Augustus: Installation Guide

Release 0.5.0.0

Open Data Group

December 14, 2011

Contents

1 Augustus: PMML model producer and consumer

Augustus is a PMML 4.1-compliant scoring engine designed to work with segmented models. It is intended for use with statistical and data mining models: providing producers and consumers for Baseline, Clustering, Rulesets and Tree models.

Augustus is written in Python and is freely available under the Apache License Version 2.0.

See the Augustus project site, http://code.google.com/p/augustus/, to download Augustus, read release notes, or report any issues.

Augustus is maintained by Open Data Group.

2 Installation

Please refer to `INSTALL` which ships with the Augustus release if you have any questions regarding installation and do not want to read this document. The file is brief.

Operating system compatibility

Augustus is written in Python; it should run on any system with the required Python environment. Augustus 0.5 has been tested on:

- Ubuntu Releases 10.04 - 11.04
- Debian 4 and 5 (etch and lenny)
- AIX 5

Older versions have been tested and used on Windows XP, Open Solaris, and earlier releases of Ubuntu.

Platform-specific instructions are available: see the sections *Installation on Ubuntu*, *Installation on RHEL 5.5*, *Installation on Mac (Darwin)*, and *Installation on Windows*

Obtaining Augustus

Download

The most recent Augustus release can be downloaded from http://code.google.com/p/augustus/downloads/. Search for “release”. Unpack the distribution in the directory of your choice and expand. For example:

```
$ cd /home/odg
$ wget http://augustus.googlecode.com/files/Augustus-<VERSION>.tar.gz
$ tar zxvf Augustus-<VERSION>.tar.gz
```

As of release 0.5, the archives only contain the scoring engine, `augsutus-scoringengine`, and the examples, `augustus-examples`, and not the tests, `augustus-tests`. If you want the tests, then you must check them out from source control.

Source checkout

It is also possible to check out a read-only copy of the trunk via Google Code’s Subversion:

```
$ cd /home/odg
$ svn checkout \
    http://augustus.googlecode.com/svn/trunk/ augustus-trunk-read-only
```

Branches and tagged releases are available from source control as well as downloads.

Checking out the trunk provides the scoring engine, tests, and examples. If you only want the scoring engine, then use:

```
$ cd /home/odg
$ svn co http://augustus.googlecode.com/svn/trunk/augustus-scoringengine \
    augustus-scoringengine
```

This saves a great deal of disk space.

Dependencies

Required

Python

Augustus is written in Python. Python can be downloaded from http://www.python.org/. We strive to test Augustus against Python 2.5, 2.6, and 2.7 and any issues encountered running against those version should be reported as bugs. Augustus is not supported for any other versions.

NumPy

Required. NumPy 1.2.1 or 1.3.0 are suggested, and Augustus is tested against those versions. NumPy 1.0 will not work because the `numpy.ma` module is in a different location. NumPy through version 1.5 should work. NumPy is available for many operating systems via their package manager. If it is not available on your system, it can be obtained from: http://www.scipy.org/Download

Setup (general)

If you have downloaded or checked out Augustus and Python and NumPy are already installed, then all that is left is to make Augustus accessible via the **PYTHONPATH** and **PATH** environment variables. This can be done by 1) manually adding the Augustus directory to each of the above environment variables, or 2) running the package's `setup.py` script to install Augustus in default locations. Instructions for both follow below.

Using `setup.py`

Augustus comes with a setup file. By default, it installs into locations already in the environment variables **PYTHONPATH** and **PATH**. To use it:

```
$ cd augustus-scoringengine
$ sudo python setup.py install
```

For options using `setup.py`, type `python setup.py --help`. If you do not have `sudo` privileges, an alternate install directory can be passed to `setup.py`. By default, if using Python 2.6, `python setup.py install` installs Augustus under the directories:

```
/usr/local/lib/python2.6
/usr/local/bin
```

As an example, installing Augustus using `setup.py` on Ubuntu creates:

```
/usr/local/lib/python2.6/dist-packages/augustus
/usr/local/lib/python2.6/dist-packages/Augustus-0.5.0.0.egg-info
```

and places under `/usr/local/bin` the files:

```
AnalysisWorkflow    PmmlSplit
Augustus            ScoresAwk
munge               ScoresDiff
PmmlDiff            ScoresDiffFast
PmmlSed             unitable
```

Setting environment variables

If a tarball is unpacked, the source checked out, or `setup.py` was sent to a directory not in the System paths, then the variables **PYTHONPATH** and **PATH** need to be updated.

For example, if `augustus-scoringengine` is in `/home/odg/`, then add these lines to your `.bashrc` or equivalent file:

```
PYTHONPATH=$PYTHONPATH:/home/odg/augustus-scoringengine
PATH=$PATH:/home/odg/augustus-scoringengine/augustus/bin
```

Platform-specific setup

Installation on Ubuntu

This section shows the installation of Augustus trunk on a new Ubuntu 10.04.3 LTS installation with up-to-date packages.

Ubuntu 10.04.3 uses Python 2.6.5, NumPY 1.3.0 and subversion 1.6.6 and all are available from APT. These default versions are used when acquiring the packages in the session below.

The example starts with modifying the bash prompt to include the timestamp– the entire install takes three minutes.

```
odg@ubuntu:~$ export PS1=’\u@\t:\w\$ ’
odg@19:58:32:~$ python --version
Python 2.6.5
odg@19:58:41:~$ python -c "import numpy"
Traceback (most recent call last):
  File "<string>", line 1, in <module>
ImportError: No module named numpy
odg@19:58:52:~$ sudo apt-get install python-numpy
Reading package lists... Done
Building dependency tree
Reading state information... Done

[output omitted ...]

Need to get 4,784kB of archives.
After this operation, 14.8MB of additional disk space will be used.
Do you want to continue [Y/n]? Y

[output omitted ...]

odg@19:59:14:~$ python
Python 2.6.5 (r265:79063, Apr 16 2010, 13:09:56)
[GCC 4.4.3] on linux2
Type "help", "copyright", "credits" or "license" for more information.
```

```
>>> import numpy
>>> numpy.__version__
’1.3.0’
>>> exit()
```

```
odg@19:59:38:~$ svn
The program ’svn’ is currently not installed.  You can install it by typing:
sudo apt-get install subversion
odg@19:59:43:~$ sudo apt-get install subversion
Reading package lists... Done
Building dependency tree
Reading state information... Done

[output omitted ...]

After this operation, 6,836kB of additional disk space will be used.
Do you want to continue [Y/n]? Y

[output omitted ...]
```

```
odg@20:00:05:$ svn checkout \
> http://augustus.googlecode.com/svn/trunk/augustus-scoringengine augustus-scoringengine

[output omitted ...]

Checked out revision 527.

odg@20:00:49:~$ cd augustus-scoringengine/
odg@20:00:59:~/augustus-scoringengine$ sudo python setup.py install
running install

[output omitted ...]

running install_egg_info
Writing /usr/local/lib/python2.6/dist-packages/Augustus-0.5.0.pre.egg-info

odg@20:01:07:~/augustus-scoringengine$ cd /tmp/
odg@20:01:13:/tmp$ python -c "import augustus"
odg@20:01:21:/tmp$
```

With minor differences in the default versions of Python and Subversion, this installation process is the same with Ubuntu 9.10, 10.10, and 11.04.

Installation on RHEL 5.5

This section shows the installation of Augustus trunk from source on a Amazon EC2 instance. The steps were performed using:

```
AMI Id: ami-2632cc4f
Zone:   us-east-1c
Type:   m1.large
Owner:  03269352882
```

The instance comes with the default version of Python for the OS installed and without NumPy:

```
# cat /etc/redhat-release
Red Hat Enterprise Linux Server release 5.5 (Tikanga)
# python -V
Python 2.4.3

# python -c "from numpy import *"
Traceback (most recent call last):
File "<string>", line 1, in ?
ImportError: No module named numpy
```

Note: Expired ssl certificates

Due to expired ssl certificates on the Red Hat side, Amzon instances for both RHEL 5 and 6 need to perform an update. For the zone / release we used, this is:

# rpm -Uhv # http://redhat-clientconfig-us-east-1.s3.amazonaws.com/rh-amazon-rhui-client-2.2.16-1.el5.noarch.rpm
# yum clean all

This is temporary an more information is available at:

https://forums.aws.amazon.com/thread.jspa?threadID=76738&tstart=0&messageID=280829

Step 1. Install Python 2.6

Check for python26 packages from the package manager:

```
# sudo yum list *python26*

... [omitted]

Error: No matching Packages to list
```

Add EPEL to the list of hosts:

```
# su -c 'rpm -Uvh \
# http://download.fedora.redhat.com/pub/epel/5/x86_64/epel-release-5-4.noarch.rpm'
```

Check again for python26 packages from the package manager, it should now be available:

```
# sudo yum list python26
Loaded plugins: amazon-id, fastestmirror, rhui-lb, security
Loading mirror speeds from cached hostfile

... [omitted]

Available Packages
python26.x86_64    2.6.5-6.el    epel
```

Install it using yum:

```
# yum install python26.x86_64
Loaded plugins: amazon-id, fastestmirror, rhui-lb, security
Loading mirror speeds from cached hostfile

... [omitted]

Installed:
  python26.x86_64 0:2.6.5-6.el5

Dependency Installed:
  libffi.x86_64 0:3.0.5-1.el5    python26-libs.x86_64 0:2.6.5-6.el5

Complete!
```

Test the Installation, both Python versions should be available:

```
# python -V
Python 2.4.3

# python26 -V
Python 2.6.5
```

Step 2. Get NumPy from EPEL

Find the package and install it:

```
# yum list available | grep numpy.x86_64
python26-numpy.x86_64    1.5.1-5.el5    epel


# yum install python26-numpy.x86_64
Loaded plugins: amazon-id, fastestmirror, rhui-lb, security
Loading mirror speeds from cached hostfile
 * epel: mirror.cogentco.com

... [omitted]

Installed:
  python26-numpy.x86_64 0:1.5.1-5.el5

Dependency Installed:
  atlas.x86_64 0:3.8.3-1.el5    libgfortran.x86_64 0:4.1.2-51.el5

Complete!
```

Test the NumPy was installed for the correct version of Python. Augustus will not run with Python 2.4, so the NumPy installation has to be for Python 2.6:

```
# python -c "import numpy"
Traceback (most recent call last):
  File "<string>", line 1, in ?
ImportError: No module named numpy

# python26 -c "import numpy"
```

Step 3. Get Augustus and install it using `setup.py`:

```
# svn checkout \
    http://augustus.googlecode.com/svn/trunk/augustus-scoringengine augustus-scoringengine

... [omitted]

Checked out revision 527

# cd augustus-scoringengine/
# python26 setup.py install
running install
running build

... [omitted]

running install_egg_info
Writing /usr/lib/python2.6/site-packages/Augustus-0.5.0.0-py2.6.egg-info
```

Verify the Installation:

```
# cd /tmp
# python26
Python 2.6.5 (r265:79063, Feb 28 2011, 21:55:45)
[GCC 4.1.2 20080704 (Red Hat 4.1.2-50)] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>> from augustus import *
>>>
```

Augustus is now installed.

Installation on Mac (Darwin)

Mac OS X has shipped with Python and NumPy installed since version 10.5 (Darwin version 9.0). A list of all Python packages that ship with Mac OS X is at:

http://developer.apple.com/library/mac/#documentation/Cocoa/Conceptual/RubyPythonCocoa/Articles/RubyPythonMacOSX.html.

This section shows the installation of Augustus trunk on Darwin version 10.7.0, which ships with **Python 2.6.1**, **NumPy 1.2.1**, and **Subversion 1.6.15**.

Download

Download a release of Augustus from the Google code website http://code.google.com/p/augustus/downloads/. When the download completes, click on the TAR icon to unpack the file where it was downloaded.

Open a Terminal shell by navigating to *Applications* → *Utilities* → *Terminal*, or by clicking on the 🔍 icon on the top right and searching for 'terminal'.

In the terminal, change directories to the Downloads folder, or the folder where Augustus was unpacked. If necessary, move the folder.

Checkout

Subversion ships with Mac OS X, and can be used from the command line to check out Augustus:

```
$ svn checkout \
  http://augustus.googlecode.com/svn/trunk/ augustus-trunk-read-only
```

If you prefer a GUI, the application **svnX** is available directly from Apple without charge.

Installation

The installation can now proceed according to the general instructions in the section *Obtaining Augustus*.

Note: If `setup.py` is used, the default install location is `/Library/Python/2.6/site-packages/`.

Note: If you have previously installed **Macports** using default settings, a file `.bash_profile` is in your home directory. Environment variables can be set there or in your `.bashrc` or equivalent shell file.

Installation on Windows

Note: When running the Augustus from the scripts in its `bin` directory, you have to specify the entire path to those files. There is a section on setting up the use of Python executable scripts for windows at the main Python site. See section 3.3.4 at: http://docs.python.org/using/windows.html

Dependencies

Python is available for Windows and can be downloaded from: http://www.python.org/getit/windows/ You can use any of Python 2.5, 2.6, or 2.7. Python devotes a section of its official documentation to Windows. It can be found at: http://docs.python.org/using/windows.html.

A Windows version of NumPy is available from sourceforge: http://sourceforge.net/projects/numpy/files/NumPy/.

Augustus

The most recent Augustus release can be downloaded from the google code site at http://code.google.com/p/augustus/downloads/list/ or can be checked out using Subversion. There are many Subversion clients for Windows, including TortoiseSVN (http://tortoisesvn.tigris.org/) and Subclipse (http://subclipse.tigris.org/), a plugin for Eclipse.

Installation

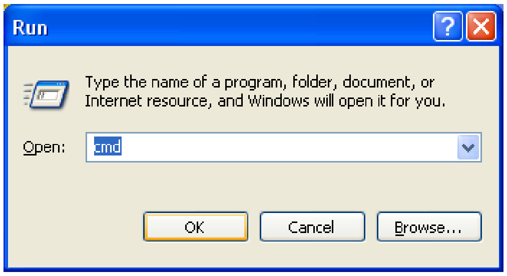
If Augustus was downloaded, open the archive.

The computer we are using runs Windows XP. Python 2.6 is installed at `C:\Python26` and the Augustus tarball was expanded to `C:\augustus-scoringengine`.

To install Augustus, either change into the installation directory and run the `setup.py` script, or modify the **Python-Path** environment variable.

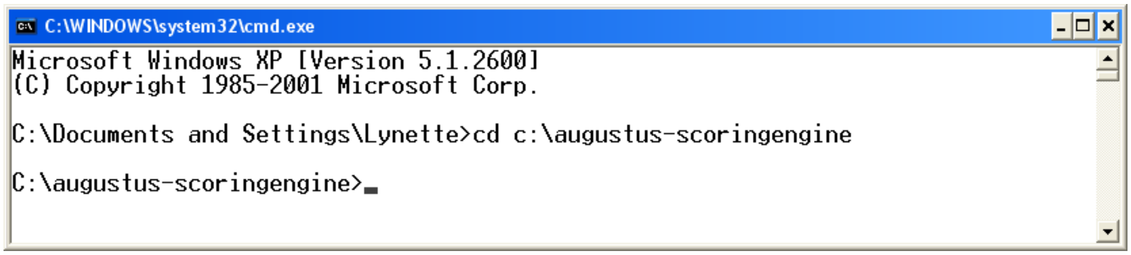
`setup.py`

First, open a command window: click on *Start* → *run*, and then type `cmd` in the dialog box.


Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

Open: cmd

Change directories to where Augustus was installed.

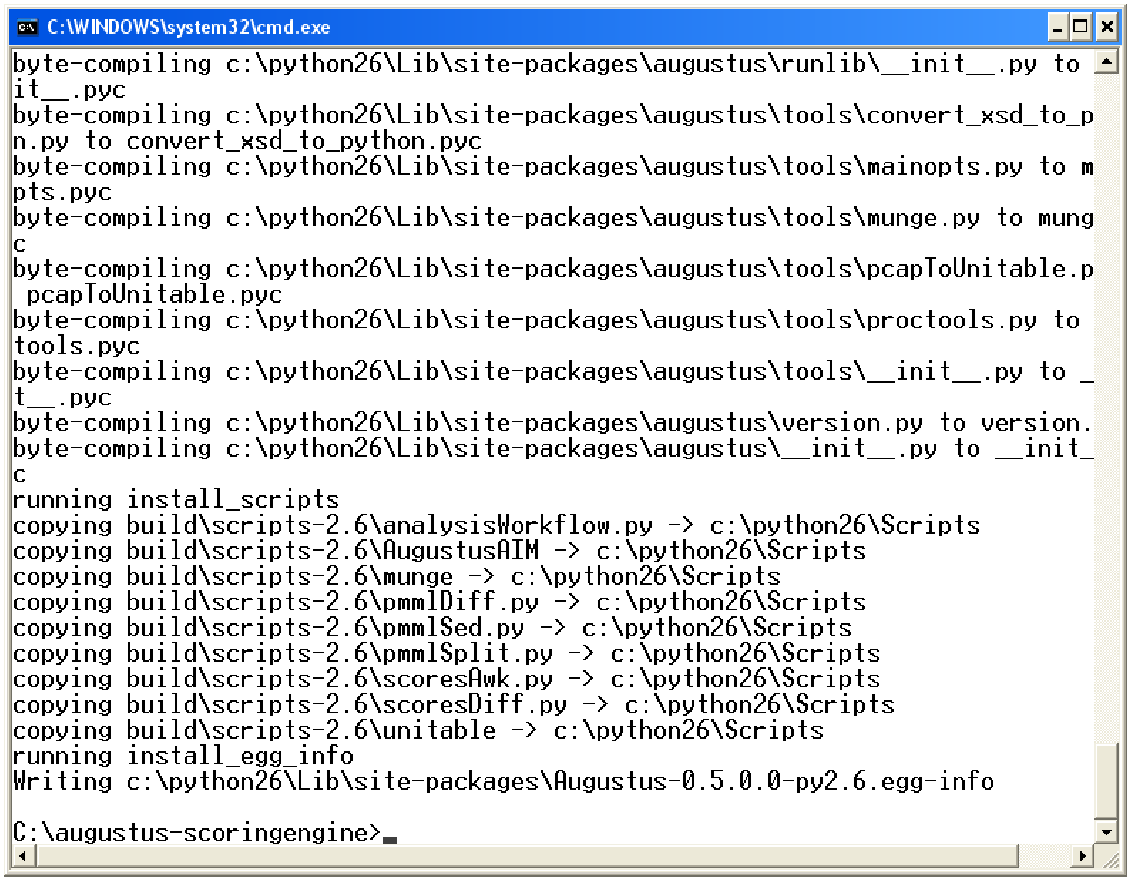


```
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.

C:\Documents and Settings\Lynette>cd c:\augustus-scoringengine

C:\augustus-scoringengine>_
```

Run `setup.py` with the install option by typing:

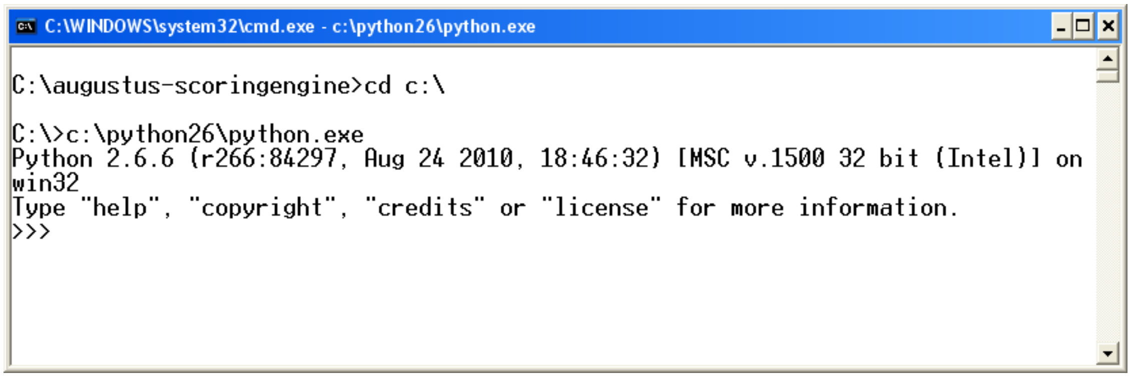
```
C:\Python26\python.exe setup.py install
```

The installation script should run. It places Augustus in the usual directories for Python.:



```
byte-compiling c:\python26\Lib\site-packages\augustus\runlib\__init__.py to
it__.pyc
byte-compiling c:\python26\Lib\site-packages\augustus\tools\convert_xsd_to_p
n.py to convert_xsd_to_python.pyc
byte-compiling c:\python26\Lib\site-packages\augustus\tools\mainopts.py to m
pts.pyc
byte-compiling c:\python26\Lib\site-packages\augustus\tools\munge.py to mung
c
byte-compiling c:\python26\Lib\site-packages\augustus\tools\pcapToUnitable.p
 pcapToUnitable.pyc
byte-compiling c:\python26\Lib\site-packages\augustus\tools\proctools.py to
tools.pyc
byte-compiling c:\python26\Lib\site-packages\augustus\tools\__init__.py to _
t__.pyc
byte-compiling c:\python26\Lib\site-packages\augustus\version.py to version.
byte-compiling c:\python26\Lib\site-packages\augustus\__init__.py to __init_
c
running install_scripts
copying build\scripts-2.6\analysisWorkflow.py -> c:\python26\Scripts
copying build\scripts-2.6\AugustusAIM -> c:\python26\Scripts
copying build\scripts-2.6\munge -> c:\python26\Scripts
copying build\scripts-2.6\pmmlDiff.py -> c:\python26\Scripts
copying build\scripts-2.6\pmmlSed.py -> c:\python26\Scripts
copying build\scripts-2.6\pmmlSplit.py -> c:\python26\Scripts
copying build\scripts-2.6\scoresAwk.py -> c:\python26\Scripts
copying build\scripts-2.6\scoresDiff.py -> c:\python26\Scripts
copying build\scripts-2.6\unitable -> c:\python26\Scripts
running install_egg_info
Writing c:\python26\Lib\site-packages\Augustus-0.5.0.0-py2.6.egg-info

C:\augustus-scoringengine>_
```

If Augustus is visible to Python, it should be possible to load Augustus modules, like unitable, from outside of the installation directory. Change back to `C:\` or another directory outside of Augustus, and start the Python interactive environment.

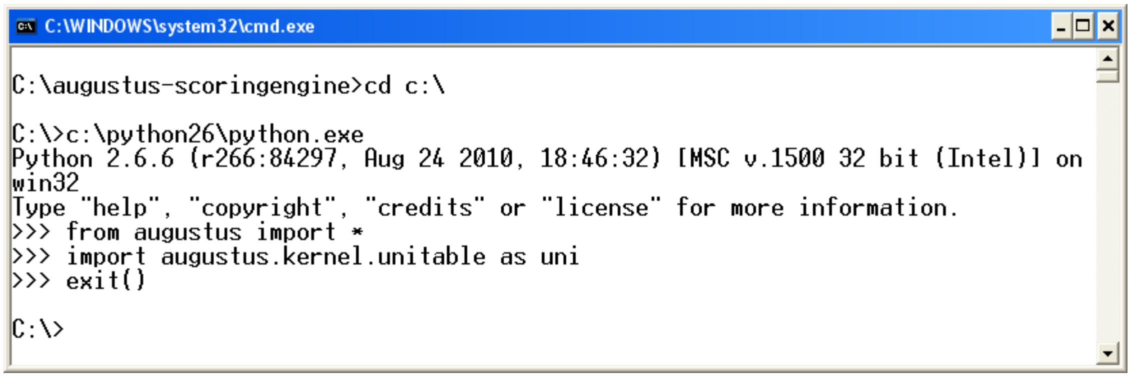


```
C:\augustus-scoringengine>cd c:\

C:\>c:\python26\python.exe
Python 2.6.6 (r266:84297, Aug 24 2010, 18:46:32) [MSC v.1500 32 bit (Intel)] on
win32
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

Then type:

```
>>> from augustus import *
>>> import augustus.kernel.unitable as uni
>>> exit()
```

If there are no error messages, Python was able to find Augustus and its modules. The session should look like the following:

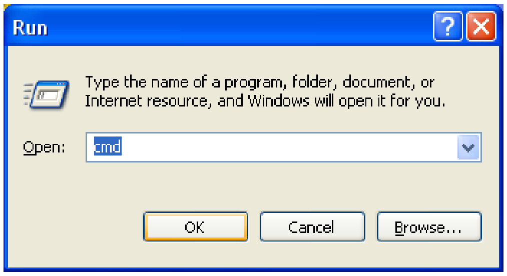


```
C:\augustus-scoringengine>cd c:\

C:\>c:\python26\python.exe
Python 2.6.6 (r266:84297, Aug 24 2010, 18:46:32) [MSC v.1500 32 bit (Intel)] on
win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from augustus import *
>>> import augustus.kernel.unitable as uni
>>> exit()

C:\>
```

Augustus is successfully installed.

Editing the PythonPath

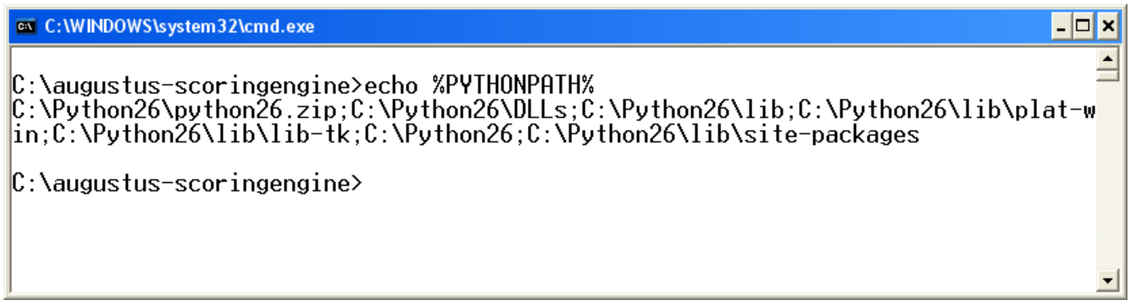
To use Augustus without running the installation script `setup.py`, edit the **PythonPath** environment variable to make Augustus's directories visible to Python. First, find out what is currently in the **PythonPath** environment variable. Open a command window: click on *Start* → *run*.



Type the name of a program, folder, document, or Internet resource, and Windows will open it for you.

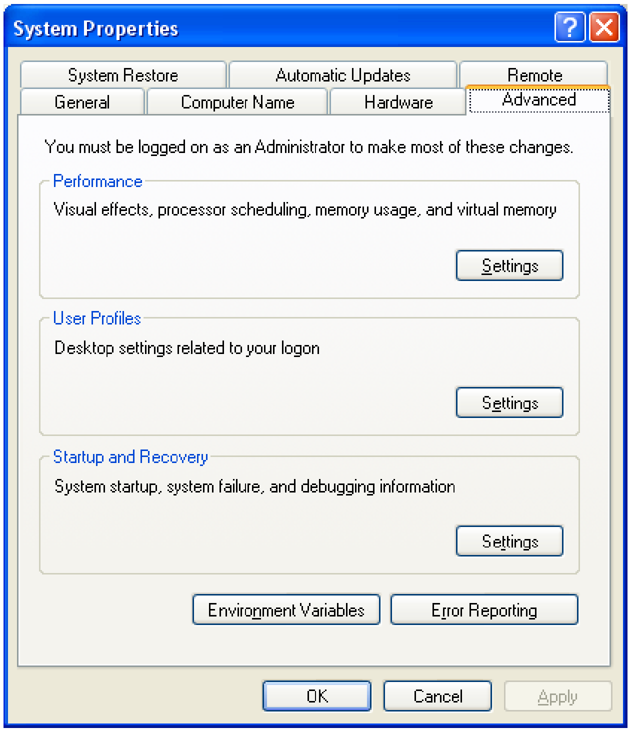
Open: cmd

You may or may not already have a **PythonPath** set up. To see an environment variable, enclose the variable names in percent (%) signs. For **PythonPath**, use the command `echo %PythonPath%`.



```
C:\augustus-scoringengine>echo %PYTHONPATH%
C:\Python26\python26.zip;C:\Python26\DLLs;C:\Python26\lib;C:\Python26\lib\plat-w
in;C:\Python26\lib\lib-tk;C:\Python26;C:\Python26\lib\site-packages

C:\augustus-scoringengine>
```

Use the System Properties widget to add or modify environment variables. Click on: *Start*, right-click on *My Computer*, and then click on *Properties*. The System Properties window will open. Click on the **Advanced** tab.



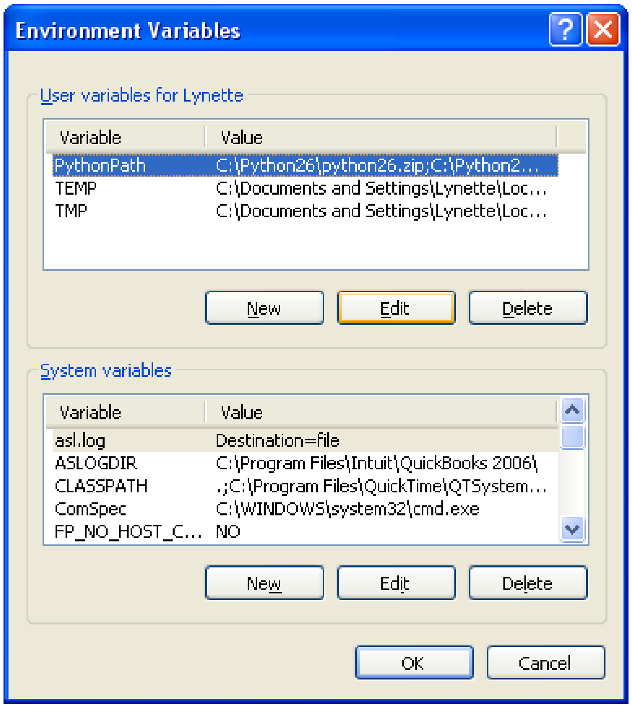
You must be logged on as an Administrator to make most of these changes.

Visual effects, processor scheduling, memory usage, and virtual memory

Desktop settings related to your logon

System startup, system failure, and debugging information

Near the bottom of the **Advanced** tab is the **Environment Variables** button. Click on it to edit both user-specific and system variables. The Environment Variables window appears:



User variables for Lynette

Variable	Value
PythonPath	C:\Python26\python26.zip;C:\Python2...
TEMP	C:\Documents and Settings\Lynette\Loc...
TMP	C:\Documents and Settings\Lynette\Loc...

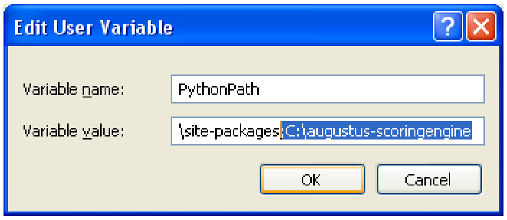
System variables

Variable	Value
asl.log	Destination=file
ASLOGDIR	C:\Program Files\Intuit\QuickBooks 2006\
CLASSPATH	.;C:\Program Files\QuickTime\QTSystem...
ComSpec	C:\WINDOWS\system32\cmd.exe
FP_NO_HOST_C...	NO

There are two sections; one for User variables and one for System variables. The User variables are defined for the individual user. Windows reloads these every time a new command window is open. The System variables are defined for all users on the system. Windows will reload these every time a new user logs in. If it is not appropriate to create **PythonPath** for all users, create it instead under *Environment Variables* → *User Variables* rather then *Environment Variables* → *System Variables*.

Edit the **PythonPath** environment variable to include the path to Augustus, or add a new variable if it does not exist. If you add a new variable when one already exists, the new varialbe overwrites the old.

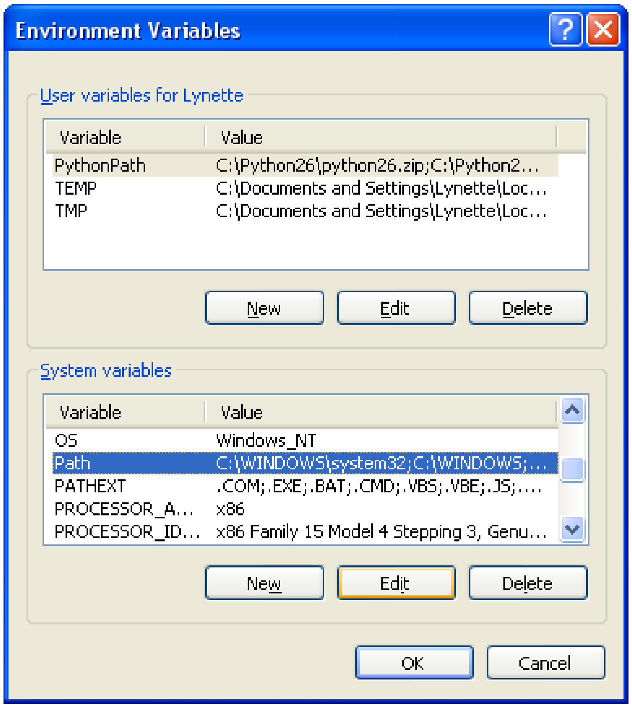
To add a new variable, click on **New** . To append to an existing path, click on **Edit**. Type a semicolon and then the path to where Augustus was installed. In this case, we add `;C:\augustus-scoringengine` to the path as shown:



Variable name: PythonPath

Variable value: \site-packages;C:\augustus-scoringengine

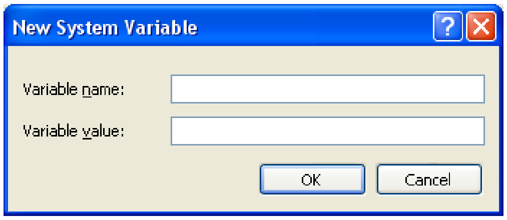
You may want to append the Python executable path to the environment variable **Path** to avoid having to type the entire path every time you invoke Python from the command line. **Path** is already listed in the System Variables for this machine, highlighted below.



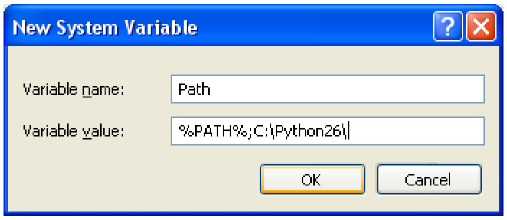
Variable	Value
PythonPath	C:\Python26\python26.zip;C:\Python2...
TEMP	C:\Documents and Settings\Lynette\Loc...
TMP	C:\Documents and Settings\Lynette\Loc...

Variable	Value
OS	Windows_NT
Path	C:\WINDOWS\system32;C:\WINDOWS;...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;....
PROCESSOR_A...	x86
PROCESSOR_ID...	x86 Family 15 Model 4 Stepping 3, Genu...

To add the path to the Python executable for only your user session, add the modification for **Path** to your User variables. Click on **New** under the User variables section to open a dialog box for a New User Variable.



Under **Variable name** type `Path` and under **Variable value** type `%PATH%;C:\Python26\`.

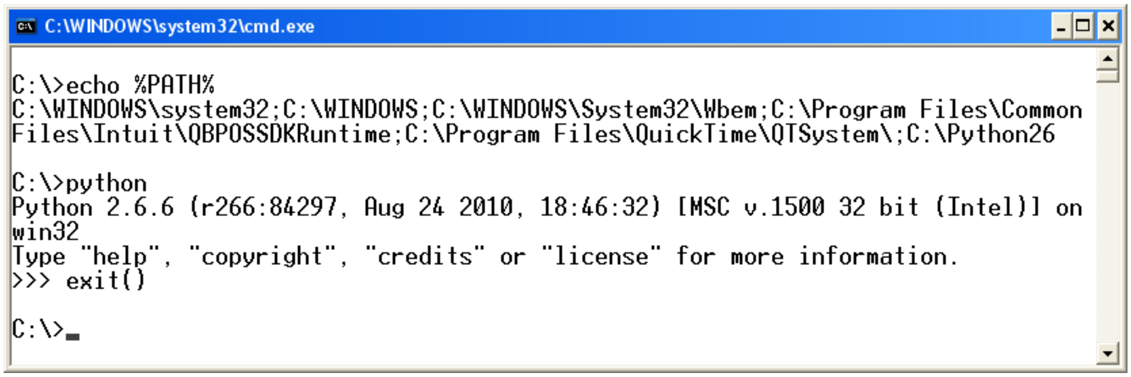


Variable name: Path

Variable value: %PATH%;C:\Python26\

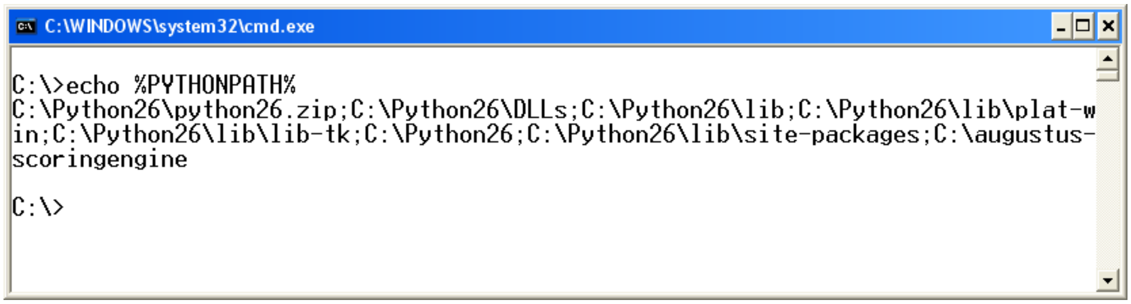
Click **OK**. You will have to accept all changes, close the System Properties window, close the **cmd** window, and then open a new **cmd** window for the changes to take effect in your session.

Once you open a new **cmd** window, you can check that C:\Python26 has been appended to the **Path** environment variable. If so, you can now type python at the command prompt instead of C:\Python26\python.exe to invoke an interactive session of Python. An example showing the new path and a successful invocation of the interactive session is below:



```
C:\>echo %PATH%
C:\WINDOWS\system32;C:\WINDOWS;C:\WINDOWS\System32\Wbem;C:\Program Files\Common
Files\Intuit\QBPOSSDKRuntime;C:\Program Files\QuickTime\QTSystem\;C:\Python26

C:\>python
Python 2.6.6 (r266:84297, Aug 24 2010, 18:46:32) [MSC v.1500 32 bit (Intel)] on
win32
Type "help", "copyright", "credits" or "license" for more information.
>>> exit()

C:\>_
```

You can also check that Augustus has been added to your **PythonPath** environment variable:



```
C:\>echo %PYTHONPATH%
C:\Python26\python26.zip;C:\Python26\DLLs;C:\Python26\lib;C:\Python26\lib\plat-w
in;C:\Python26\lib\lib-tk;C:\Python26;C:\Python26\lib\site-packages;C:\augustus-
scoringengine

C:\>
```

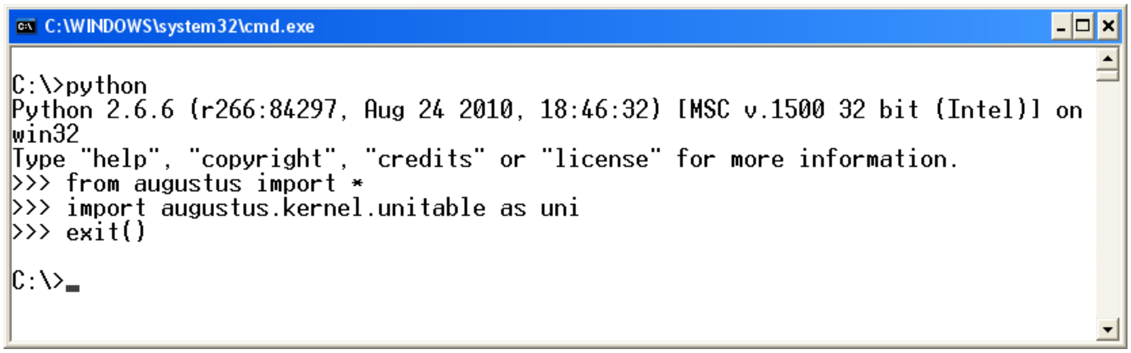
The paths look OK. Check that you can use modules from Augustus. At the **cmd** prompt, type:

```
C:\> python
```

to invoke Python. Then type:

```
>>> from augustus import *
>>> import augustus.kernel.unitable as uni
>>> exit()
```

If there are no error messages, then Python is able to find Augustus and its modules. The session should look like the following:

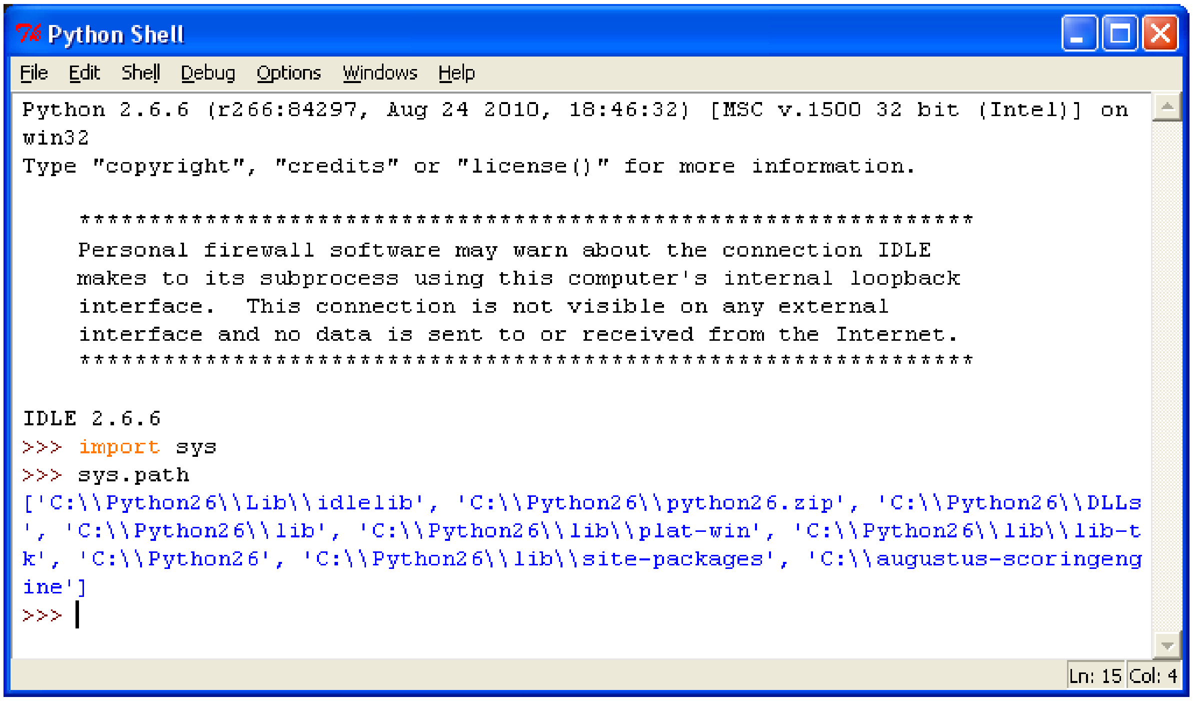


```
C:\>python
Python 2.6.6 (r266:84297, Aug 24 2010, 18:46:32) [MSC v.1500 32 bit (Intel)] on
win32
Type "help", "copyright", "credits" or "license" for more information.
>>> from augustus import *
>>> import augustus.kernel.unitable as uni
>>> exit()

C:\>_
```

Augustus is successfully installed.

Updates are visible in GUI environments

The environment update is also recognized in **IDLE**, the graphical user interface that ships with the Windows version of Python. A screenshot of the GUI is shown here:



```
Python 2.6.6 (r266:84297, Aug 24 2010, 18:46:32) [MSC v.1500 32 bit (Intel)] on
win32
Type "copyright", "credits" or "license()" for more information.

    ****************************************************************
    Personal firewall software may warn about the connection IDLE
    makes to its subprocess using this computer's internal loopback
    interface.  This connection is not visible on any external
    interface and no data is sent to or received from the Internet.
    ****************************************************************

IDLE 2.6.6
>>> import sys
>>> sys.path
['C:\\Python26\\Lib\\idlelib', 'C:\\Python26\\python26.zip', 'C:\\Python26\\DLLs
', 'C:\\Python26\\lib', 'C:\\Python26\\lib\\plat-win', 'C:\\Python26\\lib\\lib-t
k', 'C:\\Python26', 'C:\\Python26\\lib\\site-packages', 'C:\\augustus-scoringeng
ine']
>>> |
```

Note: For older versions of Augusuts to run on Windows, the `resources` module import and its use with metadata logging may have be commented out in `augustus/pmmllib/pmmlConsumer.py`, because the module was called but is not available on Windows. This impacted versions 264 to 346 of the file. Before that, the module was not used and since then, the platform is checked before performing the import.

3 Using Augustus

Augustus is now installed. See the Augustus Primer, included with the documentation and available from the project website, for a walk through of the examples.

If you want to remove Augustus from your system and you are running from the source, simply delete the directory:

```
$ rm -rf augustus-scoringengine
```

If Augustus was installed using `setup.py`, you can remove Augustus from the system by finding the directory and egg created by `setup.py` and removing them.

Index