

# Installing PyLucene

## Table of contents

1 Building PyLucene.....	2
1.1 Requirements.....	2
1.2 For the Impatient Ones.....	2
1.3 For the Rest of Us.....	2
2 Notes for Solaris.....	3

## 1. Building PyLucene

PyLucene is completely code-generated by JCC whose sources are included with the PyLucene sources.

### 1.1. Requirements

To build PyLucene a Java Development Kit (JDK) and [Ant](#) are required; use of the resulting PyLucene binaries requires only a Java Runtime Environment (JRE).

The [setuptools](#) package is required to build and run PyLucene on Python 2.3.5. With later versions of Python, setuptools is only required for shared mode. See JCC's [installation instructions](#) for more information.

### 1.2. For the Impatient Ones

1. `pushd jcc`
2. `<edit setup.py to match your environment>`
3. `python setup.py build`
4. `sudo python setup.py install`
5. `popd`
6. `<edit Makefile to match your environment>`
7. `make`
8. `sudo make install`
9. `make test` (look for failures)

### 1.3. For the Rest of Us

Before building PyLucene, JCC must be built first. See JCC's [installation instructions](#) for building and installing it.

Once JCC is built and installed, PyLucene is built via `make` which invokes JCC. See PyLucene's `Makefile` for configuration instructions.

There are limits to both how many files can fit on the command line and how large a C++ file the C++ compiler can handle. By default, JCC generates one large C++ file containing the source code for all wrapper classes.

Using the `--files` command line argument, this behaviour can be tuned to workaround various limits, for example:

- to break up the large wrapper class file into about 2 files:  
`--files 2`

- to break up the large wrapper class file into about 10 files:  
    `--files 10`
- to generate one C++ file per Java class wrapped:  
    `--files separate`

## **2. Notes for Solaris**

PyLucene's Makefile is a GNU Makefile. Be sure to use `gmake` instead of plain `make`.

Just as when building JCC, Python's `distutils` must be nudged a bit to invoke the correct compiler. Sun Studio's C compiler is called `cc` while its C++ compiler is called `CC`.

To build PyLucene, use the following shell command to ensure that the C++ compiler is used:

```
$ CC=CC gmake
```