

# PackageKit: Package Abstraction Framework

Richard Hughes

February 24, 2008

# Who am I?

- ▶ My name is Richard Hughes

# Who am I?

- ▶ My name is Richard Hughes
- ▶ Maintainer of GNOME Power Manager

# Who am I?

- ▶ My name is Richard Hughes
- ▶ Maintainer of GNOME Power Manager
- ▶ Co-maintainer of Open Hardware Manager

# Who am I?

- ▶ My name is Richard Hughes
- ▶ Maintainer of GNOME Power Manager
- ▶ Co-maintainer of Open Hardware Manager
- ▶ Contributor to HAL

# Who am I?

- ▶ My name is Richard Hughes
- ▶ Maintainer of GNOME Power Manager
- ▶ Co-maintainer of Open Hardware Manager
- ▶ Contributor to HAL
- ▶ If you are using a laptop **right now** then checkout *<http://www.packagekit.org>*

## The problem

Problems that needed solving

Re-inventing the wheel

This is not working

## Competitors

Windows Vista™

Apple Mac OS X™

Ubuntu

Ximian Red Carpet

## PackageKit: The solution

Introduction

Internal design

Important Ideas

How to use

Questions

# We suck



# Why?



# We suck

- ▶ Because we built kick-ass packaging formats (~1995)

# We suck

- ▶ Because we built kick-ass packaging formats (~1995)
- ▶ And then we layer on top dep-solvers and auto-downloaders (~2000)

# We suck

- ▶ Because we built kick-ass packaging formats (~1995)
- ▶ And then we layer on top dep-solvers and auto-downloaders (~2000)
- ▶ And then we try to bolt on a UI (~2004)

# We suck

- ▶ Because we built kick-ass packaging formats (~1995)
- ▶ And then we layer on top dep-solvers and auto-downloaders (~2000)
- ▶ And then we try to bolt on a UI (~2004)
- ▶ Want to have automatic updates without passwords?

## We suck

- ▶ Because we built kick-ass packaging formats (~1995)
- ▶ And then we layer on top dep-solvers and auto-downloaders (~2000)
- ▶ And then we try to bolt on a UI (~2004)
- ▶ Want to have automatic updates without passwords?
- ▶ Want to use fast-user-switching?

## We suck

- ▶ Because we built kick-ass packaging formats (~1995)
- ▶ And then we layer on top dep-solvers and auto-downloaders (~2000)
- ▶ And then we try to bolt on a UI (~2004)
- ▶ Want to have automatic updates without passwords?
- ▶ Want to use fast-user-switching?
- ▶ Want to have warnings translated into the correct locale?

## We suck

- ▶ Because we built kick-ass packaging formats (~1995)
- ▶ And then we layer on top dep-solvers and auto-downloaders (~2000)
- ▶ And then we try to bolt on a UI (~2004)
- ▶ Want to have automatic updates without passwords?
- ▶ Want to use fast-user-switching?
- ▶ Want to have warnings translated into the correct locale?
- ▶ Want to have common upstream GNOME and KDE tools?

# What should we have done?

- ▶ Studies on user interaction



# What should we have done?

- ▶ Studies on user interaction
- ▶ Design the tools around the use-cases

# What should we have done?

- ▶ Studies on user interaction
- ▶ Design the tools around the use-cases
- ▶ But we can't just do that...

# What should we have done?

- ▶ Studies on user interaction
- ▶ Design the tools around the use-cases
- ▶ But we can't just do that...
- ▶ Hence why I'm here giving this talk...

## What we need to achieve

- ▶ Keeping a system updated automatically

## What we need to achieve

- ▶ Keeping a system updated automatically
- ▶ Installing new software easily

## What we need to achieve

- ▶ Keeping a system updated automatically
- ▶ Installing new software easily
- ▶ Sane user interactions based on formal use-cases

## What we need to achieve

- ▶ Keeping a system updated automatically
- ▶ Installing new software easily
- ▶ Sane user interactions based on formal use-cases
- ▶ Working with native frontends

## What we need to achieve

- ▶ Keeping a system updated automatically
- ▶ Installing new software easily
- ▶ Sane user interactions based on formal use-cases
- ▶ Working with native frontends
- ▶ Working with yum, conary, apt-get, box, zypp, emerge and other backends



## What we need to achieve

- ▶ Keeping a system updated automatically
- ▶ Installing new software easily
- ▶ Sane user interactions based on formal use-cases
- ▶ Working with native frontends
- ▶ Working with yum, conary, apt-get, box, zypp, emerge and other backends
- ▶ Existing package tools should continue to work

# We re-invent the wheel for every distro!

- ▶ Fedora: pup, pirut

# We re-invent the wheel for every distro!

- ▶ Fedora: pup, pirut
- ▶ Ubuntu: gnome-app-install, update-manager

# We re-invent the wheel for every distro!

- ▶ Fedora: pup, pirut
- ▶ Ubuntu: gnome-app-install, update-manager
- ▶ Suse: libzypp command line tools, zen updater

# We re-invent the wheel for every distro!

- ▶ Fedora: pup, pirut
- ▶ Ubuntu: gnome-app-install, update-manager
- ▶ Suse: libzypp command line tools, zen updater
- ▶ Others: same shit different day

# Our current tools suck!

- ▶ Root passwords

# Our current tools suck!

- ▶ Root passwords
- ▶ **Techno-babble:** “GPG?”

# Our current tools suck!

- ▶ Root passwords
- ▶ **Techno-babble:** “GPG?”
- ▶ Not multiple-user or fast-user-switch aware



# Our current tools suck!

- ▶ Root passwords
- ▶ **Techno-babble:** “GPG?”
- ▶ Not multiple-user or fast-user-switch aware
- ▶ GNOME or KDE UI specifications

# Our current tools suck!

- ▶ Root passwords
- ▶ **Techno-babble:** “GPG?”
- ▶ Not multiple-user or fast-user-switch aware
- ▶ GNOME or KDE UI specifications
- ▶ Badly designed and translated

# Our current tools suck!

- ▶ Root passwords
- ▶ **Techno-babble:** “GPG?”
- ▶ Not multiple-user or fast-user-switch aware
- ▶ GNOME or KDE UI specifications
- ▶ Badly designed and translated
- ▶ Power down during a transaction

# Our current tools suck!

- ▶ Root passwords
- ▶ **Techno-babble:** “GPG?”
- ▶ Not multiple-user or fast-user-switch aware
- ▶ GNOME or KDE UI specifications
- ▶ Badly designed and translated
- ▶ Power down during a transaction
- ▶ Updates when on battery power

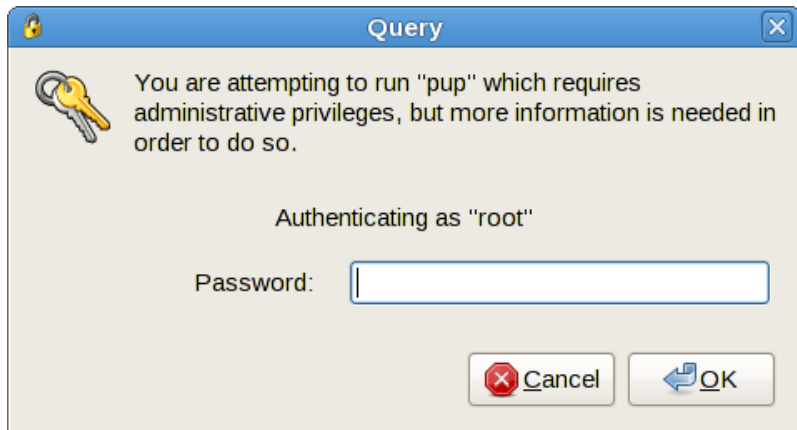
# Linux: Hall of shame



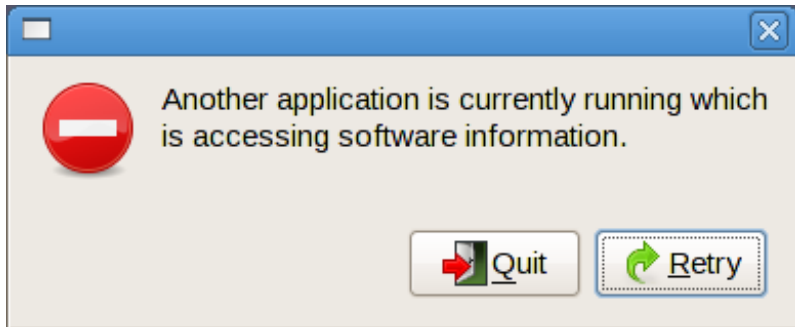
# Linux: Hall of shame



## Linux: Hall of shame



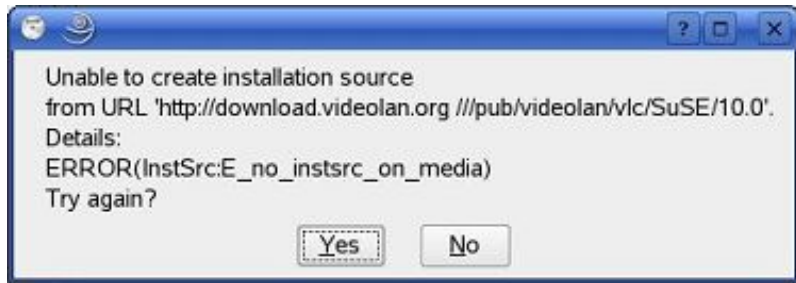
## Linux: Hall of shame







## Linux: Hall of shame



# Linux: Hall of shame



# Windows Vista™: Actually quite good



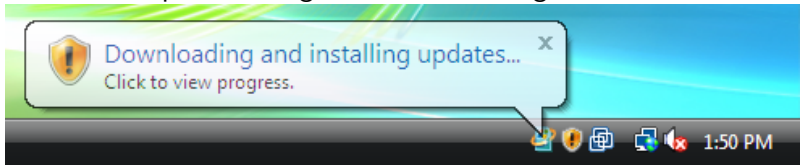
Actually quite good!

# Windows Vista™: Actually quite good

- ▶ Works with fast user switching

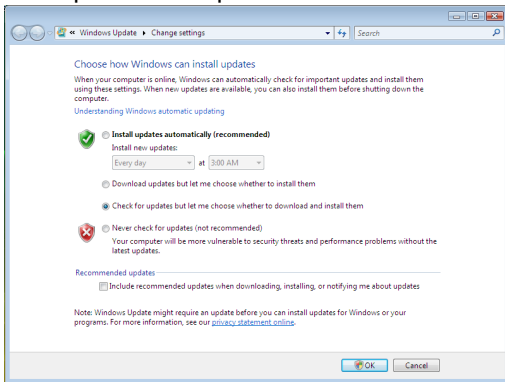
# Windows Vista™: Actually quite good

- ▶ Works with fast user switching
- ▶ Downloads updates using BITS in the background



# Windows Vista™: Actually quite good

- ▶ Works with fast user switching
- ▶ Downloads updates using BITS in the background
- ▶ Sane preferences pane

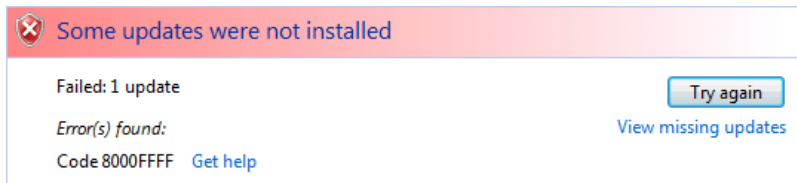


# Windows Vista™: Actually quite good

- ▶ Works with fast user switching
- ▶ Downloads updates using BITS in the background
- ▶ Sane preferences pane
- ▶ Good feedback to the user, default is automatically update.

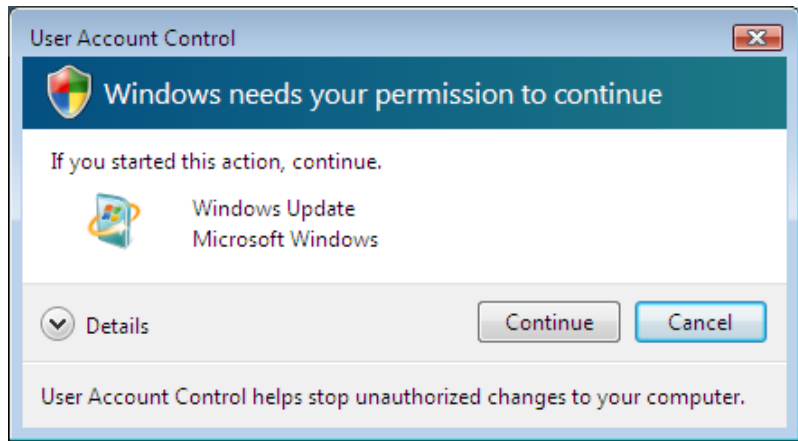


## Vista: Nice error reporting. . .



The screenshot shows a Windows Vista error dialog box with a red header bar. The title is "Some updates were not installed". Below the header, the text reads "Failed: 1 update". Underneath, it says "Error(s) found:" followed by "Code 8000FFFF" and a blue link "Get help". On the right side of the dialog, there is a blue button labeled "Try again" and a blue link "View missing updates".

## Vista: What the hell were they thinking...



# Apple Mac OS X™: The best of the bunch



The best of the bunch!

# Apple Mac OS X™: The best of the bunch

- ▶ Works with fast user switching

# Apple Mac OS X™: The best of the bunch

- ▶ Works with fast user switching
- ▶ Also downloads updates in the background

# Apple Mac OS X™: The best of the bunch

- ▶ Works with fast user switching
- ▶ Also downloads updates in the background
- ▶ Sane preferences pane and update viewer

# Apple Mac OS X™: The best of the bunch

- ▶ Works with fast user switching
- ▶ Also downloads updates in the background
- ▶ Sane preferences pane and update viewer
- ▶ Good feedback to the user, default is automatically update.

## Software Update



### New software is available for your computer.

If you're not ready to install now, you can use the Software Update preference to check for updates later.

Install	Name	Version	Size
<input type="checkbox"/>	AirPort	4.2	12.3 MB
<input type="checkbox"/>	AirPort Update 2005-001	1.0	1.4 MB
<input checked="" type="checkbox"/>	J2SE 5.0 Release 4	4.0	39.3 MB
<input checked="" type="checkbox"/>	QuickTime	7.1.3	49.7 MB
<input checked="" type="checkbox"/>	iTunes	7.0.1	25.7 MB
<input checked="" type="checkbox"/>	Mac OS X Update Combined (P...	10.4.8	153 MB

AirPort 4.2 fixes connection problems and resolves security issues with WPA2 wireless networks.

Note: Use of this software is subject to the original Software License Agreement(s) that accompanied the software being updated. A list of Apple SLAs may be found here: <http://www.apple.com/legal/sla/>.



## Ubuntu: Good try, but no cigar



Good try, but no cigar!

# Ubuntu: Good try, but no cigar

- ▶ No fast user switching

## Ubuntu: Good try, but no cigar

- ▶ No fast user switching
- ▶ Locks other tools out when synaptic or update-manager is loaded

## Ubuntu: Good try, but no cigar

- ▶ No fast user switching
- ▶ Locks other tools out when synaptic or update-manager is loaded
- ▶ No sane preferences pane

## Ubuntu: Good try, but no cigar

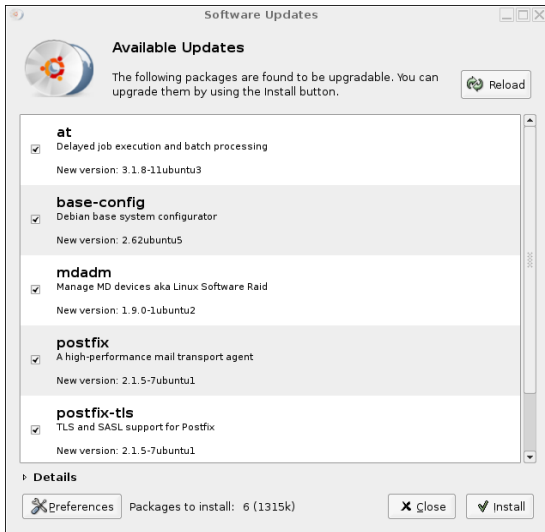
- ▶ No fast user switching
- ▶ Locks other tools out when synaptic or update-manager is loaded
- ▶ No sane preferences pane
- ▶ Update viewer isn't bad

## Ubuntu: Good try, but no cigar

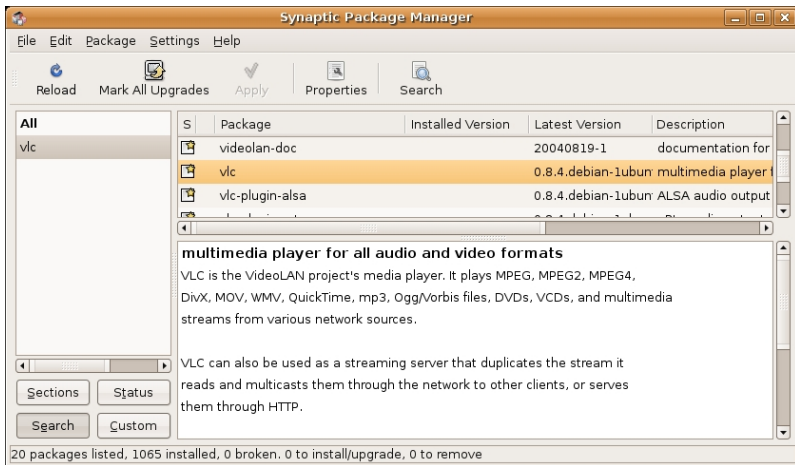
- ▶ No fast user switching
- ▶ Locks other tools out when synaptic or update-manager is loaded
- ▶ No sane preferences pane
- ▶ Update viewer isn't bad
- ▶ Synaptics scares my fiancée

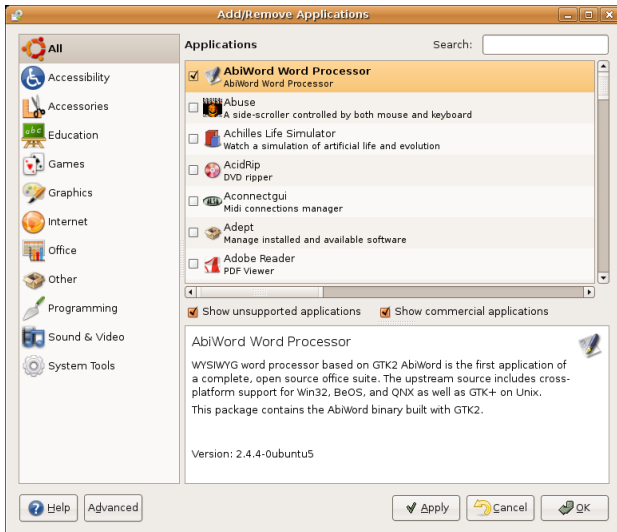
## Ubuntu: Good try, but no cigar

- ▶ No fast user switching
- ▶ Locks other tools out when synaptic or update-manager is loaded
- ▶ No sane preferences pane
- ▶ Update viewer isn't bad
- ▶ Synaptics scares my fiancée
- ▶ gnome-app-install only works with ubuntu, although looks the nutz









# Ximian (now Novell™) Red Carpet: So close, but so far...

- ▶ Well ahead of its time

# Ximian (now Novell™) Red Carpet: So close, but so far...

- ▶ Well ahead of its time
- ▶ Daemon running in the background all the time

# Ximian (now Novell™) Red Carpet: So close, but so far...

- ▶ Well ahead of its time
- ▶ Daemon running in the background all the time
- ▶ Enterprise focus

# Ximian (now Novell™) Red Carpet: So close, but so far...

- ▶ Well ahead of its time
- ▶ Daemon running in the background all the time
- ▶ Enterprise focus
- ▶ Supported `.deb` and `.rpm`

# Ximian (now Novell™) Red Carpet: So close, but so far...

- ▶ Well ahead of its time
- ▶ Daemon running in the background all the time
- ▶ Enterprise focus
- ▶ Supported `.deb` and `.rpm`
- ▶ Replaced the `depsolver` and `downloader`

# Ximian (now Novell™) Red Carpet: So close, but so far...

- ▶ Well ahead of its time
- ▶ Daemon running in the background all the time
- ▶ Enterprise focus
- ▶ Supported .deb and .rpm
- ▶ Replaced the depsolver and downloader
- ▶ Never reached critical mass



# What is PackageKit?



# What is PackageKit?

- ▶ A system designed to make installing and updating software on your computer easier

# What is PackageKit?

- ▶ A system designed to make installing and updating software on your computer easier
- ▶ Unify all the graphical tools

# What is PackageKit?

- ▶ A system designed to make installing and updating software on your computer easier
- ▶ Unify all the graphical tools
- ▶ Make the authentication process suck less

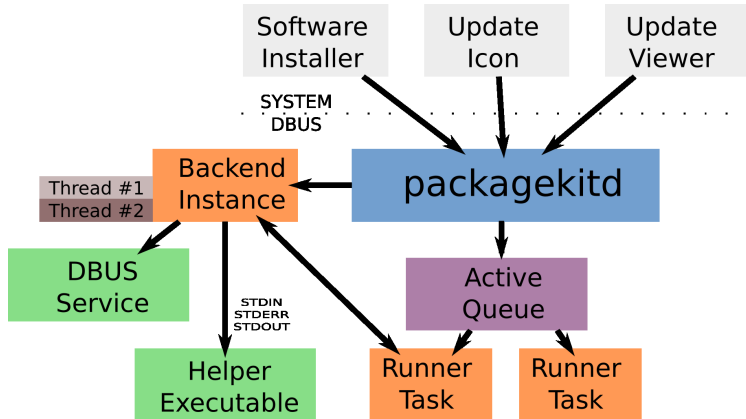
# What is PackageKit?

- ▶ A system designed to make installing and updating software on your computer easier
- ▶ Unify all the graphical tools
- ▶ Make the authentication process suck less
- ▶ Not meant to replace nuts-and-bolts distro tool

# What is PackageKit?

- ▶ A system designed to make installing and updating software on your computer easier
- ▶ Unify all the graphical tools
- ▶ Make the authentication process suck less
- ▶ Not meant to replace nuts-and-bolts distro tool
- ▶ A system activated daemon

# Logical design



# Logical design: Backends

- ▶ A backend is a compiled .so object



# Logical design: Backends

- ▶ A backend is a compiled .so object
- ▶ It provides an interface

## Logical design: Backends

- ▶ A backend is a compiled .so object
- ▶ It provides an interface
- ▶ A backend converts an async request into either a new thread which in the same process, or using external “glue” scripts

## Logical design: Backends

- ▶ A backend is a compiled .so object
- ▶ It provides an interface
- ▶ A backend converts an async request into either a new thread which in the same process, or using external “glue” scripts
- ▶ The only parts that are distro specific are the backends – typically ~900 lines of C or python

## Logical design: Backends

- ▶ A backend is a compiled .so object
- ▶ It provides an interface
- ▶ A backend converts an async request into either a new thread which in the same process, or using external “glue” scripts
- ▶ The only parts that are distro specific are the backends – typically ~900 lines of C or python
- ▶ Backends do not have to be complete

## Quick intro to PolicyKit

- ▶ PackageKit uses PolicyKit for user authentication – therefore the dameon is security sensitive

## Quick intro to PolicyKit

- ▶ PackageKit uses PolicyKit for user authentication – therefore the dameon is security sensitive
- ▶ Can specify with **fine-grained** control

## Quick intro to PolicyKit

- ▶ PackageKit uses PolicyKit for user authentication – therefore the daemon is security sensitive
- ▶ Can specify with **fine-grained** control
- ▶ Unprivileged users can update the system and do searching, but are not allowed to install or remove packages

## Quick intro to PolicyKit

- ▶ PackageKit uses PolicyKit for user authentication – therefore the daemon is security sensitive
- ▶ Can specify with **fine-grained** control
- ▶ Unprivileged users can update the system and do searching, but are not allowed to install or remove packages
- ▶ For home users we can ask the user for their own password



# PackageID

- ▶ This is the `name;version;arch;data` in a single string and is meant to represent a single package

# PackageID

- ▶ This is the `name;version;arch;data` in a single string and is meant to represent a single package
- ▶ The data field can be used for the repository name

# PackageID

- ▶ This is the `name;version;arch;data` in a single string and is meant to represent a single package
- ▶ The data field can be used for the repository name
- ▶ A PackageID must be enough to uniquely identify an object

# PackageID

- ▶ This is the `name;version;arch;data` in a single string and is meant to represent a single package
- ▶ The data field can be used for the repository name
- ▶ A PackageID must be enough to uniquely identify an object
- ▶ PackageID's are not shown to the user

# Filters

- ▶ Search filtering is done in the backend

# Filters

- ▶ Search filtering is done in the backend
- ▶ `installed` : if the package is installed on the system

# Filters

- ▶ Search filtering is done in the backend
- ▶ `installed` : if the package is installed on the system
- ▶ `devel` : development packages typically end `-devel`, `-dgb` and `-static`

# Filters

- ▶ Search filtering is done in the backend
- ▶ `installed` : if the package is installed on the system
- ▶ `devel` : development packages typically end `-devel`, `-dgb` and `-static`
- ▶ `gui` : GUI programs typically depend on `gtk`, `kde` or `xfce`.



# Filters

- ▶ Search filtering is done in the backend
- ▶ `installed` : if the package is installed on the system
- ▶ `devel` : development packages typically end `-devel`, `-dgb` and `-static`
- ▶ `gui` : GUI programs typically depend on `gtk`, `kde` or `xfce`.
- ▶ `free` : Free software defined by the FSF.

# Filters

- ▶ Search filtering is done in the backend
- ▶ `installed` : if the package is installed on the system
- ▶ `devel` : development packages typically end `-devel`, `-dgb` and `-static`
- ▶ `gui` : GUI programs typically depend on `gtk`, `kde` or `xfce`.
- ▶ `free` : Free software defined by the FSF.
- ▶ `visible` : Should show in the UI.

# Filters

- ▶ Search filtering is done in the backend
- ▶ `installed` : if the package is installed on the system
- ▶ `devel` : development packages typically end `-devel`, `-dgb` and `-static`
- ▶ `gui` : GUI programs typically depend on `gtk`, `kde` or `xfce`.
- ▶ `free` : Free software defined by the FSF.
- ▶ `visible` : Should show in the UI.
- ▶ `supported` : Is supported by the distro.

# Process communication

- ▶ If a backend can't operate in a thread

# Process communication

- ▶ If a backend can't operate in a thread
- ▶ Send standard stdin commands to the process, e.g. `force liferea;2.20.1;i386;fedora`

# Process communication

- ▶ If a backend can't operate in a thread
- ▶ Send standard stdin commands to the process, e.g. `force liferea;2.20.1;i386;fedora`
- ▶ Process UNIX stderr and stdout

# Process communication

- ▶ If a backend can't operate in a thread
- ▶ Send standard stdin commands to the process, e.g. `force liferea;2.20.1;i386;fedora`
- ▶ Process UNIX stderr and stdout
- ▶ Send the process UNIX signals for cancellation of processes

# Cancellation

- ▶ `AllowInterrupt` is called if the daemon is able to abort the transaction by killing the process



# Cancellation

- ▶ AllowInterrupt is called if the daemon is able to abort the transaction by killing the process
- ▶ Cancel a transaction in a download phase, not in an install phase

# Cancellation

- ▶ `AllowInterrupt` is called if the daemon is able to abort the transaction by killing the process
- ▶ Cancel a transaction in a download phase, not in an install phase
- ▶ Threaded backends, `Cancel()` method used to terminate the thread

# Cancellation

- ▶ `AllowInterrupt` is called if the daemon is able to abort the transaction by killing the process
- ▶ Cancel a transaction in a download phase, not in an install phase
- ▶ Threaded backends, `Cancel()` method used to terminate the thread
- ▶ For spawned backends, we send the process `SIGQUIT`, then `SIGKILL`

## Error enumerated types

- ▶ Every error should have an enumerated type and also a description

## Error enumerated types

- ▶ Every error should have an enumerated type and also a description
- ▶ The description is not translated and not converted into the users locale

## Error enumerated types

- ▶ Every error should have an enumerated type and also a description
- ▶ The description is not translated and not converted into the users locale
- ▶ Examples; out-of-memory, not-supported, package-not-installed

# Transaction ID

- ▶ Unique identifier that identifies the present or past transaction

# Transaction ID

- ▶ Unique identifier that identifies the present or past transaction
- ▶ One or more sub-transactions



# Transaction ID

- ▶ Unique identifier that identifies the present or past transaction
- ▶ One or more sub-transactions
- ▶ One *role* for the entire lifetime

# Transaction ID

- ▶ Unique identifier that identifies the present or past transaction
- ▶ One or more sub-transactions
- ▶ One *role* for the entire lifetime
- ▶ Different *status* values as the transaction is processed

# Transaction ID

- ▶ Unique identifier that identifies the present or past transaction
- ▶ One or more sub-transactions
- ▶ One *role* for the entire lifetime
- ▶ Different *status* values as the transaction is processed
- ▶ An example TransactionID would be  
`45;dafeca;checkpoint32`

## Methods

- ▶ Search Name
- ▶ Search Group
- ▶ Search Details
- ▶ Search File
- ▶ Install Package
- ▶ Install File
- ▶ Remove Package
- ▶ Get Depends
- ▶ Resolve

## Methods 2

- ▶ Get Update Detail
- ▶ Get Requires
- ▶ Get Description
- ▶ Get Files
- ▶ Update Package
- ▶ Update System
- ▶ Get Updates
- ▶ Refresh Cache
- ▶ Get Repo List
- ▶ Repo Enable
- ▶ Repo Set Data
- ▶ Service Pack

# Signals

- ▶ Progress
- ▶ Error
- ▶ Status
- ▶ RequireRestart
- ▶ Package
- ▶ Description
- ▶ UpdateDetail
- ▶ RepoSignatureRequired

# How do I use PackageKit?

- ▶ pkcon command line queries

```
[hughsie@laptop ~]$ pkcon get updates
normal    powertop    i386    1.8-1.fc8    fedora    Power cons
security  kernel      i386    2.6.23-0.115.rc3.git1.fc8 instal
security  gtkhtml2    i386    2.19.1-4.fc8 fedora    An HTML w
Runtime was 0.1 seconds
```

## How do I use PackageKit?

- ▶ pkcon command line searching

```
[hughsie@laptop ~]$ pkcon search name power
installed  powerman                i386  1.0.25-2.fc7  installed
installed  powertop                       i386  1.8-1.fc8    installed
installed  gnome-power-manager            i386  2.20.0-5.fc8 installed
available  kpowersave                    i386  0.7.3-0.2svn20070828.1
available  kadu-powerkadu                 i386  0.5.0-4.fc8  developme
available  powermanga                     i386  0.90-1       developme
Runtime was 2.1 seconds
```



## How do I use PackageKit?

- ▶ libpackagekit GObject library

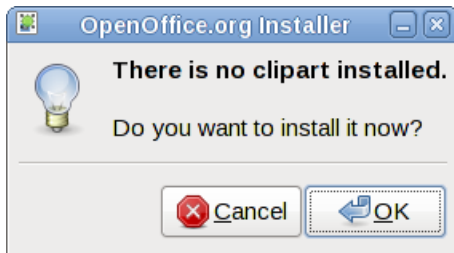
```
PkClient *client;  
client = pk_client_new ();  
pk_client_install_package (client, "openoffice-clipart");  
g_signal_connect (client, "finished",  
                  G_CALLBACK (pk_console_finished_cb), 0);  
...  
g_object_unref (client);
```

# Why is this useful?

- ▶ OpenOffice has missing clipart

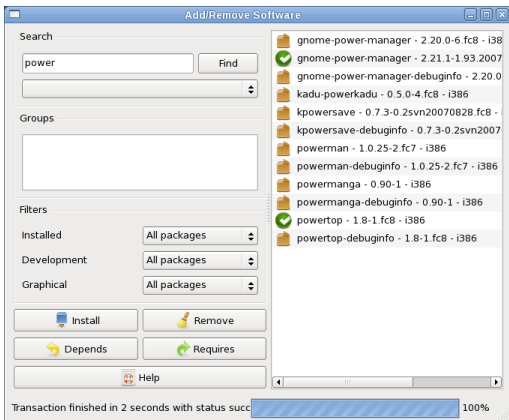
## Why is this useful?

- ▶ OpenOffice has missing clipart
- ▶ It prompts the user with:



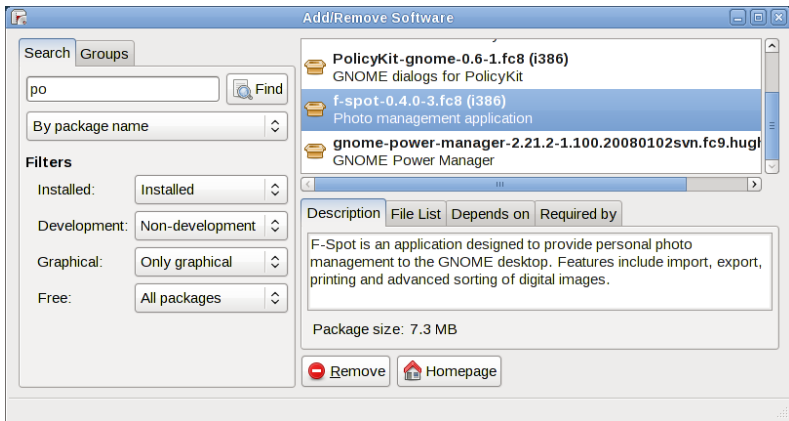
# QPackageKit frontend

## ► QPackageKit QT frontend



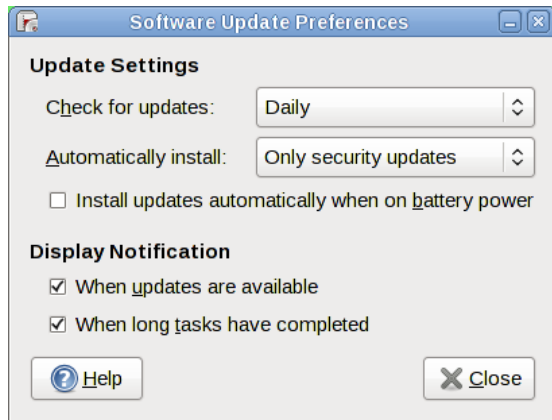
# GNOME Frontend

## ► pk-application



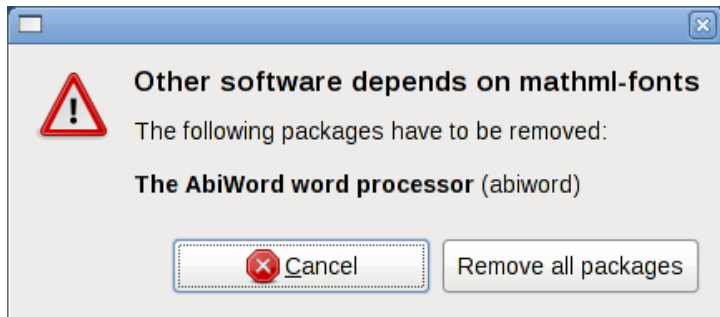
# GNOME Frontend

## ► pk-prefs



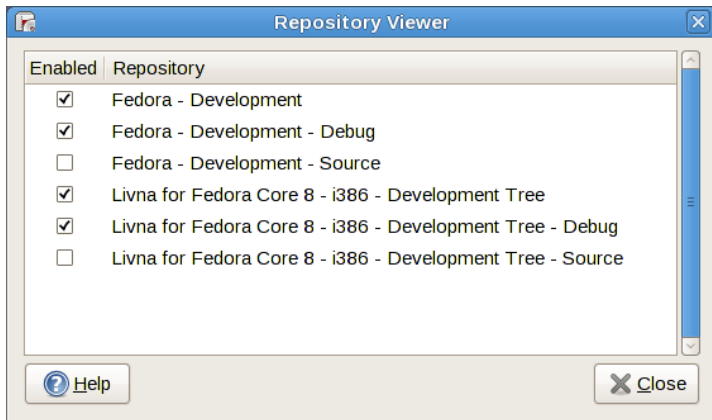
# GNOME Frontend

## ► Removing deps



# GNOME Frontend

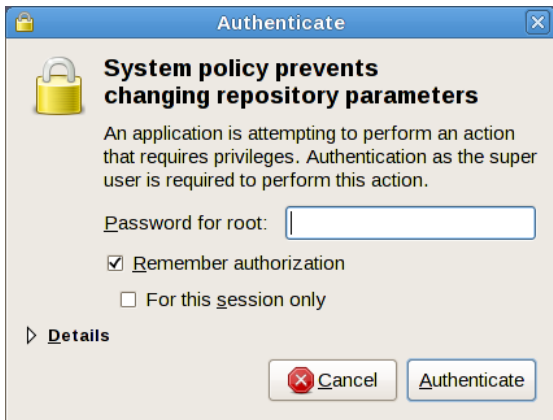
## ► pk-repo





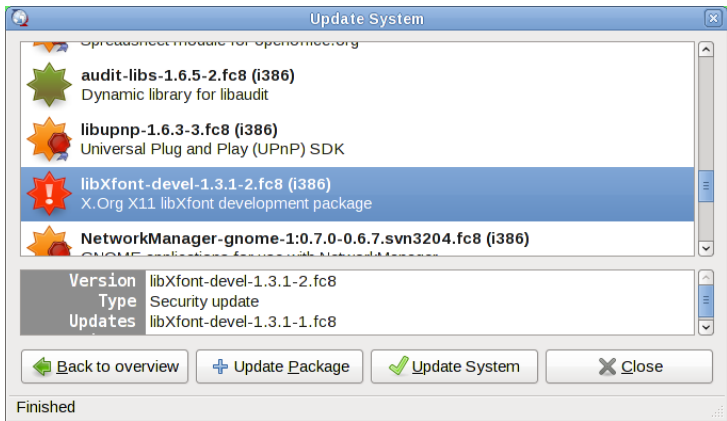
# GNOME Frontend

## ► pk-application



# GNOME Frontend

## ► pk-update-viewer



# GNOME Frontend

## ► pk-update-icon



# Development

- ▶ Done on [packagekit.org](https://packagekit.org) git server

# Development

- ▶ Done on [packagekit.org](https://packagekit.org) git server
- ▶ Fast server on a good connection (thanks Ken and Elliot)

# Development

- ▶ Done on [packagekit.org](https://packagekit.org) git server
- ▶ Fast server on a good connection (thanks Ken and Elliot)
- ▶ Developers push and pull frequently during the day

# Development

- ▶ Done on packagekit.org git server
- ▶ Fast server on a good connection (thanks Ken and Elliot)
- ▶ Developers push and pull frequently during the day
- ▶ Create git+ssh accounts in a few seconds

# Development

- ▶ Done on [packagekit.org](http://packagekit.org) git server
- ▶ Fast server on a good connection (thanks Ken and Elliot)
- ▶ Developers push and pull frequently during the day
- ▶ Create `git+ssh` accounts in a few seconds
- ▶ Commit if you have a single patch to apply



# Development

- ▶ Done on [packagekit.org](http://packagekit.org) git server
- ▶ Fast server on a good connection (thanks Ken and Elliot)
- ▶ Developers push and pull frequently during the day
- ▶ Create `git+ssh` accounts in a few seconds
- ▶ Commit if you have a single patch to apply
- ▶ Once a day synced with anonymous freedesktop

## Current status

- ▶ Last release was 0.1.8

## Current status

- ▶ Last release was 0.1.8
- ▶ Backends already fully or partially supported: conary, yum, apt, box, alpm, smart, opkg, pisi, zypp

## Current status

- ▶ Last release was 0.1.8
- ▶ Backends already fully or partially supported: conary, yum, apt, box, alpm, smart, opkg, pisi, zypp
- ▶ Shipped by default in Foresight Linux and the GNOME Developer Kit.

## Current status

- ▶ Last release was 0.1.8
- ▶ Backends already fully or partially supported: conary, yum, apt, box, alpm, smart, opkg, pisi, zypp
- ▶ Shipped by default in Foresight Linux and the GNOME Developer Kit.
- ▶ To be shipped in Fedora 9.

## Current status

- ▶ Last release was 0.1.8
- ▶ Backends already fully or partially supported: conary, yum, apt, box, alpm, smart, opkg, pisi, zypp
- ▶ Shipped by default in Foresight Linux and the GNOME Developer Kit.
- ▶ To be shipped in Fedora 9.
- ▶ Interest from Ubuntu, openSUSE, openSolaris, Mandriva and a few more

# Releases

- ▶ Rapid development

# Releases

- ▶ Rapid development
- ▶ A minor release a month



# Releases

- ▶ Rapid development
- ▶ A minor release a month
- ▶ 15-strong developer team

# Releases

- ▶ Rapid development
- ▶ A minor release a month
- ▶ 15-strong developer team
- ▶ PackageKit, QPackageKit and gnome-packagekit are GPLv2+

# Why should I use PackageKit

- ▶ Free up developers

# Why should I use PackageKit

- ▶ Free up developers
- ▶ Working together is much more sane

# Why should I use PackageKit

- ▶ Free up developers
- ▶ Working together is much more sane
- ▶ Translations and native frontends for free

# Why should I use PackageKit

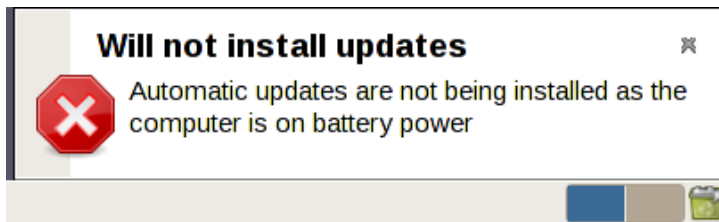
- ▶ Free up developers
- ▶ Working together is much more sane
- ▶ Translations and native frontends for free
- ▶ Only provide a small amount of support

# Why should I use PackageKit

- ▶ Free up developers
- ▶ Working together is much more sane
- ▶ Translations and native frontends for free
- ▶ Only provide a small amount of support
- ▶ Thriving development team

## Cool stuff we can do

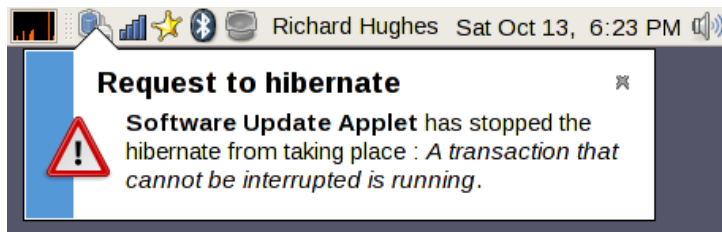
- ▶ Battery mode detection





# Cool stuff we can do

- ▶ System and session inhibit



# Problems not yet properly solved

- ▶ Time remaining

# Problems not yet properly solved

- ▶ Time remaining
- ▶ Reporting bugs in backends

# Problems not yet properly solved

- ▶ Time remaining
- ▶ Reporting bugs in backends
- ▶ Installing service packs

# Recap

- ▶ Discussed the downsides of the tools we've got already

# Recap

- ▶ Discussed the downsides of the tools we've got already
- ▶ We've seen how other operating systems do it

# Recap

- ▶ Discussed the downsides of the tools we've got already
- ▶ We've seen how other operating systems do it
- ▶ I've introduced PackageKit and the concept of distribution specific backends

# Recap

- ▶ Discussed the downsides of the tools we've got already
- ▶ We've seen how other operating systems do it
- ▶ I've introduced PackageKit and the concept of distribution specific backends
- ▶ Console, QT and GNOME tools can interact with PackageKit



# Recap

- ▶ Discussed the downsides of the tools we've got already
- ▶ We've seen how other operating systems do it
- ▶ I've introduced PackageKit and the concept of distribution specific backends
- ▶ Console, QT and GNOME tools can interact with PackageKit
- ▶ Distributions that adopt PackageKit can share common code

# Recap

- ▶ Discussed the downsides of the tools we've got already
- ▶ We've seen how other operating systems do it
- ▶ I've introduced PackageKit and the concept of distribution specific backends
- ▶ Console, QT and GNOME tools can interact with PackageKit
- ▶ Distributions that adopt PackageKit can share common code
- ▶ There is strong positive momentum with PackageKit

## PackageKit questions



- ▶ I'm expecting lots of difficult questions!
- ▶ Email: `richard@hughsie.com`
- ▶ Homepage: `http://www.packagekit.org`