Automatic Printer Driver Installation in Fedora 13

Presenter

Tim Waugh
Senior Software Engineer, Red Hat

Automatic Printer Driver Installation in Fedora 13

- The idea
- Current status
- Driver requirements
- Limitations

The idea

- PackageKit used to automatically install resources
- Special tags added to packages at build time
- Run-time hooks for installing packages:
  - Codecs (gstreamer)

    https://fedoraproject.org/wiki/Features/GStreamer_dependencies_in_RPM
  - Fonts (GTK+)
  - MIME type handlers (nautilus)

    https://fedoraproject.org/wiki/Features/AutoFontsAndMimeInstaller
  - Why not printer drivers?

The idea

- Offer to install all drivers which declare support
  - Nothing installed without user interaction
- system-config-printer then chooses “best” driver from those installed (as normal)

The idea

- Drivers declare the IEEE 1284 Device IDs they expect using existing mechanisms
- Add PackageKit queries:
  - Direct: add to existing udev hook
  - Network: add to system-config-printer before driver search

What about Jockey?

- Jockey: infrastructure for providing 3rd party drivers for system hardware
- Query is entirely OpenPrinting-based (packages not tagged with IEEE 1284 Device IDs)
- Wanted a more “PackageKit”-y solution
  - Applications can formulate their own queries
    - what packages provide a driver for this model?
    - which ones are installed? which have free licenses?
  - 3rd party repositories configured “as normal”

Current status

- RPM support done and “provides” script written
- PackageKit
  - API adjusted (`POSTSCRIPTDRIVER` 'provides' enum)
  - yum backend updated
- gnome-packagekit, kpackagekit
  - API adjusted (`InstallPrinterDrivers` method)
- system-config-printer (incl udev hook) updated
- Driver packages adjusted where necessary and rebuilt

Driver Requirements

- Packaging
  - Build requires pycups, cups
- Add IEEE 1284 Device ID tags for each device
  - **PPD files**

    ```
    *1284DeviceID: “MFG:…;MDL:…;”
    ```

  - **Driver information files** (CUPS DDK .drv)

    ```
    Attribute “1284DeviceID” “” “MFG:…;MDL:…;”
    ```

  - **PPD driver programs** (dynamically generated)

    Fill in 5th field in `list` output – see cups-driverd(8)

    Must execute in build root!

Driver Adjustments Required

Foomatic PPD files | PPD driver program

- Database (including PPDs) in `foomatic-db` source pkg
  - Device ID tags picked up from PPD files (but few)
- Filters/RIP/driver program in `foomatic` source package
  - Perl include path wrong for build root
  - Build requires `foomatic` – slightly hacky
  - Build requires `foomatic-db`
  - Device ID tags picked up from driver program

Driver Adjustments Required

HPLIP PPD files | Driver information file

IEEE 1284 Device IDs already listed:

- some PPD files
- hpcups.drv file

...but largely incorrect

- hpcups.drv corrected from foomatic data

Driver Adjustments Required

Gutenprint PPD driver program

- CUPS driver program reads XML files
- Did not declare IEEE 1284 Device IDs

↓

- Adjusted to use $DISTDIR if set
- XML adjusted to store Device ID for each model
- Populated from foomatic data

Limitations / Future Directions

- Only implemented for:
  - RPM (2 lines in PackageKit + provides script)
  - Gnome and KDE (`InstallPrinterDrivers` D-Bus method in `org.freedesktop.PackageKit.Modify` interface)
- Will not install CUPS (or start service)
- Will not discover generic drivers
  - CMD-based matching? `*1284DeviceID: “CMD:PCL;”`
- Inaccurate IEEE 1284 Device IDs in drivers

Limitations / Future Directions

- Not all printers provide Device IDs on all interfaces
  - DNS-SD, SNMP, USB, etc
- Foomatic drivers can't generally be installed this way
  - Binaries referenced by XML in `foomatic-db`
    - e.g. min12xxw
  - foomatic-kitload (XML provided by driver)
    - e.g. gutenprint-foomatic
    - Newer drivers should provide driver information files instead

  Not currently foomatic-db dependencies but could be

Thanks

- Questions?