IBM SPSS Statistics - Essentials for Python: Installation Instructions for UNIX Server

The following instructions are for installing IBM® SPSS® Statistics - Essentials for Python on UNIX Server operating systems.

Overview

IBM® SPSS® Statistics - Essentials for Python provides you with tools you need to start developing custom Python applications for use with IBM® SPSS® Statistics. It includes the following:

- The IBM® SPSS® Statistics - Integration Plug-in for Python for SPSS Statistics 21
- A set of custom procedures, implemented in Python, for use in SPSS Statistics

Install the IBM SPSS Statistics application

IBM® SPSS® Statistics - Essentials for Python is designed for the following applications:

- IBM SPSS Statistics Server for UNIX

There are no additional operating system and hardware requirements. The components installed with Essentials for Python work with any valid IBM® SPSS® Statistics license.

If you have not already done so, follow the instructions provided with the software to install one of the SPSS Statistics applications on the computer where you will install Essentials for Python.

Download and install Python 2.7

Version 21 of IBM® SPSS® Statistics - Essentials for Python requires Python version 2.7 (version 2.7.2 is recommended). Install Python on the computer where you will install Essentials for Python. The source code for Python is available from *http://www.python.org*. You must compile the source and then install Python. Note that although your system may already have Python 2.7 installed, it may not have been built with the configuration option required for the IBM® SPSS® Statistics - Integration Plug-in for Python. Therefore it is highly recommended to download the source for Python 2.7 and build Python yourself.

- Login with sufficient permissions to create and install files in the chosen locations.
- Create a temporary directory where you will uncompress and unpack the Python source. For example, at a command prompt type:

  ```
  mkdir ~/pysource
  ```

- Download the source from *http://www.python.org* and save it to the temporary directory.

- Change to the temporary directory. For example, at a command prompt type:

  ```
  cd ~/pysource
  ```

- Uncompress and unpack the Python source to the temporary directory. For example, at a command prompt type:

  ```
  tar xzf Python-2.7.2.tgz
  ```

- Change to the source directory. For example, at a command prompt type:

  ```
  cd Python-2.7.2
  ```

  Following are platform-specific instructions for building Python. We recommend that you follow these instructions, but as an alternative, you can follow the instructions provided by Python.org. If you decide to build and install Python using the instructions from Python.org, be sure to configure Python with the `--enable-shared` argument, as in: `./configure --enable-shared`. For Linux, it is recommended to also configure Python with the `--enable-unicode=ucs2` argument.

  Note: For Red Hat Enterprise Linux versions 5.x and 6.0, you must install Python 2.7 to a location other than the default location by including the `--prefix=<PYTHON_HOME>` option on `configure` and setting `<PYTHON_HOME>` to a location other than */usr* or */usr/local*.

  If you need to interrupt the build process while it is running, follow the instructions in the section To rerun the Python 2.7 build process before you restart the build.

  AIX

- To ensure that the appropriate compiler is on the path, execute the following command:

  ```
  export PATH=/opt/gnu/bin:$PATH:.:/usr/vacpp/bin
  ```

- In the file *Makefile.pre.in* (located at the root of the Python source directory) change "$(LDSHARED)" to "$(BLDSHARED)" on line 418. For example:

  ```
  sed '418s/$(LDSHARED)/$(BLDSHARED)/g' Makefile.pre.in >ftemp.pre.in
  rm -f Makefile.pre.in
  mv ftemp.pre.in Makefile.pre.in
  ```

- In the file *setup.py* (located at the root of the Python source directory) insert the following at the specified lines:

  insert 'os.environ["LDFLAGS"]="-brtl -L. -lpython2.6" ' at line 19

  insert ' if platform in ["aix3", "aix4", "aix5"]:' at line 1549

  insert ' return' at line 1550

For example:

```
sed '19i\
os.environ["LDFLAGS"]="-brtl -L. -lpython2.6" ' setup.py >ftemp1.py
sed '1549i\
      if platform in ["aix3", "aix4", "aix5"]:' ftemp1.py >ftemp2.py
sed '1550i\
         return' ftemp2.py > ftemp3.py
rm -f setup.py ftemp1.py ftemp2.py
mv ftemp3.py setup.py
```

- Use the following command to create the file *python.exp* under the *Modules* folder:

```
echo "python.exp" > ./Modules/python.exp
```

- Configure Python using the following command:

```
./configure –with-universal-archs="64-bit" –enable-universalsdk –with-gcc="xlc_r -q64" –with-cxx="xlc_r -q64"
–disable-ipv6 AR="ar -X64" CFLAGS="-qrtti=all -qopt=3 -qstrict -qlibansi -qthreaded -bexpall -brtl" LDFLAGS="-L. -bexpall -Wl,-brtl
-qrtti=all -bshared" CXX="xlc_r -q64" –enable-shared –prefix=<PYTHON_HOME>
```

  where <PYTHON_HOME> is the location where Python 2.7.2 is to be installed—typically, */usr/local*.

- Modify the resulting Makefile with the following changes:

```
Change "LDLIBRARY=      libpython$(VERSION).a" to "LDLIBRARY=      libpython$(VERSION).so" at line 156.
Change "$(BLDLIBRARY)" to "$(LIBRARY)" at line 405.
```

  For example:

```
sed '156s/LDLIBRARY=      libpython$(VERSION).a/LDLIBRARY=      libpython$(VERSION).so/g' Makefile >temp1
sed '405s/$(BLDLIBRARY)/$(LIBRARY)/g' temp1 > temp2
rm -f Makefile temp1
mv temp2 Makefile
```

- Run make and make install to build and install Python 2.7.2.

Solaris

- To specify necessary compiler settings and ensure the compiler is on the path, execute the following commands:

```
export CC="cc -xarch=v9"
export CXX="CC -xarch=v9"
export BASECFLAGS="-xarch=v9 -mt -xO2"
export CXXFLAGS="-xarch=v9 -mt -xO2"
export FC="g77 -m64 -O2"
export FFLAGS="-m64 -O2"
export LDFLAGS="-xarch=v9 -L/usr/local/lib/sparcv9 -L/usr/lib/sparcv9"
export LDSHARED="cc -G -xarch=v9"
export BLDSHARED="cc -G -xarch=v9"
export FSHARED="g77 -shared -m64"
export PATH=/opt/sfw/bin:/opt/sfw/sbin:/usr/bin:/usr/ccs/bin:/usr/sbin:/etc:
/usr/xpg4/bin:/usr/X/bin:/space/SunStudio/SUNWspro/bin:$PATH
export LD_LIBRARY_PATH=/space/SunStudio/SUNWspro/lib:/opt/sfw/lib:$LD_LIBRARY_PATH
```

- Configure Python using the following command:

```
./configure –enable-shared –prefix=<PYTHON_HOME>
```

  where <PYTHON_HOME> is the location where Python 2.7.2 is to be installed—typically, */usr/local*.

- Run make and make install to build and install Python 2.7.2.

Linux and zLinux

- If you will be using the Integration Plug-in for Python with the 64-bit version of IBM® SPSS® Statistics Server for UNIX, then execute the following commands to ensure the compiler is on the path and to configure Python:

```
export PATH=/usr/local340/bin:$PATH:.:/usr/local/bin
export LD_LIBRARY_PATH=/usr/local340/lib64:/usr/local/lib64:$LD_LIBRARY_PATH
./configure –enable-shared –enable-unicode=ucs2 –prefix=<PYTHON_HOME>
```

  or:

- If you will be using the Integration Plug-in for Python with the 32-bit version of SPSS Statistics Server for UNIX, then execute the following commands to ensure the compiler is on the path and to configure Python:

```
export PATH=/usr/local340/bin:$PATH:.:/usr/local/bin
export LD_LIBRARY_PATH=/usr/local340/lib:/usr/local/lib:$LD_LIBRARY_PATH
./configure  CC="gcc -m32" LDFLAGS="-m32" –enable-shared –enable-unicode=ucs2 –prefix=<PYTHON_HOME>
```

  where <PYTHON_HOME> is the location where Python 2.7.2 is to be installed—typically, */usr/local*.

- Run make and make install to build and install Python 2.7.2.

HP-UX

- To specify necessary compiler settings and ensure the compiler is on the path, execute the following commands:

```
export CC=cc
export CXX=aCC
export BASECFLAGS="+DD64"
export LDFLAGS="+DD64 -L. -lxnet"
export PATH=/usr/local/bin:$PATH:/opt/aCC/bin
export SHLIB_PATH=/usr/lib:/lib:$SHLIB_PATH
```

  where <PYTHON_HOME> is the location where Python 2.7.2 is to be installed—typically, */usr/local*.

- Configure Python using the following command:

```
./configure –without-gcc  –enable-shared –prefix=<PYTHON_HOME>
```

- Modify the resulting Makefile with the following changes:

  Change "+DD64 -L. -lxnet" to "-L. -L/usr/lib/hpux64 -lxnet" at line 66.

  Change "-DNDEBUG -O" to "-DNDEBUG" at line 59.

  For example:

```
sed '66s/+DD64 -L. -lxnet/-L. -L\/usr\/lib\/hpux64 -lxnet/g' Makefile > temp1
sed '59s/-DNDEBUG -O/-DNDEBUG/g' temp1 >temp2
rm -f Makefile
rm -f temp1
mv temp2 Makefile
```

- Run make and make install to build and install Python 2.7.2.

Configure the environment for Python and test Python

Python expects certain environment variables to be set. For convenience you may want to put these commands in a profile file so that they will be set upon login.

- Add the Python library location to the *LD_LIBRARY_PATH* environment variable (or whichever environment variable is appropriate for shared libraries on your operating system). In the following, *<PYTHON_HOME>* is the location where Python 2.7.2 is installed—typically, */usr/local*.

  For example, for AIX, at the UNIX prompt type:

```
export LIBPATH=<PYTHON_HOME>/lib:$LIBPATH
```

  Or, for Linux or Solaris, at the UNIX prompt type:

```
export LD_LIBRARY_PATH=<PYTHON_HOME>/lib:$LD_LIBRARY_PATH
```

Or, for HP-UX, at the UNIX prompt type:

export SHLIB_PATH=<PYTHON_HOME>/lib:$SHLIB_PATH

- Test Python to make sure that it can run. For example, at the UNIX prompt type:

  cd /usr/local/bin

  ./python2.7

  You should see a message from Python and the Python prompt (for example, >>>). You can end the Python session by pressing the Ctrl+D key combination and you can remove the Python source directory to recover disk space.

Install IBM SPSS Statistics - Essentials for Python

Note: It is not necessary to uninstall previous versions of IBM® SPSS® Statistics - Essentials for Python before installing a newer version.

For users who are working in distributed mode (with IBM® SPSS® Statistics Server) please install Essentials for Python on both the client and server machines.

The following steps document the silent installer using a response file. You can also execute the installer directly to launch a graphic version of the installer. You will need an X Window System to do so. If you are installing from the DVD/CD, the installer files for IBM® SPSS® Statistics - Essentials for Python are located under the *Programmability/Python Essentials* directory.

- Using a text editor, create a response file named *installer.properties*.
- Add the following properties and associated values to the response file:

```
INSTALLER_UI=silent
USER_SPSS_HOME=<IBM SPSS Statistics location>
USER_PYTHON_HOME=<Python 2.7 home directory>
LICENSE_ACCEPTED=true
```

where <IBM SPSS Statistics location> is the installation location of IBM SPSS Statistics and <Python 2.7 home directory> is the installation location of Python 2.7. For example:

```
USER_SPSS_HOME=/opt/IBM/SPSS/StatisticsServer21
USER_PYTHON_HOME=/opt/Python2.7
```

Note that `LICENSE_ACCEPTED=true` specifies acceptance of the license agreement.

- Save *installer.properties* to the directory containing the *.bin* file for Essentials for Python and change to that directory.
- Run the installer with the following command:

```
./<installer_name>
```

where <*installer_name*> is the name of the *.bin* file for Essentials for Python. *Note*: You must run the previous command as root, either by logging in as root or using the `sudo` command.

Note: To use a different response file (other than *installer.properties*), run the installer with the following command:

```
./<installer_name> -f <response file name>
```

Configure the environment for the IBM SPSS Statistics - Integration Plug-in for Python

The IBM® SPSS® Statistics - Integration Plug-in for Python requires additions to the *LD_LIBRARY_PATH* environment variable. For convenience you may want to put these settings in a profile file so that they will be set upon login. In the following, <*SPSS_HOME*> is the location where version 21 of the IBM® SPSS® Statistics application is installed.

- Add the SPSS Statistics library path location to the *LD_LIBRARY_PATH* environment variable (or whichever environment variable is appropriate for shared libraries on your operating system). For example, for Linux or Solaris, at the UNIX prompt type:

  export LD_LIBRARY_PATH=<SPSS_HOME>/lib:$LD_LIBRARY_PATH

- Optionally, for the AIX operating system, define the *LDR_CNTRL* environment variable. For example, at the UNIX prompt type:

  export LDR_CNTRL=MAXDATA=0x70000000

- Test the plug-in to make sure that it can run. For example, assuming that Python 2.7 is installed in */usr/local*, then at the UNIX prompt type:

  cd /usr/local/bin

  ./python2.7

  And from the Python prompt (for example, >>>) type:

  import spss

  You should not see any error messages. You can end the Python session by pressing the Ctrl+D key combination.

To rerun the Python 2.7 build process

If you have to stop the Python build process before it completes, you need to delete the files that were created by the process before attempting to rerun the process. For example, assuming that the Python 2.7 source is located in *~/pysource*, from the UNIX prompt type:

cd ~/pysource

make clean

Before you start using the IBM SPSS Statistics - Integration Plug-in for Python

After you install IBM® SPSS® Statistics - Essentials for Python, you will be able to start developing Python applications with the IBM® SPSS® Statistics - Integration Plug-in for Python. Documentation for the plug-in is provided in *Python Integration Package for IBM SPSS Statistics.pdf*, located in the *bin/lang/en/help/programmability* directory under the directory where IBM® SPSS® Statistics is installed. It describes how to use the plug-in and all of the functions available with the plug-in.

Download and install auxiliary Python modules for IBM SPSS Statistics

IBM SPSS has created a number of Python modules that build on, and in some cases extend, the functionality provided with the IBM® SPSS® Statistics - Integration Plug-in for Python. These modules are optional and require the Integration Plug-in for Python in order to function. The modules *spssdata*, *spssaux*, *namedtuple*, and *extension* are included with IBM® SPSS® Statistics - Essentials for Python. Additional modules are available for download from the SPSS community at *http://www.ibm.com/developerworks/spssdevcentral*.

You may want to read the article *How to Use Downloaded Python Modules*, also available from the SPSS community. In addition, you may need to download updated versions of auxiliary modules to take advantage of the new features available with IBM® SPSS® Statistics 21. Check the SPSS community for updates.

Running multiple versions of IBM SPSS Statistics and the IBM SPSS Statistics - Integration Plug-in for Python

You can have multiple versions of IBM® SPSS® Statistics Server for UNIX on a single machine along with a separate version of the IBM® SPSS® Statistics - Integration Plug-in for Python for each. For information on working with multiple versions, see the documentation for the plug-in.

Uninstalling IBM SPSS Statistics - Essentials for Python components

- Start a terminal program.
- Change the directory to *Uninstall_IBM_SPSS_Statistics_Essentials_for_Python_21* in the IBM® SPSS® Statistics installation directory.
- At the command prompt, type:

```
./Uninstall_IBM_SPSS_Statistics_Essentials_for_Python_21
```

Important: You must have permissions to remove the installation directory, or the uninstallation process will fail.

Custom Python procedures packaged with Essentials for Python

IBM® SPSS® Statistics - Essentials for Python includes a set of custom procedures, implemented in Python, that provide functionality beyond what is available with built-in SPSS Statistics procedures. All custom procedures, except for `FUZZY`, are available from the SPSS Statistics menus, once Essentials for Python is installed. All custom procedures (including `FUZZY`) are available from SPSS Statistics command syntax. The following table provides the menu location, the associated command name, and a brief summary for each of the custom procedures. Commands are run in the same manner as any built-in command such as `FREQUENCIES`.

Menu location	*Command name*	*Description*
none	FUZZY	Perform exact or fuzzy case-control matching.
File>Collect Variable Information	GATHERMD	Build a dataset of variable information from multiple datasets.
Data>Compare Datasets	SPSSINC COMPARE DATASETS	Compare two open datasets.
Transform>Create Dummy Variables	SPSSINC CREATE DUMMIES	Create a set of dummy variables representing the values of a variable.
Utilities>Merge Viewer Tables	SPSSINC MERGE TABLES	Merge the contents of one pivot table in the Viewer into another.
Utilities>Modify Output Titles	SPSSINC MODIFY OUTPUT	Modify output titles.
Utilities>Modify Table Appearance	SPSSINC MODIFY TABLES	Modify the appearance of pivot tables.
Transform>Programmability Transformation	SPSSINC TRANS	Apply a Python function to case data.
Analyze>Descriptive Statistics>TURF Analysis	SPSSINC TURF	Perform a TURF (Total Unduplicated Reach and Frequency) analysis.
Graphs>Regression Variable Plots	STATS REGRESS PLOT	Plots useful in assessing regression relationships.
Graphs>Compare Subgroups	STATS SUBGROUP PLOTS	Graphically compare the distributions of a set of variables across a partition of the data.

Notes

- Help for each of the procedures accessible from the menus is available from the Help button on the associated dialog box. The help is not, however, integrated with the SPSS Statistics Help system.
- Complete syntax help for each of the commands listed above is available by executing the command and including the `/HELP` subcommand—for example:

```
SPSSINC COMPARE DATASETS /HELP.
```

  The command syntax help is not, however, integrated with the SPSS Statistics Help system and is not included in the *Command Syntax Reference*.
- If the Analyze>Descriptive Statistics menu is not present in your IBM® SPSS® Statistics product, then please look on the Custom menu for the TURF analysis dialog.

- The dialogs were created with the Custom Dialog Builder in SPSS Statistics. You can view the design for any of the dialogs and/or customize them using the Custom Dialog Builder, available from Utilities>Custom Dialogs>Custom Dialog Builder. To view the design for a dialog, choose File>Open Installed from within the Custom Dialog Builder.
- The syntax commands are implemented as extension commands. The implementation code (Python modules) and XML specification files for these extension commands can be found in the *extensions* directory under the SPSS Statistics Server installation directory.

  If you have specified alternate locations for extension commands with the *SPSS_EXTENSIONS_PATH* environment variable then the files will be located in the first writable location in that variable instead of in the *extensions* directory.
- Other extension commands that are not included in Essentials for Python are available for download from the SPSS community. Newer versions of the Python procedures packaged with Essentials for Python may also be available there.