The Window Manager Construction Toolkit

KWin goes Scripting

Martin Gräßlin
mgraesslin@kde.org
Akademy 2012
01.07.2012
Agenda

1. History of KWin Scripting
2. Scriptable Types of KWin
3. Influence on Codebase
4. Example of 3rd Party Usage
5. Development of Scripts
6. What’s Next?
Agenda

1. History of KWin Scripting
2. Scriptable Types of KWin
3. Influence on Codebase
4. Example of 3rd Party Usage
5. Development of Scripts
6. What’s Next?
GSoC as Prototype

Google Summer of Code 2010
- Implemented Scripting Support
- API hand-crafted
- API Documentation hand-written
- Strong interweaving of core and scripting
- Scripts invoked at wrong places
- Scripting module undocumented

Prototype
This prototype should never have been merged!
GSoC as Prototype

Google Summer of Code 2010
- Implemented Scripting Support
- API hand-crafted
- API Documentation hand-written
- Strong interweaving of core and scripting
- Scripts invoked at wrong places
- Scripting module undocumented

Prototype
This prototype should never have been merged!
Going Generic: Animation Effect

Issues with Effects
- Many Effects to animate window state changes
- Code got copied over and adjusted
- Changes to Effect System difficult to implement
- Same errors present in many Effects

AnimationEffect
- Base implementation handling animation
- Effects only react on state changes
- Better suited for 3rd Party usage
Going Generic: Animation Effect

Issues with Effects

- Many Effects to animate window state changes
- Code got copied over and adjusted
- Changes to Effect System difficult to implement
- Same errors present in many Effects

AnimationEffect

- Base implementation handling animation
- Effects only react on state changes
- Better suited for 3rd Party usage
Plasma Active
Agenda

1. History of KWin Scripting
2. Scriptable Types of KWin
3. Influence on Codebase
4. Example of 3rd Party Usage
5. Development of Scripts
6. What’s Next?
Window Switcher

Window Switcher Problematic

- Not one size fits all possible
- Caption length very different
- Thumbnails only useful if large
- Icons partially useful
- Effects do not use a toolkit

Solution since 4.8

- Windows are provided in a Model
- QML for GUI
- DeclarativeItem to render a thumbnail
- Documentation on Techbase http://ur1.ca/9jzyw
Window Switcher

Window Switcher Problematic
- Not one size fits all possible
- Caption length very different
- Thumbnails only useful if large
- Icons partially useful
- Effects do not use a toolkit

Solution since 4.8
- Windows are provided in a Model
- QML for GUI
- DeclarativeItem to render a thumbnail
- Documentation on Techbase http://ur1.ca/9jzyw
Desktop Switcher

**Window Switcher’s Little Brother**

- Shares Framework with Window Switcher
- QML Support since 4.9
- Only one available layout
- DeclarativeItem for Desktop Preview missing
QtScript

Primary Scripting Functionality

- Clients exported to Script
- Wrapper around Workspace
- Full access to KWin’s Options
- Everything QProperty based
- Script (un)loading at Runtime through D-Bus
- Global Shortcut Support
- Screen Edge Support
- Configuration Support

```javascript
function synchronizeSwitcher(c) {
    c.skipSwitcher = c.skipTaskbar;
}

function setup(c) {
    synchronizeSwitcher(c);
    c.skipTaskbarChanged.connect(c,
        synchronizeSwitcher);
}

workspace.clientAdded.connect(setup);
// connect all existing clients
var clients = workspace.clientList();
for (var i=0; i<clients.length; i++) {
    setup(clients[i]);
}
Declarative Scripts

GUI for QtScript

- Same API exported as for QtScripts
- Support Plasma Components
- Support Window Switcher’s Thumbnail Item
- Limited Usage: no “real” windows

```cpp
import QtQuick 1.0
import org.kde.kwin 0.1 as KWin
ListView {
    objectName: "listView"
    model: clientModel
    delegate: KWin.ThumbnailItem {
        windowId
        width: 200
        height: 200
    }
}
```
Effect Scripts

**Generic++**
- Based on AnimationEffect
- QtScript based
- No Script execution in rendering phase
- API close to KWin Scripts
- No access to Workspace
- Access to Windows instead of Clients
- Not as elaborated as KWin Scripts

```javascript
var fadeInTime, fadeOutTime, fade;
function loadConfig() {
    fadeInTime = animationTime(effect.readConfig("FadeInTime", 150));
    fadeOutTime = animationTime(effect.readConfig("FadeOutTime", 150));
    fade = effect.readConfig("FadeWindows", true);
}
loadConfig();
effect.configChanged.connect(
    function() {loadConfig();});
effects.windowAdded.connect(function(w) {
    if (fade && isFadeWindow(w)) {
        effect.animate(w, Effect.Opacity, fadeInTime, 1.0, 0.0);
    }
});
```
Window Decorations

Aurorae 3
- Rewritten in QML
- Decoration Bindings
  a side-product
- Theme Preview is interactive

More To Come
QML theme support under development
Agenda

1. History of KWin Scripting
2. Scriptable Types of KWin
3. Influence on Codebase
4. Example of 3rd Party Usage
5. Development of Scripts
6. What’s Next?
Options

**Historic Artifacts**
- Options a singleton to all Config values
- Implemented access to public member variables
- Dependencies between options not ensured
- Design present in first commit

**Fixed with Scripts**
- Member variables are private
- Access only through Getters & Setters
- Dependency between options ensured through setters
- QProperties added
- Documentation added

```cpp
setAutoRaise(config.readEntry("AutoRaise", Options::defaultAutoRaise()));
setAutoRaiseInterval(config.readEntry("AutoRaiseInterval", Options::defaultAutoRaiseInterval()));
setDelayFocusInterval(config.readEntry("DelayFocusInterval", Options::defaultDelayFocusInterval()));

setShadeHover(config.readEntry("ShadeHover", Options::defaultShadeHover()));
setShadeHoverInterval(config.readEntry("ShadeHoverInterval", Options::defaultShadeHoverInterval()));
```
Desktop Change OSD

**Improvements through Script**

- Dropped a Build Option
- Workspace had code to create OSD
- Fixed several bugs
- Removed 700 lines C++ code
- Just 280 lines of QML
Simple Effect

Ported Effects

- Fade
- Fade Desktop
- Prototype for Sheet

More?
Waiting for you!
Simple Effect

Ported Effects

- Fade
- Fade Desktop
- Prototype for Sheet

More?
Waiting for you!
Window Switcher Grid

Present Windows Mode

- Window Switching in Present Windows was a Hack
- Did not work well with Multiple Screens
- Best overview for many windows
- Requires Desktop Effects

Grid Window Switcher Layout

- Does not require Desktop Effects
- Proper Multi-Screen support
- Dropped special handling in effect (180 SLOC)
Introduction of QProperties

Preparing for Wayland

- Describes Client’s interface
- Adds documentation
- Used by Effect System
- 40 properties on Toplevel
- 37 properties on Client
- 65 properties on Options
- 14 properties on Workspace Wrapper
- Most of Extended Window Manager Hints exported

```cpp
class Client
 : public Toplevel
{
  Q_OBJECT
  /**
   * Whether this Client is active or not. Use Workspace::activateClient
   * @see Workspace::activateClient
   **/
  Q_PROPERTY(bool active READ isActive NOTIFY active)
  /**
   * The Caption of the Client. Read from WM_NAME property together with a suffix for hostname and shortcut.
   * To read only the caption as provided by WM_NAME, use the getter with an additional @c false value.
   **/
  Q_PROPERTY(QString caption READ caption NOTIFY captionChanged)
  /**
   * Whether the window can be closed by the user. The value is evaluated each time the getter is called.
   * Because of that no changed signal is provided.
   **/
  Q_PROPERTY(bool closeable READ isCloseable)
  /**
   * The desktop this Client is on. If the Client is on all desktops the property has value -1.
   **/
  Q_PROPERTY(int desktop READ desktop WRITE setDesktop NOTIFY desktopChanged)
```
Multi-Screen Handling

Video-Wall is only valid use case.

- Enable multiple monitor virtual desktop support
- Enable multiple monitor window resistance support
- Enable multiple monitor window placement support
- Enable multiple monitor window maximize support
- Enable multiple monitor window fullscreen support

Replaced by Script
Multi-Screen Handling

- Enable multiple monitor virtual desktop support
- Enable multiple monitor window resistance support
- Enable multiple monitor window placement support
- Enable multiple monitor window maximize support
- Enable multiple monitor window fullscreen support

Replaced by Script

Video-Wall is only valid use case.
Agenda

1. History of KWin Scripting
2. Scriptable Types of KWin
3. Influence on Codebase
4. Example of 3rd Party Usage
5. Development of Scripts
6. What’s Next?
Showstopper: GHNS Support
Window Tiling

https://github.com/mgottschlag/kwin-tiling
Arctos Dashboard

https://github.com/ghinda/arctos-dashboard
Window Switchers

Review Request 103900
A World Beyond Plasma?

KWin would be ready

- Libplasma Dependency moved to runtime
- With Frameworks hardly any dependencies
Agenda

1. History of KWin Scripting
2. Scriptable Types of KWin
3. Influence on Codebase
4. Example of 3rd Party Usage
5. Development of Scripts
6. What’s Next?
WM Console

```javascript
workspace.currentDesktopChanged.connect(function () {
    print("You are now on desktop: ", workspace.currentDesktop);
});
```

Executing script at 26.06.2012 20:26

Runtime: 44ms

You are now on desktop: 2
You are now on desktop: 1
Plasma Package Structure

Plasma Package Documentation: http://ur1.ca/9kkbe
Techbase

- Development/Tutorials/KWin/WindowSwitcher
- Development/Tutorials/KWin/Scripting
- .../API_4.9

kdeexamples git repository

- kde:kdeexamples – kwin/scripts
- Block Compositing for Fullscreen Windows
- Demands Attention only on Current Desktop
- Keep Above only for restored windows
Deployment

For everything else
plasmapkg
Please Test

We need more dogfood!
Agenda

1. History of KWin Scripting
2. Scriptable Types of KWin
3. Influence on Codebase
4. Example of 3rd Party Usage
5. Development of Scripts
6. What’s Next?
Plasmate Integration

Google Summer of Code Project

- Support for Window Switcher
- Support for QtScript
- Support for Declarative Scripts
- Preview for what makes sense
Support for Desktop Thumbnails

Declarative Item

- Review Request 104441
- Requires changes to GL rendering
- Prerequisite to drop BoxSwitch effect
Unit Tests

Scripts to run inside KWin

- Classic Unit Testing for Window Managers difficult
- Script could be injected into running KWin instance
- Adjust Config Options
- Simulate user interaction
- Verify external changes adjust KWin’s internal state
Access to OpenGL?

What about WebGL?

- WebGL binding easy to generate
- What would be the usecase?
- Would conflict with KWin’s GL abstraction
- Requires script execution in rendering loop

Better Approach

- Bindings for KWin’s GL abstraction
- Extend AnimationEffect to support GL
## Access to OpenGL?

**What about WebGL?**
- WebGL binding easy to generate
- What would be the usecase?
- Would conflict with KWin’s GL abstraction
- Requires script execution in rendering loop

**Better Approach**
- Bindings for KWin’s GL abstraction
- Extend AnimationEffect to support GL
Open for Ideas
KWin Development Bof

Thursday, July 5th
10:30 in Room 226