

# Getting Started with Python for Series 60 Platform

Version 1.2; September 28, 2005

# Python for Series 60 Platform

**NOKIA**

Copyright © 2005 Nokia Corporation. All rights reserved.

Nokia and Nokia Connecting People are registered trademarks of Nokia Corporation. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. Other product and company names mentioned herein may be trademarks or trade names of their respective owners.

#### **Disclaimer**

The information in this document is provided "as is," with no warranties whatsoever, including any warranty of merchantability, fitness for any particular purpose, or any warranty otherwise arising out of any proposal, specification, or sample. Furthermore, information provided in this document is preliminary, and may be changed substantially prior to final release. This document is provided for informational purposes only.

Nokia Corporation disclaims all liability, including liability for infringement of any proprietary rights, relating to implementation of information presented in this document. Nokia Corporation does not warrant or represent that such use will not infringe such rights.

Nokia Corporation retains the right to make changes to this specification at any time, without notice.

#### **License**

A license is hereby granted to download and print a copy of this specification for personal use only. No other license to any other intellectual property rights is granted herein.

## Contents

<b>1</b>	<b>Introduction.....</b>	<b>5</b>
1.1	Scope.....	5
1.2	Audience.....	5
1.3	Typographical Conventions.....	5
<b>2</b>	<b>Python for Series 60 Platform .....</b>	<b>6</b>
2.1	Delivery Package.....	6
2.2	Requirements .....	6
<b>3</b>	<b>Installing Python for Series 60.....</b>	<b>8</b>
3.1	Installing on a Series 60 Mobile Device.....	8
3.2	Installing into the Series 60 SDK .....	9
<b>4</b>	<b>Using Python for Series 60 .....</b>	<b>10</b>
4.1	Starting Python for Series 60.....	10
4.2	Running a Python Script.....	11
4.3	Running an Interactive Console.....	12
4.4	Writing Your Own Scripts.....	13
4.5	Installing a New Script.....	14
4.6	Using Python Installer Program .....	14
<b>5</b>	<b>Uninstalling Python for Series 60.....</b>	<b>16</b>
5.1	Uninstalling Python for Series 60 from the phone.....	16
5.2	Uninstalling Extensions and Scripts .....	16
5.3	Uninstalling Python for Series 60 from an SDK.....	16
<b>6</b>	<b>Terms and Abbreviations .....</b>	<b>17</b>
<b>7</b>	<b>References .....</b>	<b>18</b>

## Change History

December 10, 2004	Version 1.0	Initial document release
June 29, 2005	Version 1.1.5	Section 2.2 added. Sections 1, 3, 4.4, 5, and 6 updated.
September 28, 2005	Version 1.2	Sections 1.1, 3, 4.3, and 5.1 updated.

## 1 Introduction

Python is a simple, consistent and effective programming language. It is also open source, and it can be used for application development, prototyping, and teaching. The Python for Series 60 Platform (Python for Series 60) simplifies application development and provides a scripting solution for the Symbian C++ and Series 60 APIs. This release is based on Python 2.2.2.

The current version of the Python execution environment is simply a Python script that makes it possible to run other Python scripts to be found in the Python folder. The execution environment also enables the users to initiate an interactive Python console session, which is also implemented as a Python script. In addition, the Python execution environment creates a text console that is connected to the interpreter's standard text output streams. This setup is useful for Python for Series 60 environment evaluation and script development. Python Installer and interactive console support have been implemented entirely in Python. Note that to the end-users, the Python applications may look like any other applications.

**Tip:** The Python for Series 60 developer discussion board [5] on the Forum Nokia Web site is a useful resource for finding out information on specific topics concerning Python for Series 60. You are welcome to give feedback or ask questions about Python for Series 60 through this discussion board.

### 1.1 Scope

This document provides the information required to install Python on Series 60 SDKs and on Series 60 phones (models N-Gage QD, 6600, 6630, 6670, 7610, N70, and N90), as well as information on creating, installing, and using Python scripts and programs.

### 1.2 Audience

This guide is intended for developers looking to create programs that use the native features and resources of the Series 60 phones. Users of this guide should be familiar with basic programming and preferably also with the Python programming language.

### 1.3 Typographical Conventions

The following typographical conventions are used in this document:

<b>Bold</b>	Bold is used to indicate windows, views, pages and their elements, menu items, and button names.
<i>Italic</i>	Italics are used when referring to another chapter or section in the document and when referring to a manual. Italics are also used for key terms and emphasis.
<code>Courier</code>	Courier is used to indicate parameters, file names, processes, commands, directories, and source code text.
>	Arrows are used to separate menu items within a menu path.

## 2 Python for Series 60 Platform

The current version of Python for Series 60 is based on the Python version 2.2.2.

### 2.1 Delivery Package

Python for Series 60 as installed on a Series 60 device consists of:

- **Python** execution environment, which is visible in the main menu of the device and has been written in Python on top of Python for Series 60 (see *Series 60 SDK documentation [4]*)
- Python interpreter DLL
- Standard and proprietary Python library modules
- Series 60 UI application framework adaptation component (a DLL) that connects the scripting domain components to the Series 60 UI
- Python Installer program for installing Python files on the device, which consists of:
  - Recognizer plug-in
  - Symbian application written in Python

Python for Series 60 has the following documentation:

- This document contains information on how to install Python for Series 60 and how to write your first program
- *Python for Series 60 Platform API Reference [1]* contains the API and other reference material
- *Programming with Python for Series 60 Platform [2]* contains code examples and programming patterns for Series 60 devices that can be used as basis for programs

### 2.2 Requirements

This section describes what you need in addition to the Python for Series delivery package in order to perform different tasks. Table 1 sums up the additional requirements for different use cases.

Python for Series 60 uses the following development environment and software:

- Python, current release based on Python 2.2.2
- Symbian Series 60 1<sup>st</sup> or 2<sup>nd</sup> Edition SDK
- Python for Series 60 Platform
- A text editor, for example:  
IDLE (Integrated Development Environment), included in all Python versions from <http://www.python.org/>
- Connection between the mobile device and a PC (Infrared, Bluetooth wireless technology, USB, or a cellular connection with GSM or UMTS)

Table 1: Additional requirements

Task	What you need				
	A PC with a phone connection	A PC with a Bluetooth connection	A Windows PC	Series 60 SDK + emulator	C/C++ compiler for Symbian
Run Python scripts on the phone	X				
Write your own scripts for the phone	X				
Develop and debug Python applications for the phone		X			
Write your own scripts to run on an emulator (without a phone)			X	X	
Write your own Python extensions			X	X	X
Use Python from a C/C++ program			X	X	X

For more information on programming with Python, see *Programming with Python for Series 60 Platform [2]* and Python Web site at <http://www.python.org/>.

### 3 Installing Python for Series 60

This chapter presents the steps necessary for preparing the environment to run Python for Series 60 and the installation of the Python execution environment. To check which Python for Series 60 package to install, see Table 2.

**Note:** Table 2 lists the phone models that have been used for developing and testing Python for Series 60. Python for Series 60 also works on other Series 60 Second Edition and newer devices, but due to platform differences all modules may not function correctly.

Table 2: Series 60 Compatibility

Phone model	Python for Series 60 package
Nokia N70 Nokia N90	PythonForSeries60_for_2ndEd_FP3_SIS.zip
Nokia 6600 Nokia 6630 Nokia 6670 Nokia 7610	PythonForSeries60_for_2ndEd_SIS.zip
Nokia N-Gage QD	PythonForSeries60_for_1stEd_FP1_SIS.zip

#### 3.1 Installing on a Series 60 Mobile Device

There are two ways of installing the Python for Series 60 files on a Series 60 device:

- Nokia PC Suite software, see <http://www.nokia.com/> Web site for more information.
- Infrared or Bluetooth wireless technology

If you use Nokia PC Suite and have associated files of SIS type with the PC Suite application, click the `PythonForSeries60_for_1stEd_FP1_SIS.zip`, `PythonForSeries60_for_2ndEd_SIS.zip`, or `PythonForSeries60_for_2ndEd_FP3_SIS.zip` file (see Table 2). The execution environment is installed automatically, but you may have to confirm the file transfer from the phone.

You can also install Python for Series 60 directly from a PC using the Infrared port or Bluetooth wireless technology.

**To install Python for Series 60 with Infrared or Bluetooth wireless technology:**

1. Close all applications in your device before you start the installation.
2. Establish an Infrared or a Bluetooth connection between your PC and your phone.

For more information on Infrared or Bluetooth connections, see the appropriate Infrared or Bluetooth wireless technology documentation in your phone manual.

3. Right-click `PythonForSeries60_for_1stEd_FP1_SIS.zip`, `PythonForSeries60_for_2ndEd_SIS.zip`, or `PythonForSeries60_for_2ndEd_FP3_SIS.zip` according to the model of your phone, and send the file to your phone.



4. Select whether Python for Series 60 should be installed on the Phone memory or Memory card. A progress bar indicates the status of the installation.

The **Installation complete** pop-up note indicates that the package has been installed on the phone and that the **Python** icon is available on the main menu.

5. After installation, an application called **Python**, which launches the Python execution environment, is visible in the phone's (or an emulator's) main menu.

Select the **Python** icon to start Python. Select **Options > Run script** to test Python on your phone. Follow the instructions in Chapter 4, *Using Python for Series 60*.

### 3.2 Installing into the Series 60 SDK

When you download the SDK from the Internet, note that you must choose a Nokia, Borland, or Microsoft edition of Series 60 SDK for Symbian OS. The current release does not support CodeWarrior editions.

You may install the Python emulator and sample scripts to all the supported SDK versions separately. Choose the installer from the following list:

- PythonForSeries60\_1\_2\_for\_1stEd\_FP1\_SDK.zip- installer for SDK version 1.2
- PythonForSeries60\_1\_2\_for\_2ndEd\_SDK.zip- installer for SDK version 2.0
- PythonForSeries60\_1\_2\_for\_2ndEd\_FP1\_SDK.zip - installer for SDK version 2.1
- PythonForSeries60\_1\_2\_for\_2ndEd\_FP2\_SDK.zip- installer for SDK 2<sup>nd</sup> Edition, Feature Pack 2
- PythonForSeries60\_1\_2\_for\_2ndEd\_FP3\_SDK.zip - installer for SDK 2<sup>nd</sup> Edition, Feature Pack 3

Start the installer program and follow the instructions. The installer asks for a folder to install the customer documentation and utilities, which are equal to all SDK versions.

The installer copies the emulator files and some sample scripts into the directory

```
<EPOCROOT>\Epoc32\Release\wins\udeb\z\system\apps\PYTHON
```

where <EPOCROOT> is the root folder of your SDK installation.

**Note:** The `run_script` command expects the script to be in the directory `<EPOCROOT>\Epoc32\Release\wins\udeb\z\system\apps\PYTHON` or one of its subdirectories, such as `my`.

You can copy the sample scripts by a different name and then edit the copies. Note that you do not have to restart the Python execution environment if you modify the scripts.

Using the emulator is like using Python for Series 60 on the phone. For more information, see Chapter 4, *Using Python for Series 60*.

For more information and for guidelines on how to extend Python for Series 60, see *Programming with Python for Series 60 Platform [2]*.

## 4 Using Python for Series 60

This chapter provides information on how to use Python for series 60 on a phone or in an emulator.

### 4.1 Starting Python for Series 60

You can use Python for Series 60 from the phone UI in two different ways: either by running the console, or by launching a script. Both options are available in the Python execution environment. It is also possible to install individual scripts to the terminal in a way that makes them appear in the main menu as native applications. In such a case, the Python interpreter is not visible to the users.

Using Python in the emulator is mostly similar to using it in a phone. The instructions given in this chapter apply for both phone and emulator, except for a few details that are specifically pointed out.

**Tip:** If you use Python regularly, you may want to move the **Python** icon up towards the top of the screen for easier access.

**To start Python for Series 60 on the phone:**

1. Select the **Python** icon on your phone's main menu (see Figure 1).  
Python is started and a copyright text is displayed.



Figure 1: Launching Python

2. To run the interactive Python interpreter console, select **Options > Interactive console**.
3. To launch a Python script from a menu, select **Options > Run script**.
4. To find out more about Python, select **Options > About Python**.

**To start Python for Series 60 on the SDK:**

1. Launch the debug version of the emulator.
2. Select the **Python** icon on the emulator's main menu.

## 4.2 Running a Python Script

To get a list of Python scripts, choose **Options > Run Script** (see Figure 2). This dialog displays all the scripts currently installed in the environment.



Figure 2: Script selection

`default.py` and `interactive_console.py` are the scripts that implement the Python application you are currently interacting with. For this reason, they are fundamental and should not be removed.

`filebrowser.py` is a simple file browser for the device that allows you to browse directories and open files that have a handler associated with them in the system. Note that you can press the navigation key to open or delete a file.

To run a script, select it (see Figure 3) and press **OK**.



Figure 3: Script selection

The `filebrowser.py` execution is illustrated in Figure 4.

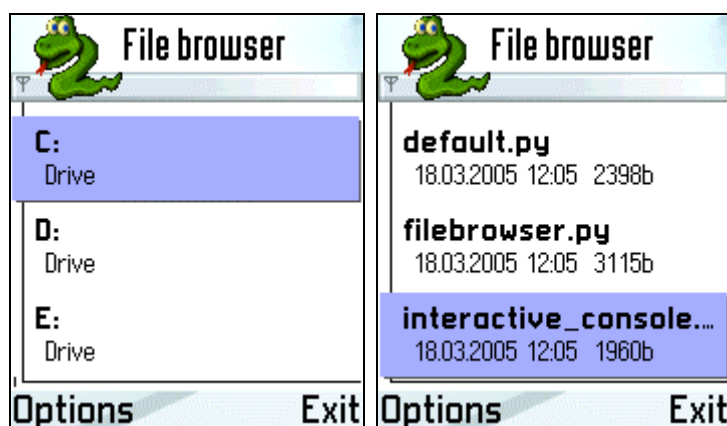


Figure 4: File browser script

To end the script, press the selection key **Exit**.

It is up to the individual script to decide how it behaves on **Exit**; it may choose to let the entire Python execution environment exit. If the execution environment exits, the main menu opens in the device display.

### 4.3 Running an Interactive Console

The interactive console allows you to enter and execute Python commands.

**Note:** The current Python version for Series 60 does not include all the standard Python library modules. Some extension modules that work in the phone are not implemented in the SDK version. For more information on standard library modules, see *Programming with Python for Series 60 Platform [2]*.

Note that in some Series 60 phone models the dictionary is on by default. This may cause difficulties while using the interactive console. It is advisable to turn the dictionary off and also to specify lowercase characters only as the case is significant in Python.

The Python interpreter prompts with "`>>>`". You can type a basic command, such as

```
>>> print "hello world"
```

and see the output on the screen (see Figure 5). You can use the **Navigation** key to produce `<enter>`. It can also be produced by pressing the **[0]** ('zero') key three times.

When using an emulator, note that you cannot type quotation marks using a keyboard; they have to be selected using the emulator's own keys instead. However, you can press **Enter** in the keyboard. Note also that sometimes the last line of text is not fully visible in the emulator's display.



Figure 5: Interactive console

To end the interactive console, press the **Exit** key. You can also select **Options > Previous command** to get any of the previous commands from the buffer for editing. Select **Previous** repeatedly until you get the appropriate command.

#### 4.4 Writing Your Own Scripts

The following example uses a Python for Series 60 specific extension module called `appuifw` and shows how to write your own Python scripts. For information on the `appuifw` module, see *Python for Series 60 Platform API Reference [1]*.

To write your own scripts:

1. Open a text editor.
2. Write the following lines (the first line is a comment):

```
# Hello
from appuifw import *
note(u"Hello World!", "info")
```

3. Save the file as `hello.py`. (In an emulator, save it to the script directory `<EPOCROOT>\Epo32\Release\wins\udeb\z\system\apps\PYTHON` and skip the following steps.)
4. Go to the location where you saved the file and right-click it.
5. Send the file to your phone. A message appears and asks you to save the file.

If sending the file fails, check that you do not have a file with the same name already in the Inbox. If this is the case, delete the earlier message.

6. Save the file as a Python script.
7. Now start Python in the main menu to test your script.

## 4.5 Installing a New Script

There are several methods of installing a new script:

- Send the new script to the device's Inbox using any of the available bearers.
- Make a SIS file and install it to the phone as any other SIS file using any of the available bearers. For information on making a SIS file for a standalone Python application, see *Programming with Python for Series 60 Platform [2]*.
- Copy the script into the device by using Nokia PC Suite software. Since the System directory in the phone is not visible to the phone browser, you need to create a new directory on the phone where you can send the script files. You can then move the files to the right location with a suitable application.

Note that the `run_script` command expects the script to be in the directory `<drivename>:\System\Apps\Python` – where `<drivename>` is the drive where the Python for Series 60 package has been installed – or one of its subdirectories such as `my`.

The `default.py` script looks in this directory for Python scripts to run.

**Note:** The `default.py` script defines where the execution environment looks for scripts and modules in the new directory/directories. For more information on `default.py`, see *Programming with Python for Series 60 Platform [2]*.

To install a Python extension (for example, a standard library module):

1. Establish a connection between your PC and your phone.
2. Right-click the extension you want to install and send it to your phone.  
For more information, see the appropriate PC or phone documentation.
3. Allow the Python Installer to install the extension as a Python library module.

Many standard Python modules and extensions work unmodified with Python for Series 60; for a list of these, see *Python for Series 60 Platform API Reference [1]*. If you have Python 2.2 installed on a PC, you can install the standard modules to the phone by copying them from the `libs` directory. You can also experiment with unlisted modules. Some standard modules (for example, Unicode codecs) were left out in order not to waste storage space in the device.

For more information on extensions, see Table 3. For writing a Python script and writing your own Python extensions, see *Programming with Python for Series 60 Platform [2]*.

## 4.6 Using Python Installer Program

The Python Installer program makes it possible for the users to install Python scripts and library modules to the phone by sending them to the phone as such. Python Installer is installed simultaneously with Python for Series 60, but it is not necessary when using an SDK.

The Python Installer comes with a MIME (Multipurpose Internet Mail Extensions) type recognizer. It recognizes files with filename extensions `.py`, `.pyc`, `.pyo`, and `.pyd`.

The Python Installer asks on installation whether the file is to be installed as a "Python script" (under Python execution environment) or as a "Python lib module" (library component). Files with the extension `.pyc`, `.pyo`, or `.pyd` can only be installed as a library (see Table 3).

Table 3: Installation options

File extension	If installed as	Will be installed in	And appear
.PY	Python script	\System\Apps\Python\my\	As an option in <b>Options &gt; Run script</b>
.PY, .PYO, .PYC	Library module	\System\libs	By importing to an application
.PYd	Library module	\System\libs	By importing to an application

Python for Series 60 can use .PYC (pre-compiled) and .PYO (optimized) files created on other platforms and moved to the phone. Using precompiled files can improve performance but the gain is usually minimal. A .PYO file leaves out documentation and comments, so it can save space. The phone itself does not save compiled or optimized files.

## 5 Uninstalling Python for Series 60

This chapter provides instructions for uninstalling Python for Series 60 from your phone and PC.

### 5.1 Uninstalling Python for Series 60 from the phone

**Warning:** Uninstalling Python for Series 60 will delete all scripts in the subfolder `my`.

To uninstall Python for Series 60:

1. Open the application manager of your Nokia device.
2. Select **Python for Series 60**. (Note: There is also a program called **Python for Symbian OS**. If you select it by mistake, you still have to uninstall the **Python for Series 60** program.)  
Do not start the program.
3. Select **Options > Remove**.  
The application manager asks if you want to remove the Python application.
4. Select **YES**.  
Python for Series 60 has now been removed from your phone.
5. You may have to use the application manager of your Nokia device to uninstall any additional Python libraries.

### 5.2 Uninstalling Extensions and Scripts

To remove `.py`, `.pyc`, `.pyd`, or `.pyo` files from your phone, open the file manager of your Nokia device. Select the file you want to remove and select **Options > Remove**.

These files are located in the directories `system/apps/python/my` or `system/libs`.

**Note:** If you need to remove the subfolder `my`, you can use the file manager of your Nokia device to remove the subfolder from the directory `system/apps/python`.

### 5.3 Uninstalling Python for Series 60 from an SDK

To remove Python for Series 60 from your PC, use the **Add/Remove Programs** feature in **Control Panel**.



## 6 Terms and Abbreviations

The following list defines the terms and abbreviations used in this document:

API	Application Programming Interface
Bluetooth	Bluetooth is a technology for wireless communication between devices and is based on a low-cost short-range radio link.
Emulator	Emulator is an application that emulates the functionality of the external application interface of brokers and servers.
IDLE; Integrated DeveLopment Environment	IDLE is a Python development environment that gives the users access to the Python interactive mode in the computer.
Infrared	Refers to wireless technology where data can be transferred between devices and systems through Infrared radiation.
MIME; Multipurpose Internet Mail Extensions	MIME is an extension of the original Internet e-mail protocol that can be used to exchange different kinds of data files on the Internet.
SDK; Software Development Kit	SDK is a set of programming tools for creating applications and enhancing the use of certain software.
SIS file	Symbian Installation System (SIS) file is a compressed file, containing an installation package. A SIS file consists of all the required files for a software component installation to a device based on the Symbian operating system.
UID; Unique Identification	A UID, or a "Unique Identifier", is a globally unique 32-bit number used in a compound identifier to uniquely identify an object, file type, etc. Each released Symbian platform application needs to have at least one Unique Identification (UID) number assigned to it.

## 7 References

1. Python for Series 60 Platform API Reference  
<http://www.forum.nokia.com/>
2. Programming with Python for Series 60 Platform  
<http://www.forum.nokia.com/>
3. Python Web site  
<http://www.python.org/>
4. Series 60 SDK documentation
5. Python for Series 60 developer discussion board  
<http://discussion.forum.nokia.com/>