Samba 4 Python Scripting

Jelmer Vernooij

Samba Team

April 21, 2008

Why Python?

Changes in the last year

Some trivial examples

Demo

Creating bindings

Future

Agenda

- Why Python?
- Changes in the last year
- Some trivial examples
- Demo
- Creating bindings
- Future

Why Python?

Changes in the last year

Some trivial examples

Demo

Creating bindings

Future

Why scripting?

- Quicker development
- Easier to understand
  - Easy to use for system administrators
  - Lower barrier for contributions?

History of Samba and scripting

- Samba 3.0 had python bindings
  - Removed because of lack of maintainance
  - Not used for any core infrastructure
- Samba 4 has embedded JavaScript
  - Originally used for SWAT, provisioning

Why Python?

Changes in the last year

Some trivial examples

Demo

Creating bindings

Future

Why Python?

- Replaces EJS as internal scripting language
- Python is now a build-dependency
  - Easy to build from source, no dependencies
  - Ported to at least as much platforms as Samba
  - Available as standard package for most platforms
- The longer we would've waited, the more code we'd have to port

Why Python?

- Comes “with batteries included”
  - No need to reimplement utility functions and bindings for Samba
- Easy to create bindings
- Most existing libraries already have Python bindings
  - GTK+, Qt, HTTP, .ini-parsers...
- Large existing developer base
  - Potential contributors
- Better scripting language
  - Nested functions
  - Modularity
- More development tools available
  - Debugger, profiler, code coverage analyser, ...

What exactly is Python?

- High-level general purpose scripting language
- Object-oriented, structured or functional programming
- Memory-managed, reference-counted
- Standardsized, several implementations
- Portable
- Created in ’91

What does it look like?

Hello world!

```
print "Hello World!"
```

Function usage

```
def addone(value):
    return value + 1
```

Using modules

```
import unittest
```

Some interesting software in Python

Written in Python:

- Original BitTorrent client
- Mailman
- Bazaar/Mercurial
- Trac
- Some apps in GNOME/KDE

Scriptable in Python

- Blender
- Amarok, Rhythmbox
- Vim
- Totem
- Epiphany

Why Python?

Changes in the last year

Some trivial examples

Demo

Creating bindings

Future

Python in Samba 4!

- pidl can now generate Python bindings
- SWIG used for binding several other libraries
- some bindings written manually
- now used instead of EJS in a lot of places
  - EJS still there but may be removed at a later point

Existing Samba Python bindings

- Credentials
- SamDB
- Most DCE/RPC modules
- LDB
- TDB
- Registry
- Libnet

- Includes matching unit tests, so all bindings should work
- Bindings should be Pythonic rather than one-on-one wrappers of C functions

Infrastructure in Python

- Provisioning
- smbclient?
  - SoC student will work on this
  - Proof-of-concept will be interesting, may not be ideal as actual implementation
- SWAT
  - SoC student will work on this hopefully
- Samba-GTK
  - SoC student will work on this

Reading TDB files

```
1  import tdb, sys
2
3  db = tdb.Tdb(sys.argv[1])
4  for (k, v) in db.items():
5      print "{"
6      print "key(%d) = %r" % (len(k), k)
7      print "data(%d) = %r" %  (len(v), v)
8      print "}"
```

Using LDB

```
1  #!/usr/bin/python
2
3  import ldb
4
5  conn = ldb.Ldb("msg.tdb")
6
7  conn.add({"dn": "dc=samba,dc=org", "attr1": "foo"})
8
9  for msg in conn.search("dc=samba,dc=org"):
0      print str(msg.dn)
```

Connecting to LDAP using LDB

```
1  #!/usr/bin/python
2
3  import ldb
4
5  # Connect to the LDAP server
6  conn = ldb.Ldb("ldap://ldap.abmas.org/")
7
8  for msg in conn.search("dc=samba,dc=org"):
9      print str(msg.dn)
```

Adding users

```
1  #!/usr/bin/python
2  import samr, lsa
3
4  # Connect to the local SAM
5  conn = samr.samr("ncalrpc:", "st/dc/etc/smb.conf")
6
7  # Get SAMR connect handle
8  samr_handle = conn.Connect(0, 0xffffffff)
9
0  domainname = lsa.String()
1  domainname.string = u"SAMBADOMAIN"
2
3  sid = conn.LookupDomain(samr_handle, domainname)
4  print "Found sid %s for SAMBADOMAIN" % sid
5
6  conn.Close(samr_handle)
```


Why Python?

Changes in the last year

Some trivial examples

Demo

Creating bindings

Future

Unit tests

```
1  import winreg
2  from samba.tests import RpcInterfaceTestCase
3
4  class WinregTests(RpcInterfaceTestCase):
5      def setUp(self):
6          self.conn = winreg.winreg("ncalrpc:", self.get_load
7
8      def test_hklm(self):
9          handle = self.conn.OpenHKLM(None,
0                      winreg.KEY_QUERY_VALUE | winreg.KEY_ENUME
1          self.conn.CloseKey(handle)
```


Why Python?

Changes in the last year

Some trivial examples

Demo

Creating bindings

Future

Demo

Demo Time

Why Python?

Changes in the last year

Some trivial examples

Demo

Creating bindings

Future

Creating bindings

- We use SWIG, `http://www.swig.org/`.
- Hard to grasp language but very powerful

Missing bindings

- NetBIOS
- SMB
- WINS
- CLDAP
- LDAP

Where to get?

- ldb
  - `http://ldb.samba.org/`
  - `python-ldb` in Debian/Ubuntu
- tdb
  - `http://tdb.samba.org/`
  - `python-tdb` in Debian/Ubuntu
- . . . others:
  - Samba 4
  - . . . not packaged yet, but hopefully soon

Learning more

- `www.python.org`
- *pydoc* < *name* >
- pydoctor
- Maybe public API docs on Samba.org?

Why Python?

Changes in the last year

Some trivial examples

Demo

Creating bindings

Future

Future expansions

- Port to Samba 3?
  - Need to make sure it stays maintained
- Help welcome :-)