**Patching Package**

Patching on Solaris

# **What’s patch?**

A patch to a package is just a sparse package designed to overwrite certain files in the original.

# **Patch Directory Structure**

It’s easier if you create all directory following.

# **Instruction**

1. Prepare your all files and copy all to source folder.
2. Prepare your scripts files that consist of checkinstall, preinstall postinstall, patch\_checkinstall, patch\_postinstall and i.none. All of the scripts you can use template.
   1. **Checkinstall** script
   2. **Preinstall** script, it initializes the prototype file, information files, and installation scripts for the backout package to be constructed. This script is very simple and the remaining scripts in this example only allow a backout package to describe regular files.
   3. **Postinstall** script, it creates the backout package using the information provided by the other scripts. Since the **pkgmk** and **pkgtrans** commands do not require the package database, they can be executed within a package installation.
   4. **Patch\_checkinstall** script,
   5. **Patch\_postinstall** script,
   6. **i.none** script,
3. Create *“****pkginfo****”* file. It’s information of package.  
   Actually, it is crucial that the **VERSION** number of the patch package be the same as that of the original package. You must to add “Patch\_label” which is version or name of patch.

PKG=**equinox**

NAME=**equinox version XX.XX.XX**

ARCH=**sparcv9**

VERSION=**XX.XX.XX**

CATEGORY=**application**

BASEDIR=**/opt**

CLASSES=**none**

*Patch\_label=****EQXPATCH-XX.XX.XX-YY***

1. Creating a **prototype** file with the **pkgproto** command. It’s easier to create prototype

$ **pkgproto** [Your source file] > [Prototype name]

1. Example “pkgproto ./equinox > /build/InfoFile/prototype”

Result:

$ **pkgproto** ./equinox > /build/InfoFile/prototype

d equinox/bin 0755 toro root

f equinox/bin/E00 0755 toro root

f equinox/bin/E01 0755 toto root

…

1. Use the **pkgmk** command to build your package. This command takes all of the objects defined in the “**prototype**” file, puts them into directory format, creates the **pkgmap** file, and produces an installable package to be used as input to the **pakadd** command.

$ cd /build/InfoFiles/

$ **pkgmk –o –r ../**source/ -d ../Released/ -f ./prototype

$ cd /build/Released/

$ ls – lrt

equinox

…

1. Use the **pkgtrans** command to compress your package. You get completely package that able to install with **pkgadd** command.

For instance,

Pkgtrans –s [source file package] [package\_name ]

$ cd /build/InfoFiles/

$ **pkgtrans** -s ../Released/ EQX-xx.xx.xx-yy-solaris.patch

$ cd /build/Released/

$ ls -lrt

equinox EQX-xx.xx.xx-yy-solaris.path

…

1. When it had already finished. You will get completely package.

# Summary

All of the scripts there are template. You can get template and edit something that you want according to [instruction](#_Instruction).