TileQt and TileGtk: current status

Georgios Petasis

Software and Knowledge Engineering Laboratory,
Institute of Informatics and Telecommunications,
National Centre for Scientific Research "Demokritos",
Athens, Greece
petasis@iit.demokritos.gr

Overview

- Ttk and support for Windows, OS X and Linux
- Qt and GTK+
  - Mapping between Ttk and other widget toolkits
- Interfacing steps
  - The steps required to interface Ttk to another toolkit
- TileQt and TileGTK
  - Current status
- Conclusions

Ttk widgets

- Ttk is the best effort so far in providing Tk widgets with native look under all major operating systems
- Both Microsoft Windows and Apple OS X offer a native widget set
  - Along with a suitable API, that applications can use in order to natively draw widgets
- Wisely, Ttk fully exploits these APIs
  - Making native look for Tk applications a reality
- But what about Linux?

Ttk and Linux

- No native widget set under Linux
  - Thus, no API that can be called
- The Linux desktop is dominated by two environments:
  - KDE, based on the Qt toolkit library
  - GNOME, based on the GTK+ toolkit library
- Both support styled widgets
  - But using the style engine for drawing widgets from outside each toolkit is difficult

Qt and GTK+

- Both widget toolkits offer a public API
- The public API usually targets:
  - Style development
    - ✓So as new widget styles or themes can be developed
  - New widget development
    - ✓So as new or composite widgets can be developed

- TileQt/TileGTK try to use these public APIs in order to draw Ttk widget elements (parts)

Mapping between Ttk and other widget toolkits

- TileQt and TileGTK are quite different from each other, implementation wise

- However, the problems of interfacing Ttk to another toolkit library (that being Qt, GTK or any other library) are exactly the same
  - And seem to be independent of the target toolkit library

Interfacing steps (1)

1. Understand the internals of the library that must be interfaced
   - A time consuming, but feasible task, if the library sources are available
     - ✓ The sources of both Qt and GTK are publically available, along with sufficient documentation

2. Understand how to initialise the library from the hosting application
   - But without initialising the never ending event loop of he library

Interfacing steps (2)

3. Understand how the toolkit library:
   - Locates themes
   - Loads themes
   - Uses themes in order to draw widget elements
5. Find a way to map Tk drawables (windows, pixmaps, etc.) to the drawables of the toolkit library, and vice versa
   - This is an important step, as each library expects its own structures while using its API
   - Not always easy to achieve through the public API
     - ✓ Native structures are usually abstracted to enhance portability

Interfacing steps (3)

5. Find a way to map Ttk widget states to the ones supported by the target toolkit library
   - Not always a straight-forward mapping
     - ✓ Mapping differences usually result in visual differences
7. Separate widgets whose elements can be drawn directly from widgets that are drawn as a whole
   - Widgets of the latter category must:
     - ✓ Drawn in an offline pixmap
     - ✓ Segmented in elements, using available pixel metrics
     - ✓ Elements of interest copied back to Ttk window
   - Segmentation is a vulnerable process
     - ✓ Rarely exact element dimensions can be retrieved

Interfacing steps (4)

7. Ensure thread-safety
   - Easy task if the target toolkit library is already thread safe (i.e. Qt)
   - More tricky if it is not (i.e. GTK, where widgets must be created/drawn only by the thread the library was initialised)

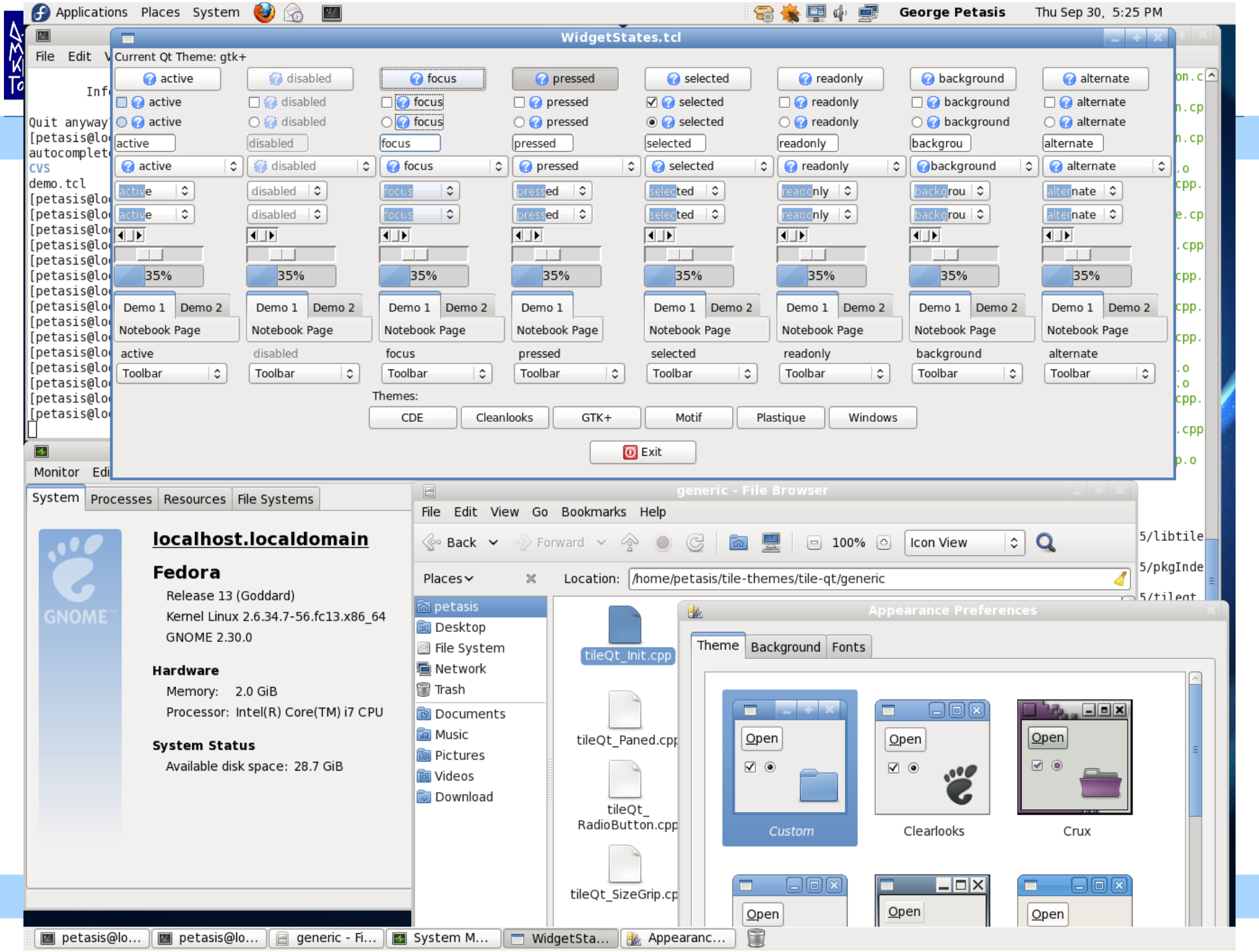
9. Cope with the large number of available themes for each toolkit library
   - Problems range from different layouts to wrong pixel metrics

TileQt: a Ttk interface to the Qt widget toolkit

- One of the first C/C++ extensions that attempted to provide a Ttk theme, based on the Ttk public API, without being part of Ttk
- Development started in 2003
  - Initially supporting Qt 3.x at that time
- Compiling TileQt was never easy
  - As the build system was not robust for quite some time
- Currently, TileQt:
  - Has substantial support for Qt 3.x/4.x
  - Uses a build system based on CMake

TileQt: supported widgets

Widget	Qt 3.x	Qt 4.x	Widget	Qt 3.x	Qt 4.x
Background	☑	☑	LabelFrame	☑	☑
Button	☑	☑	NoteBook	☑	☑
CheckButton	☑	☑	TreeView	☑	☑
RadioButton	☑	☑	Progress	☑	☑
MenuButton	☑	☑	Paned	☑	☑
ToolButton	☑	☑	SizeGrip	☑	☑
Entry	☑	☑	ScrollBar	☑	☒
ComboBox	☑	☑	Scale	☑	☒


Applications Places System
George Petasis
Thu Sep 30, 5:25 PM
WidgetStates.tcl
Current Qt Theme: gtk+
active
disabled
focus
pressed
selected
readonly
background
alternate
Demo 1
Demo 2
Notebook Page
Toolbar
Themes:
CDE
Cleanlooks
GTK+
Motif
Plastique
Windows
Exit
Monitor
System
Processes
Resources
File Systems
localhost.localdomain
Fedora
Release 13 (Goddard)
Kernel Linux 2.6.34.7-56.fc13.x86_64
GNOME 2.30.0
Hardware
Memory: 2.0 GiB
Processor: Intel(R) Core(TM) i7 CPU
System Status
Available disk space: 28.7 GiB
generic - File Browser
File Edit View Go Bookmarks Help
Back
Forward
100%
Icon View
Places
Location: /home/petasis/tile-themes/tile-qt/generic
petasis
Desktop
File System
Network
Trash
Documents
Music
Pictures
Videos
Download
tileQt_Init.cpp
tileQt_Paned.cpp
tileQt_RadioButton.cpp
tileQt_SizeGrip.cp
Appearance Preferences
Theme
Background
Fonts
Open
Custom
Clearlooks
Crux
petasis@lo...
petasis@lo...
generic - Fi...
System M...
WidgetSta...
Appearanc...

TileGTK: a Ttk interface to the GTK+ toolkit

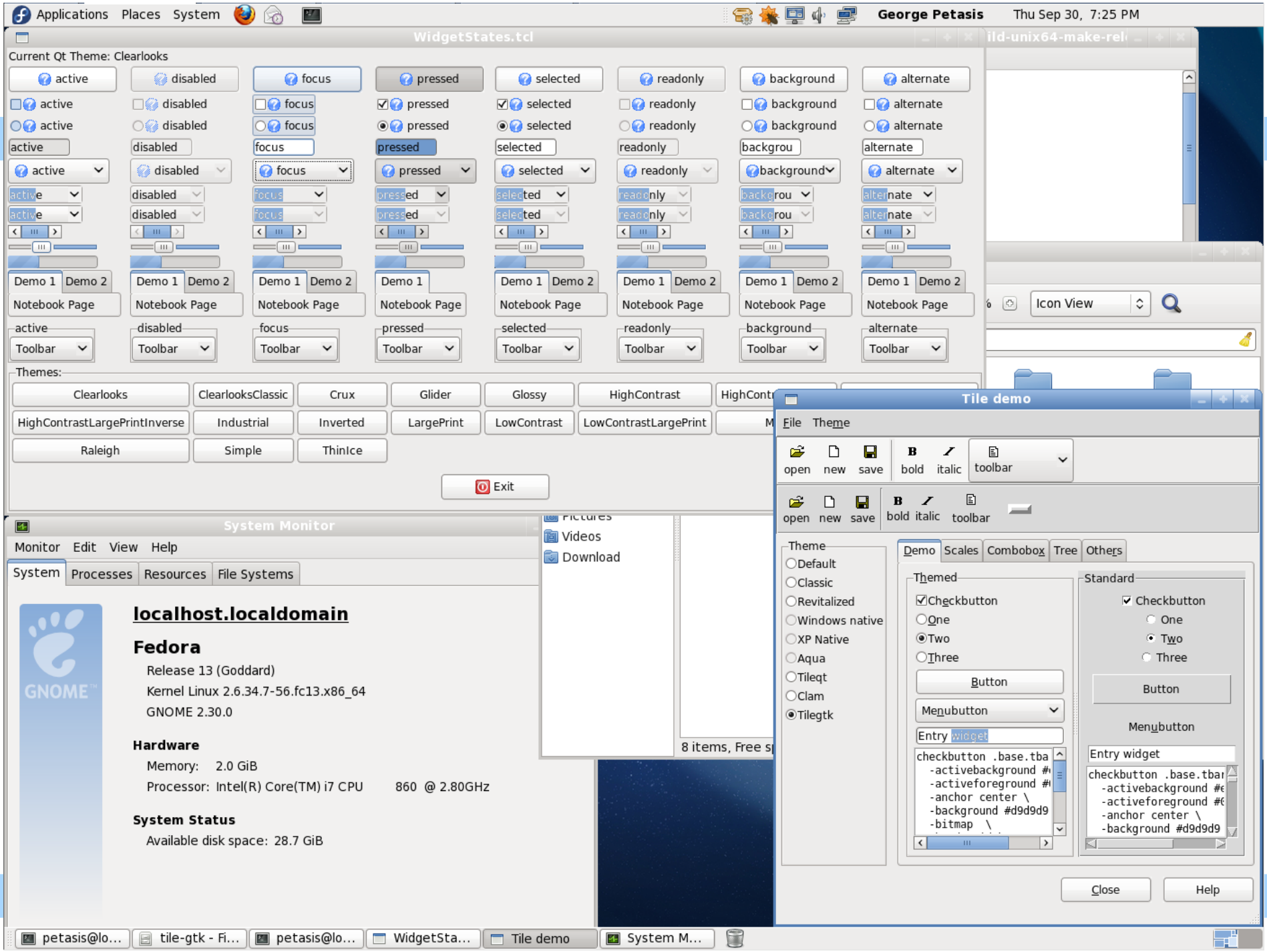
- A far more recent extension, as development started in 2008
- TileGTK was based on the expereince obtained from TileQt
  - In fact, development started by search/replace of Qt to GTK, followed by an gradual adaptation of widgets ☺
- Compiling TileGTK has always been easy
  - As the CMake build system was also inherited

TileGTK and a GTK in C

- GTK+ is written in C, which makes presents an interesting opportunity:
  - To access the GTK+ API through stubs!
  - No need to link with GTK+
    - ✓Which may result in distributing TileGTK binaries under the BSD license
- However, currently Glib facilities are used in order to load and initialise the GTK+ library
  - It would have been better if Tcl offered similar capabilities ☺
- For the time being, TileGTK binaries are covered by GPL

TileGTK: supported widgets

Widget	GTK 2.x	Widget	GTK 2.x
Background	☑	LabelFrame	☑
Button	☑	NoteBook	☑
CheckButton	☑	TreeView	☒
RadioButton	☑	Progress	☑
MenuButton	☑	Paned	☑
ToolButton	☑	SizeGrip	☑
Entry	☑	ScrollBar	☑
ComboBox	☑	Scale	☑


Applications Places System
George Petasis
Thu Sep 30, 7:25 PM
WidgetStates.tcl
Current Qt Theme: Clearlooks
active
disabled
focus
pressed
selected
readonly
background
alternate
Demo 1
Demo 2
Notebook Page
Toolbar
Themes:
Clearlooks
ClearlooksClassic
Crux
Glider
Glossy
HighContrast
HighContrastLargePrintInverse
Industrial
Inverted
LargePrint
LowContrast
LowContrastLargePrint
Raleigh
Simple
ThinIce
Exit
Icon View
Tile demo
File Theme
open
new
save
bold
italic
toolbar
Theme
Default
Classic
Revitalized
Windows native
XP Native
Aqua
Tileqt
Clam
Tilegtk
Demo
Scales
Combobox
Tree
Others
Themed
Standard
Checkbutton
One
Two
Three
Button
Menubutton
Entry widget
Close
Help
System Monitor
Monitor Edit View Help
System
Processes
Resources
File Systems
localhost.localdomain
Fedora
Release 13 (Goddard)
Kernel Linux 2.6.34.7-56.fc13.x86_64
GNOME 2.30.0
Hardware
Memory: 2.0 GiB
Processor: Intel(R) Core(TM) i7 CPU 860 @ 2.80GHz
System Status
Available disk space: 28.7 GiB
Videos
Download
8 items, Free s
petasis@lo...
tile-gtk - Fi...
petasis@lo...
WidgetSta...
Tile demo
System M...

Conclusions (1)

- TileQt & TileGTK try to interface Qt & GTK+ to Ttk
- Both extensions are not actively maintained
  - Mainly because Linux usage by the main author constantly diminishes
- Both TileQt & TileGTK compile and run with recent Tcl versions
  - Tested with ActiveTcl 8.6 beta 3 on an updated Fedora 13 system
  - However loading TileQt/TileGTK in a second interpreter seems to crash wish
    - ✓ No idea why, it used to work

Conclusions (2)

- Some widgets are missing:
  - Separators
  - Scrollbars/Scales (for Qt 4.x)
  - Treeview (for GTK+ 2.x)
- Several visual differences between TileQt/TileGTK and Qt/GTK+
  - TileGTK has more differences
  - TileQt is missing an essential widget (scrollbars)
- Both extensions try to retrieve the colour scheme from the corresponding toolkit
  - Even for GTK+ who has no such capability

Thank you!