



I18n Testing with Dogtail

Harry Fu(付宁)

Asian Globalization Center

Sun China



AGENDA

- 1 – I18n(L10n) Testing
- 2 – Introduction of Dogtail
- 3 – Dogtail's I18n Module
- 4 – I18n Testing with Dogtail
- 5 – Demo
- 6 – Q&A

I18n/L10n Testing -- Scope

- Stability & Function Integrity
- Double-Byte Character Processing(Print)
- Date/Time Format
- GUI Display(font, size, etc...)
- Input Method
-

I18n/L10n Testing – Challenges

- Large language/locale matrix
 - > 30+ languages
 - > 80+ locales
- Human Verification is needed
- GNOME is evolving

I18n/L10n Testing – Automation

- Record & Replay
 - > Key Strokes
 - > Different Hot Keys
- Image Capturing & Comparing
 - > Golden Images
 - > Compare Differences

Dogtail – What's it?

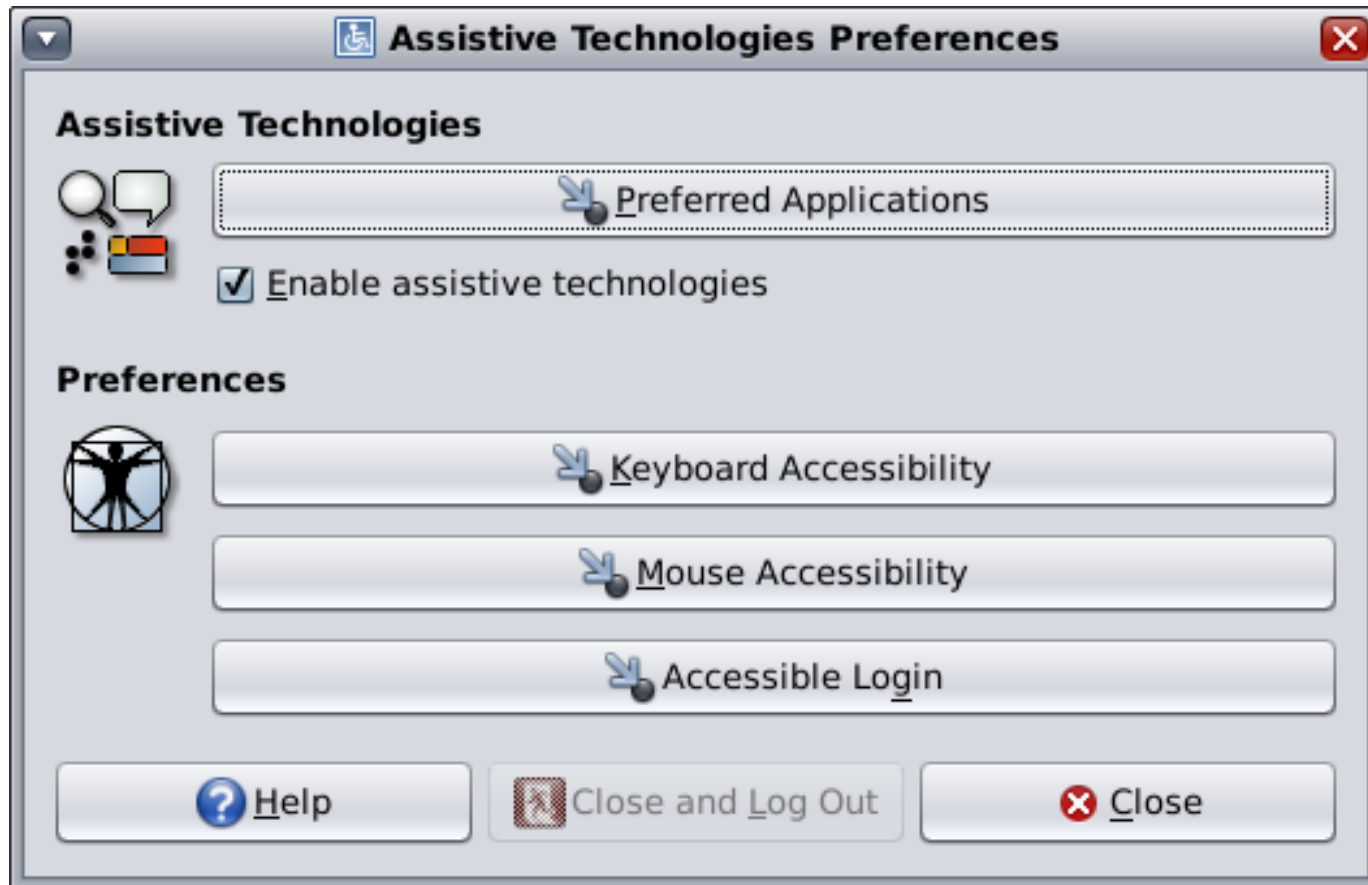
- A GUI test tool and automation framework written in Python.
- Supports GNOME applications and many other GTK+ applications.
- It uses Accessibility (A11Y) technologies to communicate with desktop applications.
- Dogtail scripts are written in Python and executed like any other Python program.
- Dogtail is an open source project.

Dogtail – How does it work

- **Step 1 - Install Dogtail**
 - > AT-SPI-enabled desktop(GNOME)
 - > Python 2.3 or higher
 - > ImageMagick 6.2 or higher
 - > pyspi - Python AT-SPI bindings
 - > rpm-python or python-apt
 - > ElementTree for Python

Dogtail – How does it work

- Step 2 - Set up accessibility



Dogtail – How does it work

- **Step 3 - Identify the GUI elements**
 - > Dogtail-recorder
 - > Sniff or Accerciser
 - > `root.application().dump`

Dogtail – How does it work

```
root@opensolaris:~# cat dump.py
#!/usr/bin/python
from dogtail.tree import root
root.application('gedit').dump()
root@opensolaris:~# ./dump |more
bash: ./dump: No such file or directory
root@opensolaris:~# ./dump.py |more
Creating logfile at /tmp/dogtail/logs/dump_20081017-151111_debug ...
Solaris distribution
{"gedit" application}
Node roleName='frame' name='未保存文档 2 - gedit' description=''
Node roleName='filler' name='' description=''
Node roleName='menu bar' name='' description=''
Node roleName='menu' name='文件(F)' description='' text='文件(F)'
click
Node roleName='tear off menu item' name='' description='' text=''
click
Node roleName='menu item' name='新建(N)' description='' text='新建(N)'
click
Node roleName='separator' name='' description='' text=''
click
Node roleName='separator' name='' description='' text=''
click
Node roleName='menu item' name='打开(O)...' description='' text='打开(O)...
```

Dogtail – How does it work

- **Two Types of APIs**
 - > Procedural API(dogtail.procedural)
 - Require less Python programming experience
 - Limited control of GUI objects
 - Simple but still powerful enough for general desktop GUI automated testing
 - > Object-oriented API(dogtail.tree)
 - Require much Python programming
 - Fine-grain control of GUI objects
 - Easy customization through subclassing

Dogtail – How does it work

- **I18n Module**

- > Makes use of 'gettext'(po-->mo)

- > LoadTranslationsFromPackageMoFiles

- import dogtail.i18n

- dogtail.i18n.loadTranslationsFromPackageMoFiles('gedit')

- getMoFilesForPackage

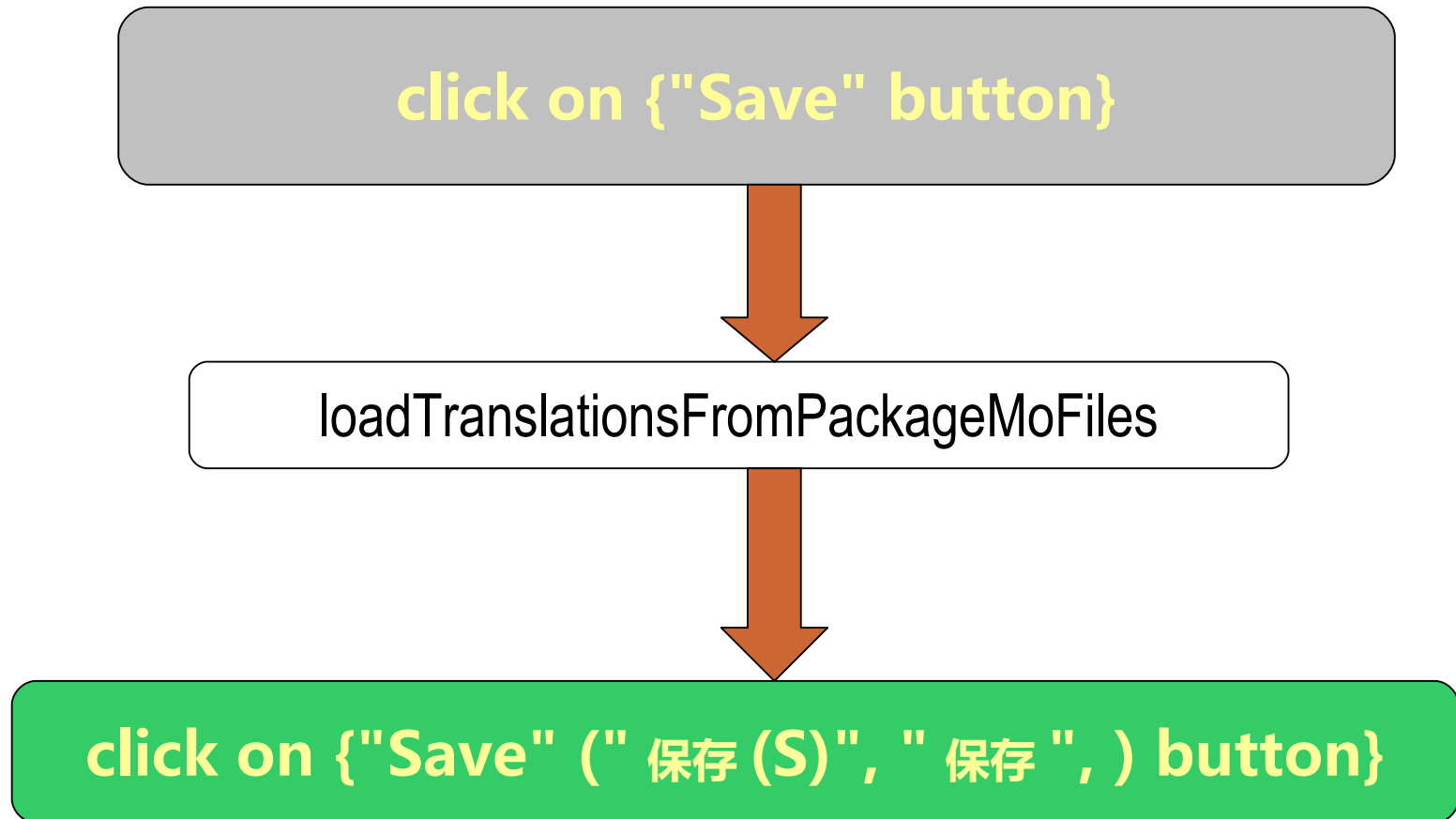
- looks up the named package and finds all gettext MO files within it and its dependent packages

- isMoFile

- >

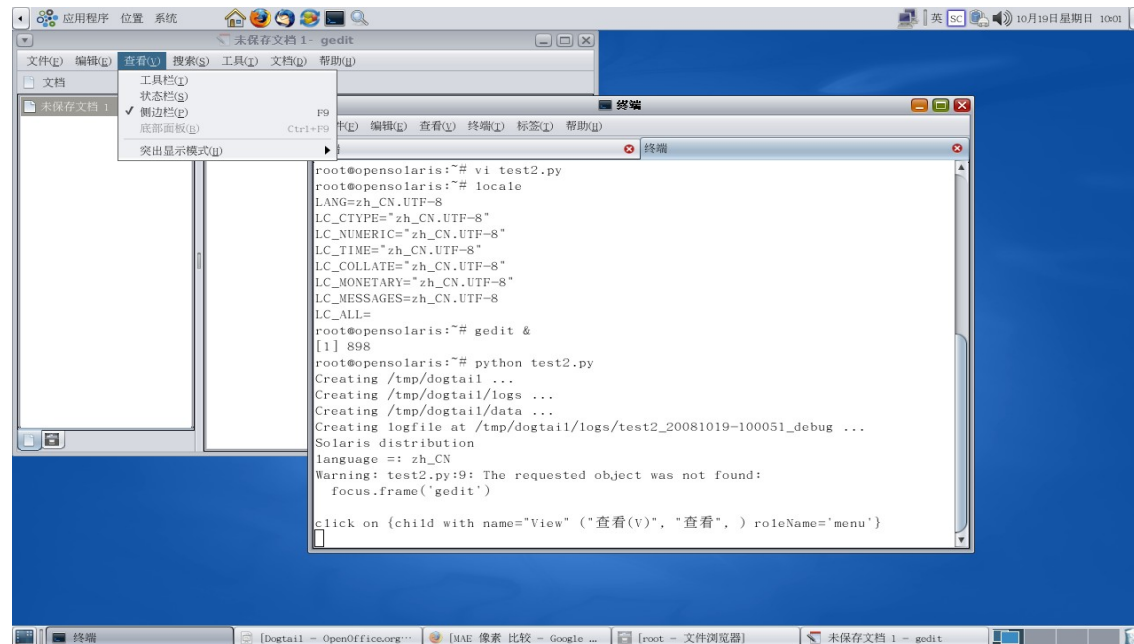
Dogtail – How does it work

- I18n Module



Dogtail – How does it work

- **Screenshot**
 - > Makes use of 'ImageMagic'
 - dogtail.utils.screenshot()
 - import -window ID



```
root@opensolaris:~# vi test2.py
root@opensolaris:~# locale
LANG=zh_CN.UTF-8
LC_CTYPE="zh_CN.UTF-8"
LC_NUMERIC="zh_CN.UTF-8"
LC_TIME="zh_CN.UTF-8"
LC_COLLATE="zh_CN.UTF-8"
LC_MONETARY="zh_CN.UTF-8"
LC_MESSAGES=zh_CN.UTF-8
LC_ALL=
root@opensolaris:~# gedit &
[1] 898
root@opensolaris:~# python test2.py
Creating /tmp/dogtail ...
Creating /tmp/dogtail/logs ...
Creating /tmp/dogtail/data ...
Creating logfile at /tmp/dogtail/logs/test2_20081019-100051_debug ...
Solaris distribution
language = zh_CN
Warning: test2.py:9: The requested object was not found:
  focus.frame('gedit')

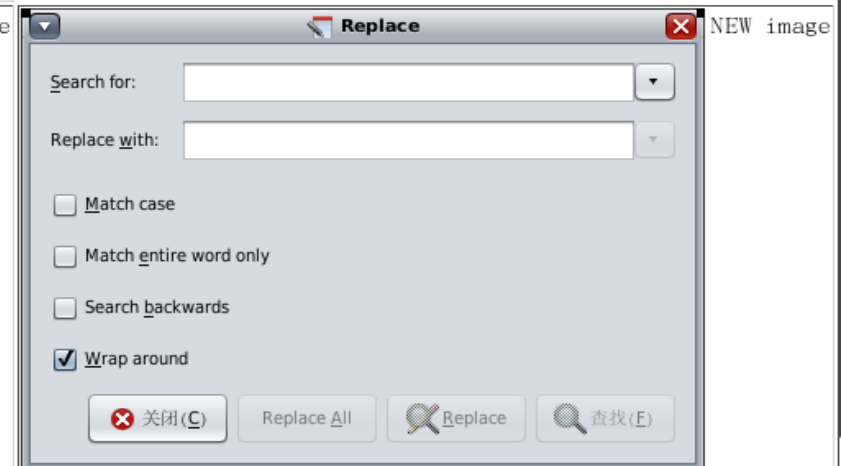
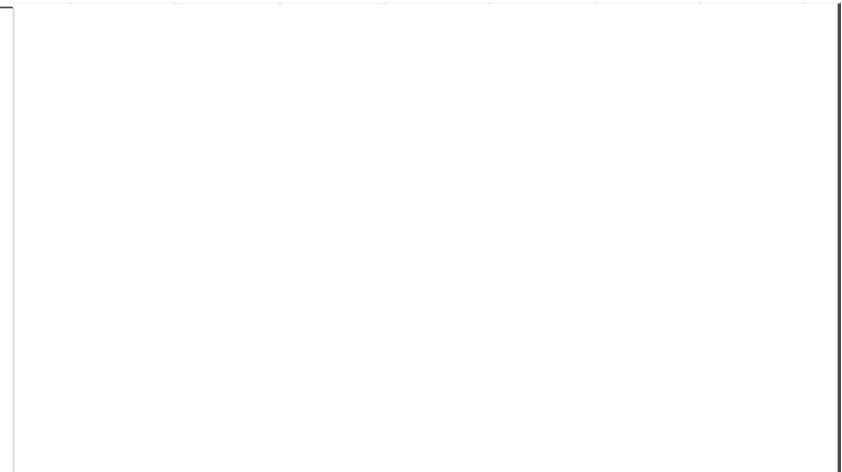
click on {child with name="View" ("查看(V)", "查看", ) roleName='menu'}
```

Dogtail – How does it work

- **Results Comparison**
 - > TCString Class
 - > TCNumber Class
 - > TCImage Class
- Each has a “compare” method
- TCImage utilize ImageMagic to do comparison
 - > `def compare(self, label, baseline, undertest, dfile='default', metric='none', threshold=0.0)`

Dogtail – How does it work

- Comparing with Golden Images



Summary

- Accessibility (A11Y) technologies.
- Python scripts
- Function loadTranslationsFromPackageMoFiles
- ImageMagic
- Reuse automation scripts among locales
- Suitable for regression testing
- Limited to applications using gettext

Reference

- <http://people.redhat.com/zcerza/dogtail/>
- A series of articles written by Len DiMaggio
 - > Automated GUI testing with Dogtail
 - > Dogtail's object oriented tree API
 - > Using Dogtail to automate Frysk GUI tests
- <http://en.wikipedia.org/wiki/Dogtail>
- <http://www.imagemagick.org/>
- i18n-discuss@opensolaris.org

Q&A



Everyone and Everything Participating on the Network



Thanks

harry.fu@sun.com