delete this instance. Otherwise a device would miss some essential information or relationships could be corrupted.



Cannot delete some instances of x because there is still a reference.

This is one of the errors, that a user will see, when it tries to delete an instance which has a link with a device or template.

4. Installation yaNMS

In order for Django to work on Apache, you need `mod_wsgi`. Since our Django application is based on python 3.3, we need `mod_wsgi` in combination with `libpython3.3`. Installation of Apache, `mod_wsgi` and `libpython` are different for each operating system, so the instructions here are only valid for CentOS 6.x.

```
yum groupinstall "Development tools"
yum install zlib-devel bzip2-devel openssl-devel ncurses-devel sqlite-devel readline-devel tk-devel
yum install wget
wget http://python.org/ftp/python/3.3.0/Python-3.3.0.tar.bz2
tar xf Python-3.3.0.tar.bz2
cd Python-3.3.0
./configure --prefix=/usr/local --enable-shared
make && make altinstall
cd /
wget https://modwsgi.googlecode.com/files/mod_wsgi-3.4.tar.gz
tar xvfz mod_wsgi-3.4.tar.gz
cd mod_wsgi-3.4
yum install httpd-devel
LD_LIBRARY_PATH=/usr/local/lib
-ln /usr/local/lib/libpython3.3m.so.1.0 /usr/lib/libpython3.3m.so.1.0
ln -s /usr/local/lib/python3.3 /usr/lib/libpython3.3m.so.1.0
export LD_LIBRARY_PATH=/usr/local/lib
./configure --with-apxs=/usr/sbin/apxs --with-python=/usr/local/bin/python3.3
edit Makefile and change: LDLIBS = -lpython3.3 to LDLIBS = -lpython3.3m

make
make install
ln -s /usr/local/lib/libpython3.3m.so.1.0 /usr/lib64/libpython3.3m.so.1.0
service httpd restart
```

After this, a database needs to be configured for the Django application. In the case of `sqlite3`, a writable directory needs to be available for the webserver. Create a folder 'db' on the location of `nms.db`, make it writable and move the file `nms.db` into the db folder. Then edit your `nms_root/settings.py` file to point at the right database file.

4.1 Apache Configuration

In order to compile the small shared-memory C library, navigate to the path where you cloned the yaNMS code repository. Go to `.../nms/c/` and run "make" to compile the shared object.

(Based on django documentation found here: <https://docs.djangoproject.com/en/1.5/howto/deployment/wsgi/modwsgi/>)

Once you've got `mod_wsgi` installed and activated, edit your Apache server's `httpd.conf` file and add the following. If you are using a version of Apache older than 2.4, replace `Require` all granted with `Allow` from all:

```

WSGIScriptAlias / /path/to/mysite.com/mysite/wsgi.py
WSGIPythonPath /path/to/mysite.com
<Directory /path/to/mysite.com/mysite>
<Files wsgi.py>
Order deny,allow
Require all granted
</Files>
</Directory>

```

The first argument in the `WSGIScriptAlias` line is the base URL path you want to serve your application at (/ indicates the root url), and the second is the location of a “WSGI file” – see below – on your system, usually inside of your project package (mysite in this example). This tells Apache to serve any request below the given URL using the WSGI application defined in that file.

The `WSGIPythonPath` line ensures that your project package is available for import on the Python path; in other words, that `import mysite` works. The `<Directory>` piece just ensures that Apache can access your `wsgi.py` file.

4.2 Serving Files

Django does not serve files itself; it leaves that job to whichever web server you choose. We recommend using a separate Web server – i.e., one that’s not also running Django – for serving media. If, however, you have no option but to serve media files on the same Apache VirtualHost as Django, you can set up Apache to serve some URLs as static media, and others using the `mod_wsgi` interface to Django. This example sets up Django at the site root, but explicitly serves `robots.txt`, `favicon.ico`, any CSS file, and anything in the `/static/` and `/media/` URL space as a static file. All other URLs will be served using `mod_wsgi`:

```

Alias /robots.txt /path/to/mysite.com/static/robots.txt
Alias /favicon.ico /path/to/mysite.com/static/favicon.ico
AliasMatch ^(/[^\/*]*\.css) /path/to/mysite.com/static/styles/$1
Alias /media/ /path/to/mysite.com/media/
Alias /static/ /path/to/mysite.com/static/
<Directory /path/to/mysite.com/static>
Order deny,allow
Allow from all
</Directory>
<Directory /path/to/mysite.com/media>
Order deny,allow
Allow from all
</Directory>
WSGIScriptAlias / /path/to/mysite.com/mysite/wsgi.py
<Directory /path/to/mysite.com/mysite>
<Files wsgi.py>
Order allow,deny
Allow from all
</Files>
</Directory>

```