

Software Installation/Management

Thorsten Kukuk
Technical Project Manager
kukuk@suse.de

December 6, 2006



Novell.[®]

Rationale

Distributions are too fat

not all *requires* are required

- > weak dependencies

Lots of knowledge in installers

which drivers, fonts, translations ?

- > reverse dependencies

Fight package flood / support appliances

take building block approach

- > patterns

Support of management standards

CIM (Common Information Model) modeling

- > object hierarchy

Management Standards

CIM defines object hierarchy / abstraction

- Product
 - Consists of patterns and packages
 - Defines dependencies to other products
- Pattern
 - Group of packages for a special purpose
 - Can define configuration workflow for this group of packages
- Package (RPM)
 - Consists of binaries and data/config files
 - Contains dependencies for software

Product Management

Different Products

- Base Products (openSUSE, SLED, SLES)
 - > Fully self-contained
- Add-On Products (SLE SDK, SLE Real Time, SGI Propack)
 - > Requires a Base Product, which fulfills the requirements regarding:
YaST2, Base System, special versions of packages, ...

Obsolete old products

- Remove predecessor products, but not different Base/Add-On products

Require pattern/packages

- What defines the product ?

Patterns

Building block approach

- Group packages and patterns, build hierarchy

Full set of dependencies

- Can target patterns or packages
- Requires (strong), Recommends (weak), Suggests (ask user)
- Require/obsolete other patterns
- Enhance other patterns
 - > Install 32bit runtime packages on 64bit system depending on selected patterns

Run installer/configuration YaST2 module

- Which configuration modules have to be called

Existing Dependencies

Requires

- A *strong* dependency, all requirements must be fulfilled

Provides

- Exposed capabilities of this object, they can be used to match Requires from others

Conflicts

- This object cannot be installed if another object is installed and provides matching capabilities

Obsoletes

- When this object is installed, it will remove any other object with a matching capability

Weak dependencies

Help the software management application

- **Recommends**

- Weak requires, break up dependency graphs
- Don't complain if requirement cannot be fulfilled

- **Suggests**

- Application hint
- Think of Amazons 'Customers who bought this item also bought'

See http://en.opensuse.org/Software_Management/Dependencies

Reverse dependencies

Let objects tell if/when they should be installed

- **Supplements**

- Issuer *should* be installed if
 - > hardware drivers
 - > languages (fonts, input methods, translations)

- **Enhances**

- Issuer *may* be installed if
 - > reverse suggests (application hint)

RPMs – Problems/Missing features

Rename of package

- Project has changed upstream (ethereal -> wireshark), on update new package should be installed and replace old one

Package split

- New subpackage should be installed on update

Documentation

http://en.opensuse.org/Software_Management/Content

Novell®

General Disclaimer

This document is not to be construed as a promise by any participating company to develop, deliver, or market a product. Novell, Inc., makes no representations or warranties with respect to the contents of this document, and specifically disclaims any express or implied warranties of merchantability or fitness for any particular purpose. Further, Novell, Inc., reserves the right to revise this document and to make changes to its content, at any time, without obligation to notify any person or entity of such revisions or changes. All Novell marks referenced in this presentation are trademarks or registered trademarks of Novell, Inc. in the United States and other countries. All third-party trademarks are the property of their respective owners.

