

# Incorporating Python into LSB

## Identification of Issues

# Python in LSB: Goals

- Allow Python scripts to run unmodified across Linux distributions
  - provided the scripts restrict themselves to supported API
- Optional: Allow extension modules to be used in binary form across distribution
  - requires specification of an C/Python ABI
- Optional: Allow installation of binary foreign Python add-on packages
  - requires specification of file system layout
  - does not necessarily require ABI specification (could still work for pure-Python libraries)

# Portability of Scripts

- Issue: How to specify language semantics and available API?
  - Option 1: don't specify it at all; make it explicitly unspecified what Python version is installed
  - Option 2: explicitly specify all packages, modules, classes, functions, methods, and symbolic values, as well as the language
  - Option 3: include Python language reference and library reference as a normative reference (for a specified version of these)
  - Option 4: like 3, but exclude some of the standard modules, and include some non-core modules
- Issue: How to deal with alternative Python implementations?
  - Jython, IronPython, PyPy
  - Proposal: Explicitly mandate CPython

## Portability of Scripts (2)

- Issue: Assuming CPython 2.x, for some x (=4 or =5), what “standard” “extension” modules should be available?
  - standard: Included in the core Python release, and documented in the library reference
  - extension: implemented in C, wrapping some other library
  - Proposal: all modules that only require libraries included in LSB
  - builtin: `_ast`, `codecs`, `_sre`, `_symtable`, `_types`, `errno`, `exceptions`, `gc`, `imp`, `marshal`, `posix`, `pwd`, `signal`, `sys`, `thread`, `zipimport`
  - included: `array`, `audioop`, `binascii`, `_bisect`, `cmath`, `_codecs_{cn,hk,iso2022, jp,kr,tw}`, `collections`, `cPickle`, `crypt`, `cStringIO`, `_csv`, `_ctypes`, `_curses`, `datetime`, `dl`, `_elementtree`, `fcntl`, `_functools`, `grp`, `_hashlib`, `_heapq`, `_hotshot`, `imageop`, `itertools`, `linuxaudiodev`, `_locale`, `_lsprof`, `mmap`, `_multibytecodec`, `operator`, `ossaudiodev`, `parser`, `pyexpat`, `_random`, `resource`, `rbgimg`, `select`, `socket`, `spwd`, `strop`, `_struct`, `syslog`, `termios`, `time`, `unicodedata`, `_weakref`, `zlib`
  - excluded: `_bsddb`, `bz2`, `_curses_panel`, `dbm`, `nis`, `_ssl`, `_tkinter`,
  - questionable: `_ctypes_test`, `_testcapi`, `xxsubtypes`

# Portability of Scripts (3)

- Issue: Assuming CPython 2.x, what pure-Python modules should be included?
  - Proposal: Everything “make install” installs
    - Explicit list of modules should be specified (referencing the standard library documentation)
  - in particular including distutils (even though cc(1) is not in LSB)
- Issue: What should be the name of the Python interpreter?
  - Specification needed for #!-support
  - Option 1: /usr/bin/python
    - pro: script may also run on some non-LSB systems
    - con: requires to hard-code version of /usr/bin/python
  - Option 2: /usr/bin/python2x (x=4 or x=5)
  - Option 3: something else
- Issue: What non-core libraries should be required?
  - No proposal here

# Portability of Extension Modules

- Issue: How should the ABI be specified?
  - Python guarantees ABI stability between 2.x.y1 and 2.x.y2, but not between 2.x1 and 2.x2
  - Proposal: Mandate support for a specified ABI (e.g. 2.5)
- Issue: What is `sizeof(Py_UNICODE)`?
  - Proposal: 4
- Issue: `--enable-shared` (i.e. `libpython2.x.so`)?
  - Proposal: no

# Installation of Foreign Packages

- Issue: What aspects of file hierarchy are guaranteed?
  - Potential FHS matter?
  - Proposal: `--prefix=/usr`
  - Proposal: specify layout of “make install” installation
    - `/usr/bin/python2.5`, `/usr/lib/python2.5`