

OpenSolaris HotTopic seminar 2009.04



opensolaris[™]

新パッケージシステム「IPS」で広がる
OpenSolarisの世界

OpenSolaris Users Group
ジャストプレイヤー株式会社
瀧 康史 / TAKI, Yasushi

Agenda

ユーザのためのIPS

IPSについて
pkg コマンドの利用方法(逆引き的な利用方法)
各種レポジトリについて(主観)

システム管理者、開発者のためのIPS

pkg/serverの立ち上げ方



皆さんに質問！

どちらの頻度が高いですか？

A. GUI派

X-Window (UNIXやLinuxのデスクトップ) で、Xで書かれた**GUIツール**を利用して、サーバを設定、管理する (OverSSH X forwardingも含む) ことが多い。

B. CLI派

SSH経由 (WindowsのPuTTY、TeraTermなど) で、サーバにアクセスし、**CLIツール**を利用して、サーバを設定、管理する。

C. AもBも50:50.... (誤差率±5%未満)





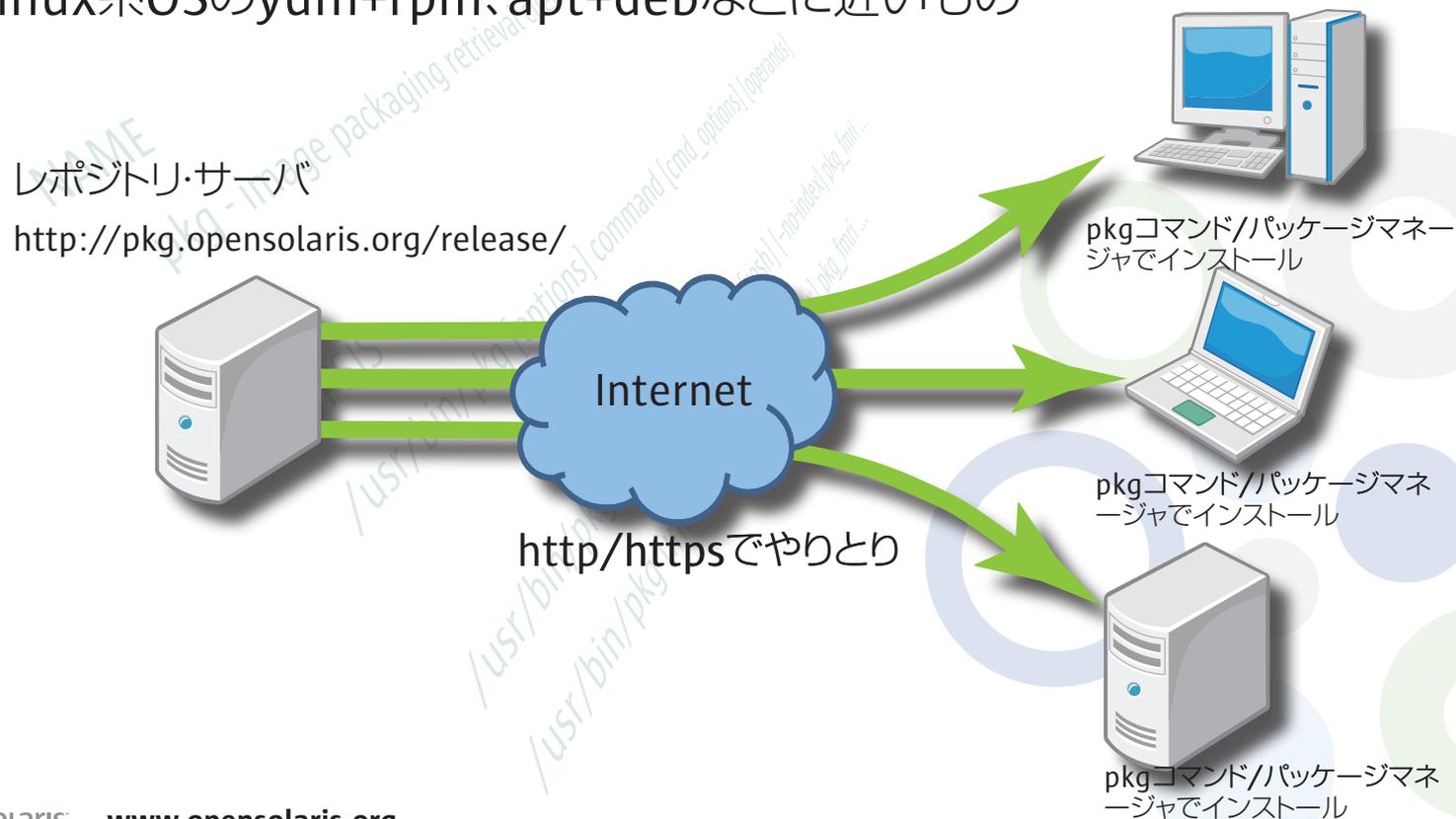
opensolaris™

ユーザのための IPS

IPS) Image Packaging System とは？

OpenSolarisで利用されているパッケージングシステム

- ソフトウェアをInternet越しに簡単に配信・インストールできる
- Solaris10のSVR4パッケージシステム(pkgadd等)を代替する
- Linux系OSのyum+rpm、apt+debなどに近いもの



IPS の特徴

pkgコマンド、パッケージマネージャー双方の特徴

- GUI版(パッケージマネージャー)とCLI版(pkgコマンド)の両方が用意されている。
- 依存関係などを自動的に解決するので、ソフトのインストールが簡単。
 - 依存するものを自動的にインストールする。
- ネットワークベースのシステムであり、基本、オフラインで利用しない。
 - http経由でアクセスを行う。
 - ProxyServerの利用が可能 (env http_proxy=http://.../ pkg ...)。
- image-updateなど、他のパッケージ管理ツールにはない、独自の機能がある。
- 証明書付きレポジトリとのアクセスが考えられている。
- pythonで書かれている。

パッケージマネージャー

- IPSのGUIパッケージ管理ツール。
- コマンド名は/bin/pkginstall
- 通常運用では、CLI版とほぼ同じ機能がある。

The screenshot shows the OpenSolaris GUI Package Manager interface. The window title is "ファイル(E) 編集(E) パッケージ(P) 設定(S) ヘルプ(H)". The menu bar includes "再読み込み", "すべて更新", "インストール/更新", and "削除". The repository is set to "opensolaris.org".

The main window displays a list of packages with columns for "名前" (Name), "状態" (Status), and "説明" (Description). The following table represents the data shown in the screenshot:

名前	状態	説明
SUNWaclg		Apache Common Logging
SUNWapch		Apache Web Server
SUNWapch22		Apache Web Server V2.2
SUNWapch22d		Apache Web Server V2.2 Documentation
SUNWapch22m-fcgid		FastCGI plugin for Apache Web Server V2.2
SUNWapch22m-jk		Tomcat Connector plugin for Apache Web Server V2.2
SUNWapch22m-php52		PHP Server 5.2.6 module for Apache Web Server V2.2
SUNWapch22m-security		Mod Security plugin for Apache Web Server V2.2
SUNWapchd		Apache Web Server Documentation
SUNWapr13	✓	Apache Portable Runtime (APR) 1.3 development headers
SUNWapr13doc		Apache Portable Runtime (APR) 1.3 documentation
SUNWapu13	✓	Apache Portable Runtime Utility (APR-util) 1.3 development headers

The "SUNWapch22" package is selected, and its details are shown below:

SUNWapch22
Apache Web Server V2.2

一般情報 | ファイルの詳細 | 依存関係 | 使用条件

最新の利用可能なパッケージの情報:

説明: Apache Web Server V2.2
名前: SUNWapch22
FMRI: pkg:/SUNWapch22@2.2.9.5.11-0.101:20081119T214340Z
バージョン: 2.2.9-0.101
パッケージ化されました: 2008-11-19 21:43:40

1669 パッケージがリストされました, 682 がインストールされました, 0 が選択されました, 0 が壊れています。

pkg コマンド

- IPSのCLIパッケージ管理ツール(今回の第1部の主役)
- 1つのコマンドで、ほぼ全ての作業が可能。
- 証明書入りのレポジトリ追加(extra)は、現時点ではpkgコマンドからしかできない。

```
ファイル(E) 編集(E) 表示(V) 端末(T) タブ(D) ヘルプ(H)
kohju@hadriel(7614)% pkg
使用方法:
  pkg [options] command [cmd_options] [operands]

基本的なサブコマンド:
  pkg install [-nvq] [--no-refresh] [--no-index] package...
  pkg uninstall [-fnrvq] [--no-index] package...
  pkg list [-ahsvf] [package...]
  pkg image-update [-fnvq] [--no-refresh] [--no-index]
  pkg refresh [--full] [authority ...]
  pkg version
  pkg help

高度なサブコマンド:
  pkg info [-lr] [--license] [pkg_fmri_pattern ...]
  pkg search [-lrI] [-s server] token
  pkg verify [-Hqv] [pkg_fmri_pattern ...]
  pkg fix [pkg_fmri_pattern ...]
  pkg contents [-Hmr] [-o attribute ...] [-s sort_key]
  [-t action_type ...] [pkg_fmri_pattern ...]
  pkg image-create [-ffPUz] [--force] [--full|--partial|--user] [--zone]
  [-k ssl_key] [-c ssl_cert] [--no-refresh] -a <prefix>=<url> dir

  pkg set-property propname propvalue
  pkg unset-property propname ...
  pkg property [-H] [propname ...]

  pkg set-authority [-P] [-k ssl_key] [-c ssl_cert] [--reset-uid]
  [-O origin_url] [-m mirror_to_add | --add-mirror=mirror_to_add]
  [-M mirror_to_remove | --remove-mirror=mirror_to_remove]
  [--no-refresh] authority
  pkg unset-authority authority ...
  pkg authority [-HP] [authority ...]
  pkg history [-HL]
  pkg purge-history
  pkg rebuild-index

オプション:
  -R dir
  -D/--debug name=value

環境:
  PKG_IMAGE
kohju@hadriel(7615)% █
```



インストール作業に伴うコマンド

ソフトウェアをインストールしてみましょう。

インストールにつかう主なコマンドは次の通りです。

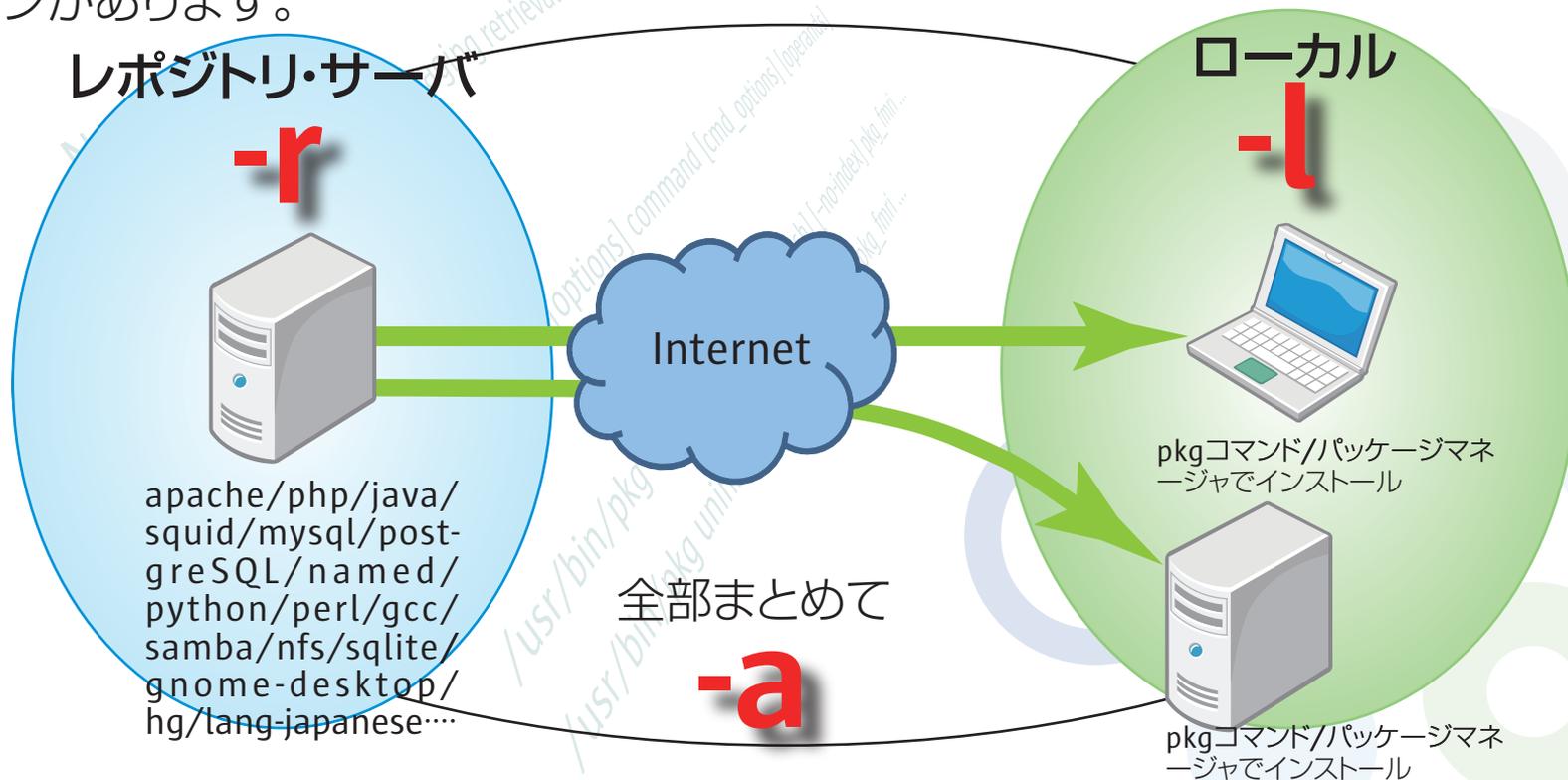
- pkg search パッケージの検索
- pkg info パッケージの情報取得
- pkg contents パッケージの構成ファイル一覧
- pkg list パッケージの一覧
- pkg install パッケージのインストール



レポジトリとインストール

IPSにおけるインストールとは…

- レポジトリにあるパッケージ(ソフトウェア、リソース)を、ローカルマシンにインストールすることです。
- pkgコマンドでは'-r'、'-l'(省略可能)、'-a'(一部のみ)という、直交的なオプションがあります。



Apache を探す

Apacheをインストールしてみましょう。

第1部では、様々なプログラムのインストールが分かるように、あえて検索方法から示すので、好きなプログラムに読み替えてみてください。

まずは検索してみます。レポジトリから検索するので'-r'をつけます。

```
% pkg search -r apache
```

INDEX	ACTION	VALUE	PACKAGE
basename	dir	usr/apache2/lib/perl/Apache	pkg:/SUNWapch2@2.2.3-0.75
basename	dir	usr/apache2/2.2/lib/perl/Apache	pkg:/SUNWapch22@2.2.8-0.86
basename	dir	usr/apache2/2.2/lib/perl/Apache	pkg:/SUNWapch22@2.2.8-0.99
basename	dir	usr/apache/perl5/5.8.4/i86pc-solaris/Apache	pkg:/SUNWapch@1.3.41-0.94
		～中略～	
basename	file	etc/init.d/apache	pkg:/SUNWapch@1.3.41-0.97
basename	file	etc/init.d/apache	pkg:/SUNWapch@1.3.41-0.96

ものすごい数が出てきます。wc -l で数えると「670行」。

これは、searchコマンドは、パッケージに含まれている構成要素(ファイル、ディレクトリ、メタ情報等々)を検索しているためです。

出力結果の見方

searchコマンドの出力結果はこのように読みます。

```
% pkg search -r apache
INDEX      ACTION    VALUE                                PACKAGE
basename  dir       usr/apache2/lib/perl/Apache        pkg:/SUNWapch2@2.2.3-0.75
```

INDEX ここでは、この値が**basename**の時の説明をします。

ACTION dir=ディレクトリ、file=ファイル、そのほか…

VALUE dir/fileの時、実際の構成要素。

PACKAGE パッケージのFMRI。

FMRI

FMRIとは……

パッケージの識別子。Fault Management Resource Identifierだそうです。SMFの「svc:/」と同じです。IPSは障害管理にはほとんど無関係なのですが……。

```
pkg:/SUNWapch2@2.2.3-0.75
```

pkg:/

FMRIを示す識別。

SUNWapch2

パッケージ名。

2.2.3

ソフトウェアのバージョン番号

0

わかりませんでした。すみません。

75

パッケージが対応するKernel番号。

uname -vで出てくるsnv_XXのXXの部分。

インストールの時には、対象パッケージのFMRIか、簡略したパッケージ名称が必要になります。したがって、FMRIかパッケージ名を見つけられないとなりません。

プログラム名で検索

Apacheのプログラム名はhttpdなので、httpdを探してみます。

```
% pkg search -r httpd
```

INDEX	ACTION	VALUE	PACKAGE
basename	file	usr/apache2/bin/httpd	pkg:/SUNWapch2@2.2.3-0.75
basename	file	usr/apache2/2.2/bin/amd64/httpd	pkg:/SUNWapch22@2.2.8-0.86
basename	file	usr/apache2/2.2/bin/httpd	pkg:/SUNWapch22@2.2.8-0.86
basename	file	usr/apache2/2.2/bin/amd64/httpd	pkg:/SUNWapch22@2.2.8-0.99
basename	file	usr/apache2/2.2/bin/httpd	pkg:/SUNWapch22@2.2.8-0.99
basename	file	usr/apache/bin/httpd	pkg:/SUNWapch@1.3.41-0.94
basename	file	usr/apache/bin/httpd	pkg:/SUNWapch@1.3.41-0.89
basename	file	usr/apache2/2.2/bin/amd64/httpd	pkg:/SUNWapch22@2.2.8-0.86
basename	file	usr/apache2/2.2/bin/httpd	pkg:/SUNWapch22@2.2.8-0.86
basename	file	usr/apache/bin/httpd	pkg:/SUNWapch@1.3.41-0.86
basename	file	usr/apache/bin/httpd	pkg:/SUNWapch@1.3.39-0.79
basename	file	usr/apache/bin/httpd	pkg:/SUNWapch@1.3.41-0.91
basename	file	usr/apache2/2.2/bin/amd64/httpd	pkg:/SUNWapch22@2.2.8-0.94

以後、割愛。wc -l で42行。lessで探せる範囲になりました。

でも、なんだかちょっと、パッケージ名が知りたいだけなのに、情報が冗長ですね。

出力結果を絞り込む

IPSレポジトリには過去のバージョンも含めて登録されているため、出力がとても煩雑になります。

自分が利用しているOpenSolarisのバージョン用「のみ」、検索結果として出力したい場合、次のようにします(残念ながら、オプションではできません)。例を2つ。

```
% pkg search -r httpd | grep `uname -v` | sed  
  's/[^0-9]//g' '$' | uniq
```

```
% pkg search -r httpd | grep '101$' | uniq  
basename file      usr/apache2/2.2/bin/httpd pkg:/SUNWapch22@2.2.9-0.101  
basename file      usr/apache/bin/httpd    pkg:/SUNWapch@1.3.41-0.101  
basename file      usr/apache2/2.2/bin/amd64/httpd pkg:/SUNWapch22@2.2.9-0.101  
basename file      usr/apache2/2.2/bin/httpd pkg:/SUNWapch22@2.2.9-0.101
```

このように、自分が今からインストールすべきApacheのパッケージ名が、SUNWapch22、SUNWapchのいずれかであることが分かります。

ここまでくると、「SUNWapch22」をインストールすると、Apache Ver2.2.9が、インストールされるのかなあ?とわかりますね。

パッケージの情報を得る

もう一步、確証を持ちたいなら、パッケージの情報と内容を調べます。

```
% pkg info -r SUNWapch22
```

```
名前: SUNWapch22
```

```
概要: Apache Web Server V2.2
```

```
カテゴリ: Web Services/Application and Web Servers
```

```
状態: インストール済みでない
```

```
権限: opensolaris.org
```

```
バージョン: 2.2.9
```

```
ビルドリリース: 5.11
```

```
分岐: 0.101
```

```
パッケージ化の日付: Wed Nov 19 21:43:40 2008
```

```
サイズ: 9.02 M バイト
```

```
FMRI: pkg:/SUNWapch22@2.2.9,5.11-0.101:20081119T214340Z
```

まず間違いなく、これで問題なさそうです。



パッケージの構成要素を得る

パッケージの中身は次のように調べます。どうやら間違いないようですね…。

```
% pkg contents -r SUNWapch22
```

```
PATH
```

```
etc
```

```
etc/apache2
```

```
etc/apache2/2.2
```

```
etc/apache2/2.2/conf.d
```

```
etc/apache2/2.2/conf.d/modules-32.load
```

```
etc/apache2/2.2/conf.d/modules-64.load
```

```
etc/apache2/2.2/envvars
```

```
etc/apache2/2.2/httpd.conf
```

```
etc/apache2/2.2/magic
```

```
etc/apache2/2.2/mime.types
```

```
etc/apache2/2.2/original
```

```
etc/apache2/2.2/original/httpd.conf
```

```
etc/apache2/2.2/samples-conf.d
```

```
etc/apache2/2.2/samples-conf.d/autoindex.conf
```

```
etc/apache2/2.2/samples-conf.d/dav.conf
```

```
etc/apache2/2.2/samples-conf.d/default.conf
```

```
etc/apache2/2.2/samples-conf.d/info.conf
```

```
etc/apache2/2.2/samples-conf.d/languages.conf
```

```
etc/apache2/2.2/samples-conf.d/manual.conf
```



一覧から探したい!

今までの例では、構成するファイルから検索をしましたが、パッケージ名から検索する方法もあります。

```
% pkg list -a | grep -i apache
```

勇み足……。これは、何も出てきません。pkg list -aでは、パッケージ名とバージョン、ステートだけで、説明が何も無いからです。

```
% pkg list -a
```

NAME (AUTHORITY)	VERSION	STATE	UFI
BRCMbnx	0.5.11-0.101	installed	----
Cheetah (contrib)	2.0.1-0.101	known	----
Cython (contrib)	0.9.8.1.1-0.101	known	----
DPhostserver (localhost)	0.1-0.101	known	----
FSWfontconfig-devel-docs	0.5.11-0.101	known	----
FSWxorg-client-docs	0.5.11-0.101	known	----
FSWxorg-client-programs	0.5.11-0.101	known	----
FSWxorg-clientlibs	0.5.11-0.101	known	----
FSWxorg-data	0.5.11-0.101	known	----
FSWxorg-devel-docs	0.5.11-0.101	known	----
FSWxorg-fonts	0.5.11-0.101	known	----
FSWxorg-fonts-core	0.5.11-0.101	installed	----

一覧に 1 行説明を取得する

そこで、1行説明が出力されるリストモードで検索することも出来ますが……

```
% pkg list -as | grep -i apache
SUNWaclg          Apache Common Logging
SUNWapch          Apache Web Server
SUNWapch22        Apache Web Server V2.2
SUNWapch22d       Apache Web Server V2.2 Documentation
SUNWapch22m-dtrace DTrace plugin for Apache Web Server V2.2
SUNWapch22m-fcgid FastCGI plugin for Apache Web Server V2.2
SUNWapch22m-jk    Tomcat Connector plugin for Apache Web Server V2.2
SUNWapch22m-php52 PHP Server 5.2.6 module for Apache Web Server V2.2
SUNWapch22m-security Mod Security plugin for Apache Web Server V2.2
    ～以後割愛～
```

実はこのコマンド、とっても遅くて数分ぐらい返ってきません。

```
% pkg list -as > pkg-list-as.txt
% cat pkg-list-as.txt | grep -i apache
```

こんなふうに、出力結果を、定期的にとどこかにとっておくと良いでしょう。

インストール前の最後の確認

パッケージ名がわかったらインストールをします。どんな依存関係があるのかわからないので、先にdry-runをしておくとい良いでしょう。

```
% pfexec pkg install -vn SUNWapch22
```

```
Creating Plan - 評価前:
```

```
UNEVALUATED:
```

```
+pkg:/SUNWapch22@2.2.9,5.11-0.101:20081119T214340Z
```

```
評価後:
```

```
None -> pkg:/SUNWapch22@2.2.9,5.11-0.101:20081119T214340Z
```

```
None -> pkg:/SUNWapr13@1.3.3,5.11-0.101:20081119T214506Z
```

```
None -> pkg:/SUNWapu13@1.3.4,5.11-0.101:20081119T214514Z
```

```
None -> pkg:/SUNWperl584core@5.8.4,5.11-0.101:20081119T215706Z
```

```
Actuators:
```

```
restart_fmri: svc:/system/manifest-import:default
```

```
None
```

-vを指定すると、依存関係で芽づる式にインストールするパッケージ名などをレポートします。-nはdry-run(評価のみで実際には実行しない)です。

pkg:/と、@以降のバージョン番号は省略してもかまいません。省略するとレポジトリから最新の物をインストールします。古いバージョンが必要になるときだけ、Version番号を指定しましょう。

いざ、インストール

実際にインストールする。

```
% pfexec pkg install -v SUNWapch22
```

```
Creating Plan / Before evaluation:
```

```
UNEVALUATED:
```

```
+pkg:/SUNWapch22@2.2.9,5.11-0.101:20081119T214340Z
```

```
After evaluation:
```

```
None -> pkg:/SUNWapch22@2.2.9,5.11-0.101:20081119T214340Z
```

```
None -> pkg:/SUNWapr13@1.3.3,5.11-0.101:20081119T214506Z
```

```
None -> pkg:/SUNWapu13@1.3.4,5.11-0.101:20081119T214514Z
```

```
None -> pkg:/SUNWperl584core@5.8.4,5.11-0.101:20081119T215706Z
```

```
Actuators:
```

```
restart_fmri: svc:/system/manifest-import:default
```

```
None
```

```
DOWNLOAD
```

```
Completed
```

```
PKGS
```

```
4/4
```

```
FILES
```

```
1342/1342
```

```
XFER (MB)
```

```
5.85/5.85
```

```
PHASE
```

```
Install Phase
```

```
PHASE
```

```
ACTIONS
```

```
1660/1660
```

```
ITEMS
```

以上で、インストールが完了します。

Apache の設定と起動は？

さて、pkg://SUNWapch22はどうやって起動すれば良いのでしょうか？

- pkg info SUNWapch22

設定情報に関する情報はありません。

- man SUNWapch22

manはありません。

そこで、インストールされたファイル一覧を見ます。

```
% pkg contents SUNWapch22
```

Apache の設定 (その 1)

1. pkg contents SUNWapch22から検索

```
% pkg contents SUNWapch22 | grep httpd.conf
```

この方法でapacheの設定ファイルが/etc/apache/2.2/conf/httpd.confにあることが分かります。

しかし、Apacheのconfigファイルが、httpd.confと知っていたから、この検索ができたわけで、*.cfg、*.xmlだったりすると、contentsの一覧をじっくり見なくてはなりません。

Apache の設定 (その 2)

2. svcprop apache22で検索

OpenSolarisのSMFでは、daemonの起動設定を、svcpropで参照可能なプロパティで変更することができます。これは、RHELなどで、/etc/sysconfig/httpdなどのファイルに、環境変数を書いて動作モードを変更することに似ています。

右は、svcpropの出力の一部です。daemonの起動スクリプトや、ものによっては設定ファイルの場所をここで変更できるものもあります。

```
% svcprop apache22 | grep exec
start/exec astring /lib/svc/method/http-apache22\ start
stop/exec astring /lib/svc/method/http-apache22\ stop
refresh/exec astring /lib/svc/method/http-apache22\ refresh
```

execのmethodはシェルスクリプトなので、あとは、/lib/svc/method/http-apache22を読むことで、何ができるのか理解を深めることができます。

```
httpd/enable_64bit boolean false
httpd/server_type astring prefork
httpd/stability astring Evolving
httpd/startup_options astring ""
httpd/value_authorization astring
solaris.smf.value.http/apache22
network/entities fmri svc:/milestone/
network:default
network/grouping astring require_all
network/restart_on astring error
network/type astring service
filesystem-local/entities fmri svc:/
system/filesystem/local:default
filesystem-local/grouping astring
require_all
```

Apache の起動方法 (その 1)

apacheはdaemonなので、普通、SMFのためのmanifestがあります(旧型の場合はinit.dを検索)。

```
% pkg contents SUNWapch22 | grep manifest
var/svc/manifest
var/svc/manifest/network
var/svc/manifest/network/http-apache22.xml
```

manifestは、ソフトウェアの実行環境や状況を記載したxmlファイルです。

manifestを見つけたら、svcs -aで、サービスの一覧をみて、探します。

大体、xmlファイルのPATH名で、FMRIが定義されています。

ここでは、http-apache22.xmlという名前だったので、apache22という名前で検索します。

```
% svcs -a | grep apache22
disabled          2:58:15 svc:/network/http:apache22
```

ありました。これをどうやって、動かしましょう?

Apache の起動方法 (その 2)

次は、`svc:/network/http:apache`の`svc`の諸情報を見ます。

```
% svcs -xv apache22
svc:/network/http:apache22 (Apache 2.2 HTTP server)
State: disabled since Fri Mar 27 02:58:15 2009
Reason: Disabled by an administrator.
See: http://sun.com/msg/SMF-8000-05
See: man -M /usr/apache2/2.2/man -s 8 httpd
See: http://httpd.apache.org
See: /var/svc/log/network-http:apache22.log
Impact: This service is not running.
```

あとは、「`svcadm enable apache22`」で起動可能です。もし、起動できなければ、再び「`svcs -xv apache22`」でログなどの情報を調べ直すことが可能です。

■`man`を上記に記述されてるとおりに入力して、つれない返事をもらった方へ

今まで記載した方法を応用すると、何のパッケージが足りないのかわかります。

```
% pkg search -r '/usr/apache2/2.2/man/*httpd*'
ヒント:最後の*がポイントです!
```

このようにして、記載されている情報から、`man`を得ることができます。

インストール系コマンドのまとめ

pkg search -r パス・ファイル名

レポジトリにあるファイルをさがす。'*'を使って検索範囲を広げることができる。

pkg info -r FMRI

パッケージの諸情報の取得ができる。

pkg contents -r FMRI

パッケージに含まれているファイルの一覧を取得することができる。

pkg list [-as]

パッケージの一覧がとれる。-aがないとインストール済みのもの、-aはall。このコマンドには-rがないが、-aはレポジトリよりも広い範囲を示す。-sをつけると1行説明がつく。

pkg install [-vn] FMRI

パッケージのインストール。-vを付けると依存してインストールするものが画面出力される。-nをつけると、実際には実行しない。

※たいていのコマンドは、-rでレポジトリを示し、つけないとインストール済みのものを指します。

SMFの参考(manで調べましょう)

svcadm サービス上げ下げ/svcs 状態確認/svcprop プロパティ参照/svccfg プロパティ変更

アップデート方法

パッケージ管理ツールの最大のメリットは、アップデートが簡単なことにつきます。

下記の方法で、インストールされているパッケージにアップデートがあるか調べることが出来ます。

```
% pkg list -u
```

このようなスクリプトを作っておき、1日に1度ぐらいcronで回しておけば、アップデート状況をすぐに見つけることができます。

```
#!/bin/sh
```

```
pkg refresh --full  
pkg list -u 2>&1 | grep -v 'no installed packages have available updates'
```

実際にパッケージをアップデートするには、`install`コマンド利用します。

```
pkg install -v パッケージ名
```

環境全体のアップデート

下記のようにすると、インストールされているパッケージ全体のアップデートを行います。

これは、CentOSなどで `yum update` を全体にかける感じで、直接ファイルの更新作業を行うのですが、問題があると復元には努力が必要です。

```
% pkg install -v entire
```

どのOSでも、アップデートを全体的に行うのは、いろいろ問題をはらみます。

アップデートに失敗すると、最悪、起動に失敗したり、いくつかのファイルが壊されたり、元の環境に戻ることもできず、一大決心が必要です。

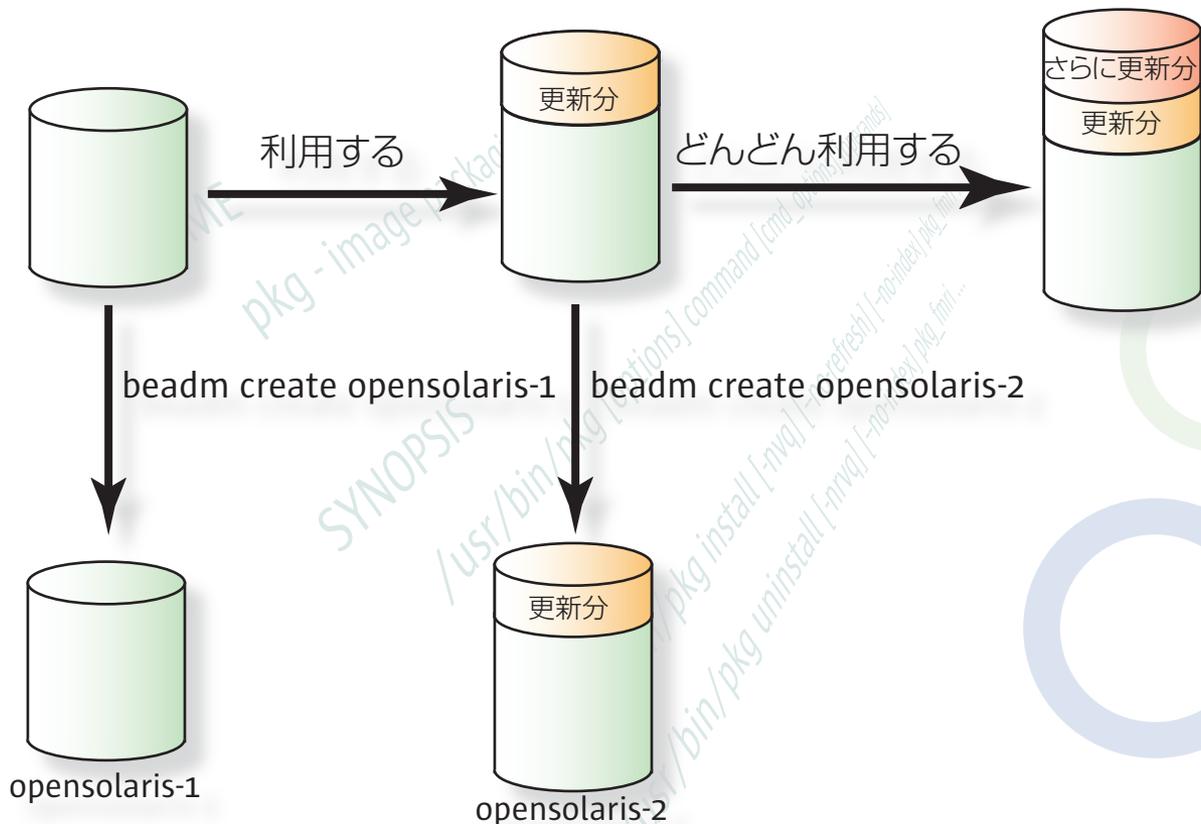
OpenSolarisではもっと安心な方法があります!



BE)boot environment

OpenSolarisには、ZFSのsnapshotとcloneを利用したBE)boot environmentという概念があります。

beadmを利用して、現状のROOTイメージのスナップショット(=ブート環境:BE)を取っておくことで、好きなタイミングのBEに、いつでも戻すことができます。



GRUBを利用することで、どのタイミングのBEからでも起動することも可能です。

beadm の利用の様子

BEの作成

```
beadm create opensoalris-2
```

BEの一覧

```
beadm list
```

ACTIVEのフラグで、現在のBE(N)と、次回ブートの時にどのBEが使われるか(R)がわかります。

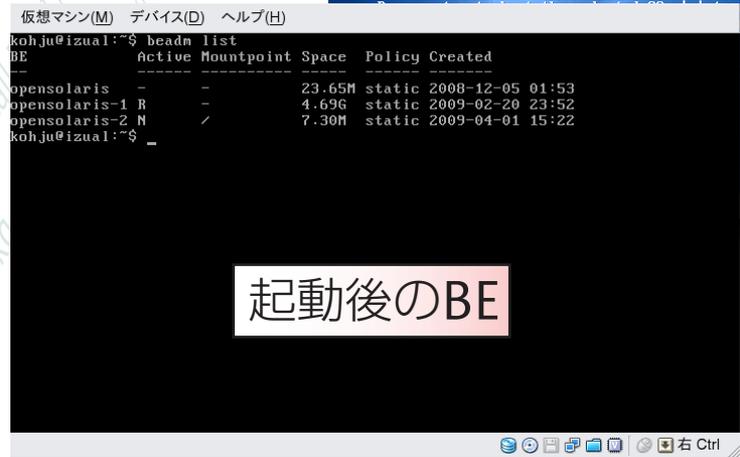
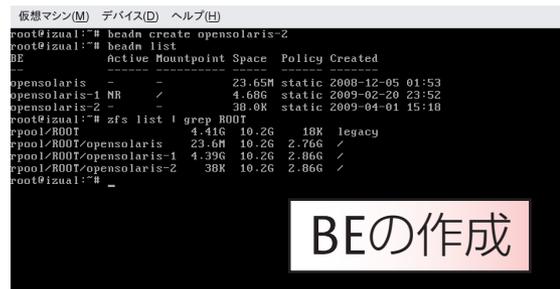
起動時のGRUBメニューでは、(R)で選択されたBEが選択されています。ここで別のBEを選択することも可能です。

右はopensolaris-2を選んで起動したあとの様子です。

デフォルトのBEの選択は、

```
beadm activate opensoalris-2
```

で、設定可能です。



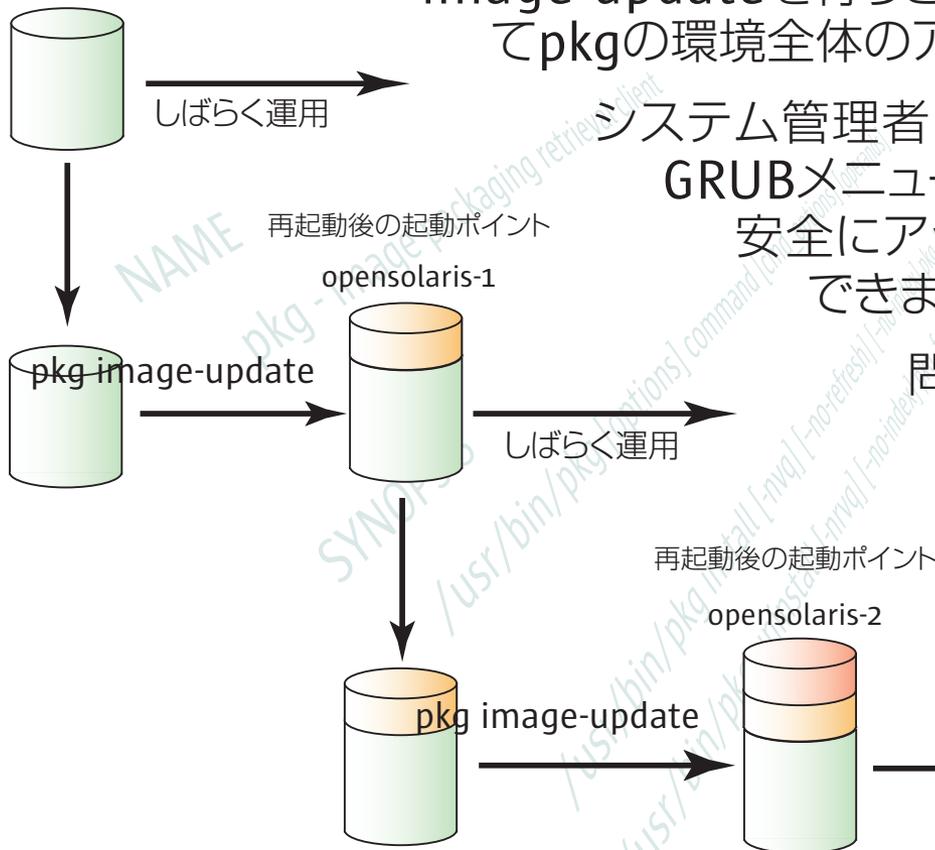
pkg image-update

pkg image-updateは、beadmを内部的に呼び出し、アップデート前の環境全体を確実に保存します。

image-updateを行うと、内部的に作成した新しいbeに対してpkgの環境全体のアップデートを行います。

システム管理者はアップデート後にリブートを行い、GRUBメニューから新しい環境を起動することで、安全にアップデートの後の環境を試すことができます。

問題がなければ使い続け、問題があればすぐに戻れるので、安心してアップデートができます。



複数のレポジトリを登録する

pkgコマンドでインストールできるソフトウェアは、レポジトリを追加することで、増やすことができます。標準では、[opensolaris.org](http://pkg.opensolaris.org)のみ登録されています。

```
% pkg authority
AUTHORITY                                URL
opensolaris.org (preferred)             http://pkg.opensolaris.org/release/
```

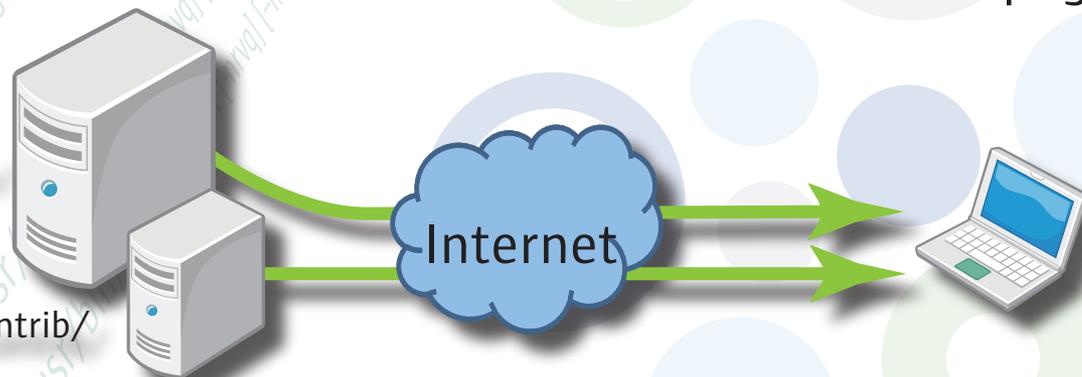
ために、**contrib**というレポジトリを追加してみます。

```
% pkg set-authority -0 http://pkg.opensolaris.org/contrib/ contrib
% pkg authority
AUTHORITY                                URL
contrib                                  http://pkg.opensolaris.org/contrib/
opensolaris.org (preferred)             http://pkg.opensolaris.org/release/
```

以後、追加したレポジトリにあるパッケージは、同じようなオペレーションで、**pkg install**可能となります。

<http://pkg.opensolaris.org/release/>

<http://pkg.opensolaris.org/contrib/>



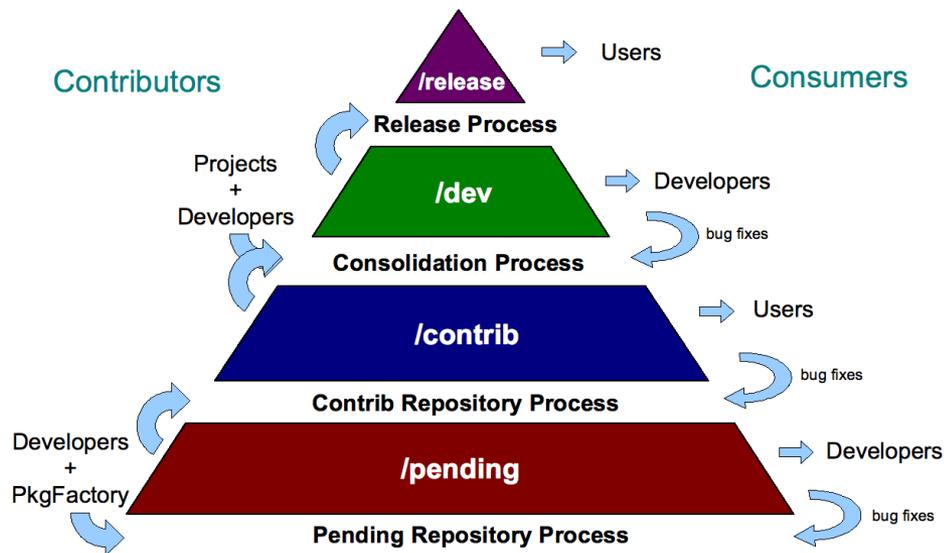
レポジトリのポリシーを意識する

レポジトリを追加するだけで、ソフトウェアのインストールが簡単に行えますが、無節操に追加してインストールすると、やっかいな問題がしばしば起きます。

それぞれのレポジトリのポリシーを意識しなくてはなりません。

pkg.opensolaris.orgで提供されているレポジトリには、次のものがあります。

OpenSolaris Package Repositories



release (OSのコア)

OSのcoreが提供されているリリースレポジトリ。現在、snv_101。

dev

OSのcoreが提供されている開発レポジトリ。現在、snv_110。

contrib

ある程度安定性がでてきたもの

pending

まずユーザから提供されたばかりのパッケージ

release or dev

release/dev/contrib/pending/そのほか、レポジトリはすべてauthorityに同じように登録されます。自分で追加するレポジトリは、あくまで拡張パッケージ的なものですが、releaseとdevに関しては、opensolarisのON(OS+Network)が含まれているため、排他的に利用します。

devリリースを追従する方法

```
% pkg set-authority -P -O http://pkg.opensolaris.org/dev/ dev  
% pkg image-update
```

release



or

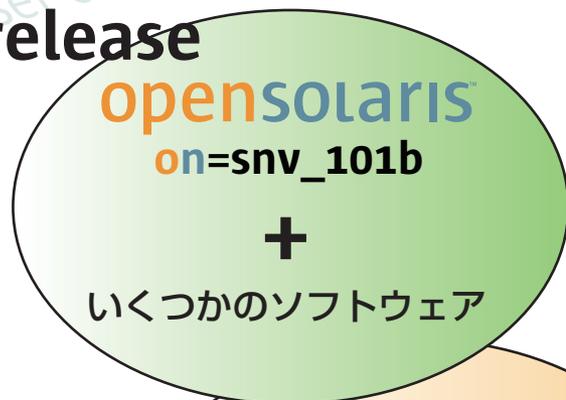
dev



+ contrib , +pending

contribとpendingは、release or devに追加するパッケージです。
ほとんどのレポジトリは、このような形でOSに追加する形をとります。

release



contrib

現時点では、まだ数は少ない。

ある程度安定したもの。

追加しておいてもほとんど問題はない。

+

pending

数がとても多い。

単品でつかう程度であれば、問題が起きたことはないが、依存関係まできちんとデバッグされているかは不明。

意識して利用の時だけset-authorityし、いらなくなったらunset-authorityすることをおすすめ。



dev

opensolaris.orgで配布されているものは、どちらもOSとおなじディレクトリPATHにインストールされます。



その他のレポジトリ

opensolaris.org以外で配布されているレポジトリもあります。

次ページからは、それぞれのレポジトリの私の「主観」を紹介します。

これらのレポジトリは、どれも拡張用のレポジトリです。

それぞれのポリシーのもとに作成されているので、ポリシーを把握した上で、インストールすることをおすすめします。



sunfreeware.com

- パッケージ内容

- オープンソースのパッケージ集。

- WEBサイト

<http://www.sunfreeware.com>

- パッケージ提供形式

- IPSとSVR4 (Solaris用)

- IPS追加方法

```
pkg set-authority -0 http://pkg.sunfreeware.com:9000/ sunfreeware.com
```

- インストール場所

/opt/sfw

- 私感

- 老舗中の老舗
- ディレクトリがSFEと、CompanionDISCと重なる。

blastwave.org

- パッケージ内容
 - オープンソースのパッケージ集。Solaris8からサポートされていて、パッケージ点数が多いのが特徴。
- WEBサイト
 - `http://www.blastwave.org/` `http://wiki.blastwave.org/`
- パッケージ提供形式
 - IPSとpkgutil+SVR4 (OpenSolaris、Solaris両用)
- IPS追加方法(ただし推奨しない)
 - `pkg set-authority -0 http://blastwave.network.com:10000/ blastwave.org`
- pkgutil追加方法
 - `pkg install -v SUNWwget SUNWpkgcmds`
`pkgadd -d http://dist.justplayer.com/csw/pkgutil_i386.pkg`
`/opt/csw/bin/pkgutil -a` (リスト一覧)
- インストール場所
 - `/opt/csw/`

Sun extra

- パッケージ内容
 - opensolaris 用のパッケージで、ライセンス上、再配布やコピーが不可能なサン固有もしくはサードパーティの内容を含むパッケージ。

- WEBサイト

<http://pkg.sun.com/>

- パッケージ提供形式

- IPS

- IPS追加方法

- レポジトリがアクセスキーと電子証明書でアクセス制御されているので、WEBから登録を行い、登録と参照の際に関連付ける必要があります。

```
pkg set-authority \  
-k /var/pkg/ssl/OpenSolaris_extras.key.pem \  
-c /var/pkg/ssl/OpenSolaris_extras.certificate.pem \  
-0 https://pkg.sun.com/opensolaris/extra/ extra
```

- 私感

- なにか良いものが配布される場所。



pkglabo.justplayer.com

- パッケージ内容

- オープンソースのサーバ向けパッケージ集。一部、当社（ジャストプレイヤー株式会社）からの配布物あり。
- 4/2にサービスインした当社サービスのOpenSolarisのレンタルサーバ（VPS）、**Phase2Server** (<http://www.justplayer.ne.jp/>) 向けに作成されたパッケージ集。基本的に外部からのアクセスも無償。
- OpenSolarisにあるものはなるべくそのまま使う。contribに依存するがpendingには依存しない。

- WEBサイト

<http://pkglabo.justplayer.com/>

- IPS追加方法

後日、pkglabo.justplayer.comと、www.justplayer.ne.jpにて公開。

- インストール場所

/opt/JPC

COUPON: IrfKxn55j



混ぜるな危険!

レポジトリがどのような運営方針によって、できているかを意識しないといけない。使う使わないだけでなく、使っても距離感を意識する。

- 運営方針

- 恒常性

- インストールされるPATH

- アップデートサイクル

発生する問題

- 依存関係が絡むと大変

- コンパイル時のライブラリ、実行時のライブラリ。

- soライブラリの順番で動かない(LD_LIBRARY_PATH等)

- プログラムが別のプログラムをどれを動かしているのか、どのPATHのライブラリを使うのか、PATHの意識が必要。

- 解決方法のヒント

- PATH/LD_LIBRARY_PATH/truss -t open,read等。



依存関係を知る

パッケージの依存関係を知りたいときにはcontentsコマンドを利用します。

実は、インストールを容易にするために、依存関係しか記述がないパッケージも存在します。

```
% pfexec pkg contents -rm amp | grep ^depend
depend fmri=SUNWapch22m-security@2.1.5-0.101 type=require
depend fmri=SUNWmysql5@5.0.67-0.101 type=require
depend fmri=SUNWapch22m-dtrace@0.3.1-0.101 type=require
depend fmri=SUNWphp52@5.2.6-0.101 type=require
depend fmri=SUNWapch22@2.2.9-0.101 type=require
depend fmri=SUNWphp52-mysql@5.2.6-0.101 type=require
depend fmri=SUNWapch22m-fcgid@2.2-0.101 type=require
depend fmri=SUNWapch22m-php52@5.2.6-0.101 type=require
```

上記のampはApache/MySQL/PHPのパッケージです。それ以外にも、gcc-dev、sunstudio、lang-support-*、entire等々、様々なものが存在します。

コマンドの対比

	IPS	pkgutil+pkgadd	pkg-get+pkgadd	yum+rpm	apt+dpkg
ファイル検索(インストール済/レポジトリ)	pkg search/pkg search -r	pkgchk -lp/なし	pkgchk -lp/なし	rpm -qf / yum provides	dpkg -S/apt-cache search
リスト取得(インストール済/レポジトリ)	pkg list/pkg list -a(リモートではなく全体)	pkginfo(pkgutil -c)/pkgutil -a	pkginfo(pkg-get -l) /pkg-get -D	rpm -qa/yum list	dpkg -l/apt-cache pkgnames
メタ情報取得(インストール済/レポジトリ)	pkg info/pkg info -r	pkginfo -l/	pkginfo -l/	rpm -qi/yum info	dpkg -s / apt-cache showpkg
内容の一覧(インストール済/レポジトリ)	pkg contents/pkg contents -r	pkgchk -l	pkgchk -l/	rpm -ql/?	dpkg -L/?
ベリファイ	pkg verify	pkgchk -v	pkgchk -v	rpm -V	debsums
インストール	pkg install	pkgutil -i (pkgadd)	pkg-get -i (pkgadd)	yum install	apt-get install
アップデート	pkg install	pkgutil -u	pkg-get -u	yum update/install	apt-get install
アンインストール	pkg uninstall	pkgutil -r (pkgrm)	pkg-get -r	yum uninstall	apt-get remove
環境全体のアップデート	pkg image-update / pkg install entire	pkgutil -U && pkgutil -u	pkg-get -U && pkg-get -u	yum upgrade	apt-get dist-upgrade

- IPSにはメタ情報検索系がなさそう。
- レポジトリとインストール済みのローカルで、コマンド体型が同じである。
- 他のパッケージシステムに比べて、メタ情報はキャッシュをあまりしない。

~XEM~

User Commands

NAME

pkg - image packaging retrieval client

SYNOPSIS

`/usr/bin/pkg` [options] command [cmd_options] [operands]
`/usr/bin/pkg install` [-nvq] [-no-refresh] [-no-index] pkg_fmri...
`/usr/bin/pkg uninstall` [-nvq] [-no-index] pkg_fmri...



User Commands

pkg(4)

NAME

pkg - image packaging retrieval client

SYNOPSIS

`/usr/bin/pkg [options] command [cmd_options] [operands]`
`/usr/bin/pkg install [-nvq] [-no-refresh] [-no-index] pkg_fmri...`
`/usr/bin/pkg uninstall [-nvq] [-no-index] pkg_fmri...`





opensolaris™

システム管理者・開発者のための IPS

configure && make よりも . . .

なぜIPSを立てるのか? pkg(4)

- ソフトウェアの配信用としてのIPS
- 環境準備を簡単にするためのIPS
- 独自レポジトリの作成

IPSには従来のSVRのように.pkg形式というローカルに保存出来る形式がありません。SVR4のPKGは非常にシンプルな構成であったので、configure && make && make install するよりも、SVR4のパッケージを作り、インストールした方が、後でアンインストールが容易です (post scriptのぞく...)

環境の準備

ビルド用、環境テスト用、レポジトリ用と3つの役割をもつサーバが必要になる。

- ビルド用
 - ビルドに必要なパッケージを入れる。
- 環境テスト用
 - 依存関係がちゃんと取れるかを確認用。
 - ほとんど何も無い状態でsnapshotをとっておき、何度でも構成できるようにする。
- IPSサーバ
 - pkg.depodを起動するもの

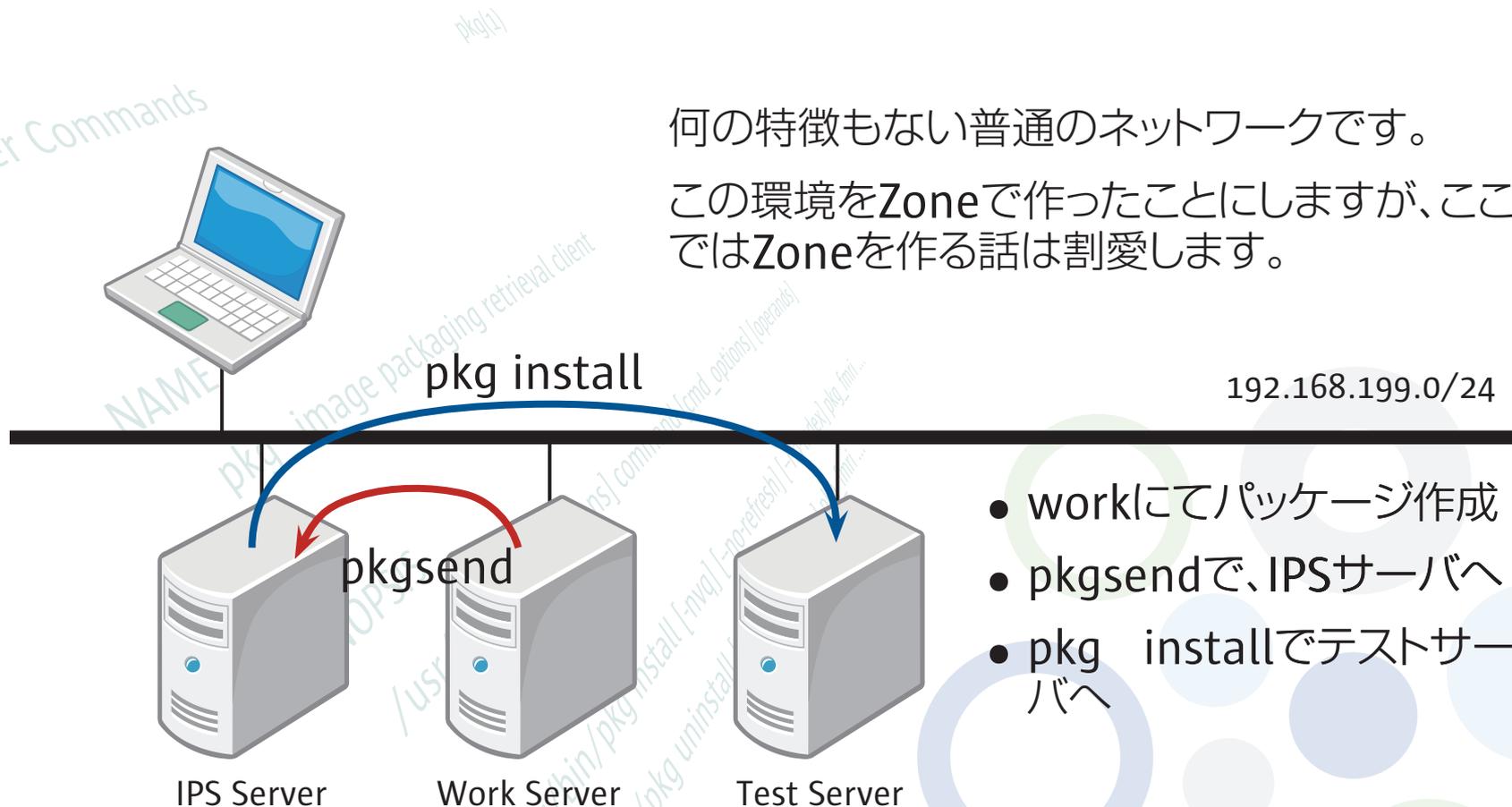
VirtualBoxで何台も作っても良いが、Kernelに関係するパッケージをIPSに入れない限り、Zoneでもよい。

Zoneはリソースもあるし、snapshotもとれるので便利でしょう。

パッケージビルドファーム #1

何の特徴もない普通のネットワークです。

この環境をZoneで作ったことにしますが、ここではZoneを作る話は割愛します。



IPS サーバの設定 #1

pkg/serverは、SUNWipkgに含まれているので、pkgコマンドとともにインストールされています。

右は、起動の様子とログの状態です。

デフォルトでは、

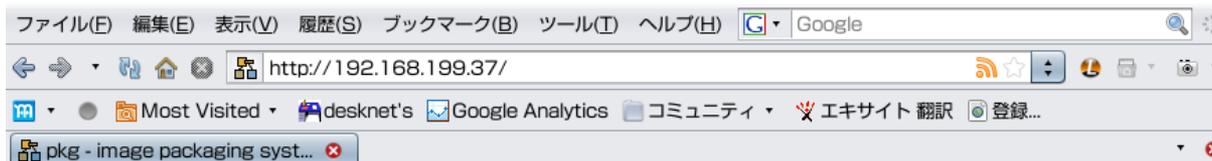
- ポート80でListen
- 読み書き両用(pkgsend可能)

で、レポジトリが立ち上がっています。この状態は外からなんでもできるのでご注意。

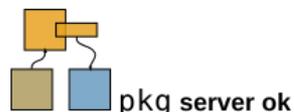
```
root@test-ips:~# svcs -xv pkg/server
svc:/application/pkg/server:default      (image      packaging
repository)
State: disabled since Thu Apr 02 16:04:43 2009
Reason: Disabled by an administrator.
See: http://sun.com/msg/SMF-8000-05
Impact: This service is not running.
root@test-ips:~# svcadm enable pkg/server
root@test-ips:~# svcs -xv pkg/server
svc:/application/pkg/server:default      (image      packaging
repository)
State: online since Thu Apr 02 16:05:32 2009
See: /var/svc/log/application-pkg-server:default.log
Impact: None.
root@test-ips:~# cat /var/svc/log/application-pkg-server:
default.log
[ Apr 2 16:05:32 Enabled. ]
[ Apr 2 16:05:32 Executing start method ("/lib/svc/method/svc-
pkg-depot start"). ]
ppriv -s A=basic,-file_link_any,-proc_info,-proc_session,net_
privaddr -e /usr/lib/pkg.depotd -d /var/pkg/repo -p 80 -s
10 -t 60 --content-root=/usr/share/lib/pkg --log-access=none
--log-errors=stderr
[02/Apr/2009:16:05:32] INDEX Search Available
[02/Apr/2009:16:05:32] ENGINE Listening for SIGHUP.
[02/Apr/2009:16:05:32] ENGINE Listening for SIGTERM.
[02/Apr/2009:16:05:32] ENGINE Listening for SIGUSR1.
[02/Apr/2009:16:05:32] ENGINE Bus STARTING
[02/Apr/2009:16:05:32] ENGINE Started monitor thread '_
TimeoutMonitor'.
[02/Apr/2009:16:05:33] ENGINE Serving on 0.0.0.0:80
[02/Apr/2009:16:05:33] ENGINE Bus STARTED
```



立ち上がった IPS の確認



ブラウザでアクセスをすれば、立ち上がった、状態の確認ができます。



Statistics

Number of packages: 0
Number of in-flight transactions: 0
Number of catalogs served: 0
Number of manifests served: 0
Number of files served: 0
Number of flists requested: 0
Number of files served by flist: 0
Number of packages renamed: 0



Last Updated: None

Catalog

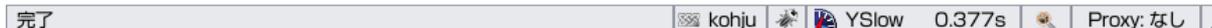


現時点では、なにも立ち上がっていないので、なにもありません。

確認の様子

```
# netstat -an | grep LISTEN | grep *.80
*.80          *.*
49152        0 LISTEN
```

※Listenプロセスを知りたい場合はPIDをしらべてpfilesする



Work 環境を設定する

いくつかのパッケージインストールをして、開発環境を整えます(例)

便利系

```
pkg install -v SUNWpkgcmds SUNWcar SUNWbeadm
SUNWipc SUNWicu SUNWlinks SUNWrcmds
SUNWgnu-coreutils SUNWmercurial SUNWman
SUNWdiffstat SUNWlogrotate SUNWncft
SUNWrpm SUNWlocate SUNWtexi SUNWp7zip
SUNWunzip SUNWzip SUNWgtar SUNWless
SUNWggrp SUNWwget SUNWw3m SUNWrsync
SUNWscp SUNWscreen SUNWtnetc SUNWbind
```

D-TRACE

```
pkg install -v SUNWDTraceToolkit
SUNWdtrp
```

UNICODE、日本語処理/エディタ系

```
pkg install -v SUNWlang-common SUNWlang-
ja SUNWlang-ja-extra SUNWiconv-unicode
SUNWiconv-extra SUNWuiu8 SUNWuium
SUNWman-l10n-ja SUNWgnu-emacs SUNWgnu-
emacs-el SUNWgnu-emacs-nox
```

開発ツール系

```
pkg install -v sunstudio SUNWbtool SUNWaconf
SUNWgcc SUNWgmake SUNWbison SUNWcvcs
```

```
SUNWsvn SUNWflexlex SUNWflexruntime
SUNWgd2 SUNWgdb SUNWgnu-automake-110
SUNWgpch SUNWgroff SUNWgsed SUNWlibtool
SUNWjdoc SUNWsolaris-devel-docs
SUNWgawk SUNWgnu-diffutils ss-dev
```

ライブラリ系

```
pkg install -v SUNWlibexif SUNWlibcrypt
SUNWlibgpg-error SUNWlibpopt
SUNWlibtasn1 SUNWloc SUNWlxml SUNWgcmn
SUNWglib SUNWcairomm SUNWgnu-gettext
SUNWgnu-gperf SUNWgnu-idn SUNWgnu-
libiconv SUNWgnu-which SUNWgnu-mc
SUNWlibxmlpp SUNWlibrsync SUNWlibpopt
SUNWlibrsvg SUNWltdl SUNWpcre SUNWsigcpp
SUNWgnome-common-devel SUNWperl-xml-
parser SUNWgnome-xml-root SUNWgnome-
xml SUNWgnome-xml-share
```

開発言語系

```
pkg install -v SUNWruby18 SUNWperl584usr
SUNWperl584man SUNWperl-xml-parser
SUNWpmdbi SUNWPython25 SUNWIPython
SUNWPython-extra
```



依存のみのパッケージを作る

下記のようなMANIFESTファイル「envstandard.manifest」を作り、パッケージをIPSに登録します。

これをいちいち作るのは面倒なので、支援ツールを書きました。

http://kohju.justplayer.com/Tips_Solaris_IPS_dependmanifest.html

次のように実行し、赤いところは後で手で自分で足しました。

```
echo SUNWpkgcmds .... SUNWrcmds | xargs -n1 | work/DimensionPlus/mk-depend.pl  
>envstandard.manifest
```

```
# JUSTPLAYER Phase2Server standard-set.  
#  
depend fmri=SUNWTraceToolkit@0.5.11,5.11-0.101 type=require  
depend fmri=SUNWGlib@1.2.10,5.11-0.101 type=require  
  
～割愛～  
  
depend fmri=SUNWw3m@0.5.11,5.11-0.101 type=require  
depend fmri=SUNWwget@1.10.2,5.11-0.101 type=require  
depend fmri=SUNWzip@2.32,5.11-0.101 type=require  
depend fmri=sunstudio@0.2008.11,5.11-0.86 type=require  
set name=description value="JUSTPLAYER Phase2Server standard-set."
```

パッケージをレポジトリに登録

次のようにして登録を行います。

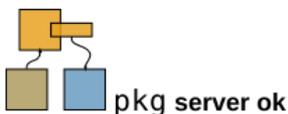
```
% pkgsend -s http://192.168.199.37/ open envstandard@1.0,5.11-101
export                               PKG_TRANS_ID=1238697731_pkg%3A%2Fenvstandard%401.0%2C5.11-
101%3A20090402T184211Z
%      export                         PKG_TRANS_ID=1238697731_pkg%3A%2Fenvstandard%401.0%2C5.11-
101%3A20090402T184211Z
% pkgsend -s http://192.168.199.37/ include envstandard.manifest
% pkgsend -s http://192.168.199.37/ close
PUBLISHED
pkg:/envstandard@1.0,5.11-101:20090402T184211Z
```

IPSは、パッケージファイルという実態が存在しません。

パッケージは、SVR4のパッケージの作り方とちがい、直接レポジトリサーバにコミットするような感じです。

一連の操作は、pkgsendを用い、open～closeの間で行われます。

登録状況の確認



Statistics

Number of packages: 1
Number of in-flight transactions: 0

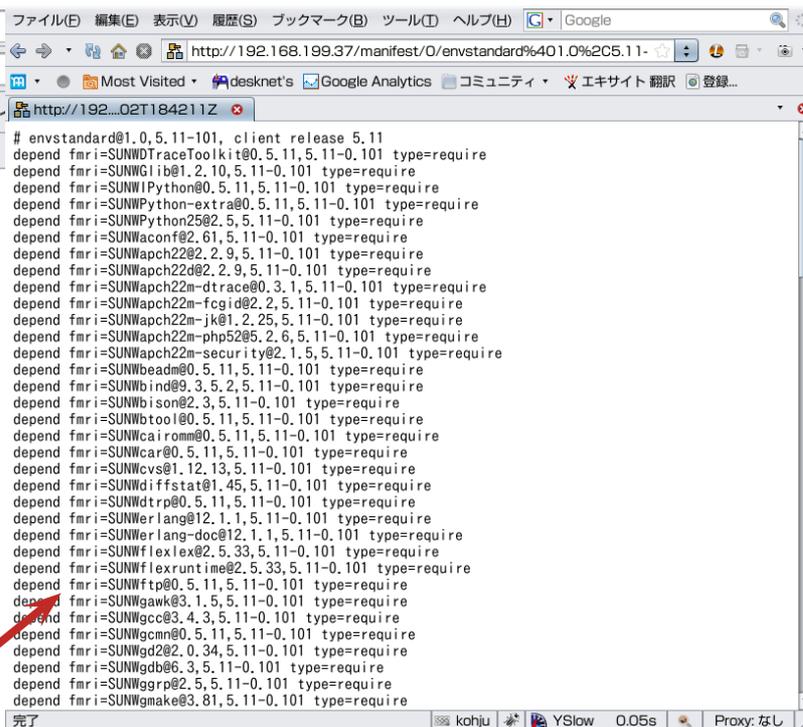
Number of catalogs served: 0
Number of manifests served: 0
Number of files served: 0
Number of flists requested: 0
Number of files served by flist: 0
Number of packages renamed: 0



Last Updated: 2009-04-03 03:42:39.420931

Catalog

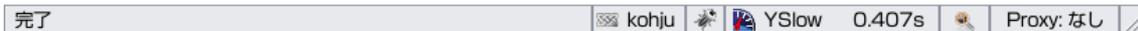
FMRI	Info	Manifest
pkg:envstandard@1.0,5.11-101:20090402T184211Z	Info	Manifest



2. MANIFESTをクリック

1. 登録されてる!

IPSに登録されました。



テスト環境インストール

レポジトリの登録

```
pkg set-authority -0 http://192.168.199.37/ test-ips
```

パッケージ一覧

```
# pkg list -a | grep test-ips
```

```
envstandard (test-ips) 1.0-0.101 known ----
```

インストール

```
# pkg install -vn envstandard
```

```
Creating Plan / Before evaluation:
```

```
UNEVALUATED:
```

```
+pkg://test-ips/envstandard@1.0,5.11-101:20090402T184211Z
```

```
After evaluation:
```

```
None -> pkg://test-ips/envstandard@1.0,5.11-101:20090402T184211Z
```

```
None -> pkg:/SUNWDTTraceToolkit@0.5.11,5.11-0.101:20081119T213917Z
```

～この間多量に割愛～

```
None -> pkg:/sunstudio@0.2008.11,5.11-0.86:20081113T205836Z
```

```
Actuators:
```

```
restart_fmri: svc:/system/manifest-import:default
```

```
restart_fmri: svc:/application/desktop-cache/input-method-cache:default
```

依存だけのパッケージについて

- いつも、同じようなソフトをインストールするときに、とても役立つ。
- パックを作っておけば、複数のパターンにも対応
 - 開発セットパック
 - 言語集パック
 - WEBサーバ用パック
 - DBMSパック
 - 等々...
- opensolarisのレポジトリにも、同じようなものがあります。
 - amp Apache+MySQL+PHP
 - ss-dev SunStudio系開発環境
 - gcc-dev gcc系開発環境
 - storage-nas NAS作成用Package集
 - 等々



ファイル入りパッケージ

次の支援ツールを使うことで、MANIFESTの基盤を作ることができます。

<http://opensolaris.org/os/community/sw-porters/contributing/indiana-ips/pkgflistgen.pl>

```
% ./pkgflistgen.pl root/
```

```
#
# directories
#
dir mode=0555 owner=root group=bin path=
dir mode=0555 owner=root group=bin path=/opt
dir mode=0555 owner=root group=bin path=/opt/JPC
dir mode=0555 owner=root group=bin path=/opt/JPC/bin
dir mode=0555 owner=root group=bin path=/opt/JPC/share
dir mode=0555 owner=root group=bin path=/opt/JPC/share/man
dir mode=0555 owner=root group=bin path=/opt/JPC/share/man/man1
#
# files
#
file root/opt/JPC/bin/tree mode=0555 owner=root group=bin path=/opt/JPC/bin/tree
file root/opt/JPC/share/man/man1/tree.1 mode=0555 owner=root group=bin path=/opt/
  JPC/share/man/man1/tree.1
#
# symlinks
#
```

```
% /opt/JPC/bin/tree root/
root/
|-- opt
    |-- JPC
        |-- bin
            |-- tree
        |-- share
            |-- man
                |-- man1
                    |-- tree.1

6 directories, 2 files
```

メタ情報を追記

前ページのMANIFESTには、メタ情報がたりないので追記します。

参考) <http://opensolaris.org/os/community/sw-porters/contributing/pkgcomp/>

- pkg.name パッケージを説明する短い名前
- pkg.description パッケージの1行説明文
- description pkg.descriptionと同じ
- pkg.detailed_url 関連URL(オプション)
- info.maintainer IPSパッケージのメンテナのE-MAIL情報
- info.upstream ソフトウェアのプロジェクトリーダーのE-MAIL情報
- info.upstream_url 配布元のURL。
- info.source_url ソースコードの配布URI(ダウンロードしたtarballのURL)
- info.repository_url ソースコードのレポジトリURL(オプション)
- info.classification クラス情報(<http://opensolaris.org/os/community/sw-porters/contributing/ipsclass/>)
- license ライセンス情報(ファイルで指定)

実際に作成したメタ情報

```
set name=pkg.name value="tree"
set name=pkg.description value="tree - A utility which displays a tree view of the contents
of directories."
set name=description value="tree - A utility which displays a tree view of the contents
of directories."
set name=pkg.detailed_url value=http://pkglabo.justplayer.com/
set name=info.maintainer value="TAKI,Yasushi <taki@justplayer.com>"
set name=info.upstream value="TAKI,Yasushi <taki@justplayer.com>"
set name=info.upstream_url value=http://pkglabo.justplayer.com
set name=info.source_url value=ftp://mama.indstate.edu/linux/tree/tree-1.5.2.2.tgz
set name=info.repository_url value=http://mama.indstate.edu/users/ice/tree/
set name=info.classification value="org.opensolaris.category.2008:Applications/System Utilities"
license tree-1.5.2.2/LICENSE license="GPLv2"
```

参考)SVR4 PKGのpkginfo

```
PKG=JPCtree
NAME=tree - A utility which displays a tree view of the contents of directories.
ARCH=i386
VERSION=1.5.2.2,REV=
CATEGORY=application
VENDOR=http://mama.indstate.edu/users/ice/tree/ packaged for pkglabo, JUSTPLAYER, co.,ltd. by
Taki.
HOTLINE=http://pkglabo.justplayer.com/
EMAIL=pkgadmin@justplayer.com
BASEDIR=/
```

パッケージの登録

できたMANIFESTを元に、登録をおこないます (WORK)

```
% eval `pkgsend -s http://192.168.199.37/ open JPCtree@1.5.2.2,5.11-0.101`  
% pkgsend -s http://192.168.199.37/ include JPCtree.manifest  
% pkgsend -s http://192.168.199.37/ close
```

あっさり…。

NAME

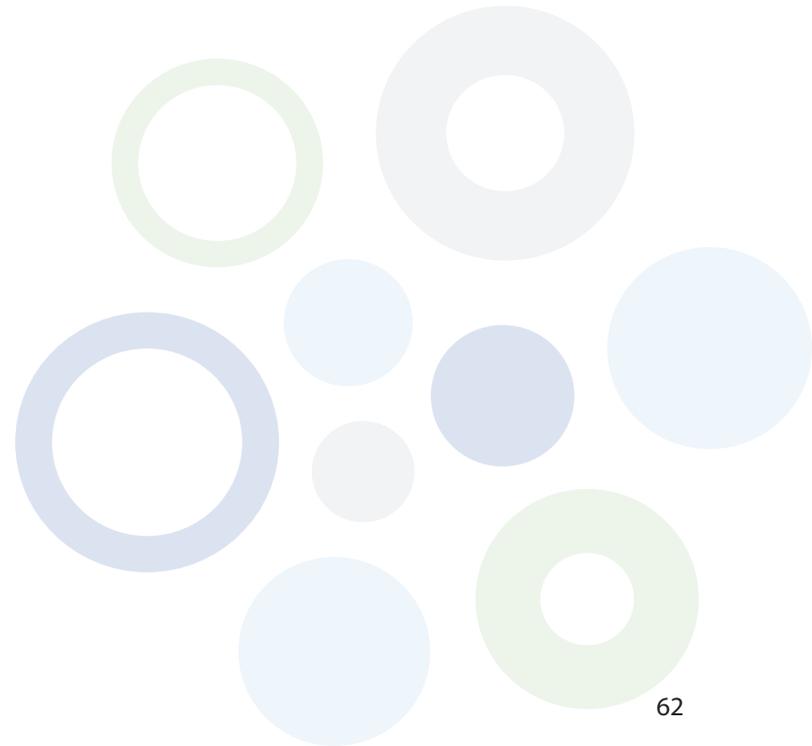
pkg - image packaging retrieval client

SYNOPSIS

/usr/bin/pkg [options] command [cmd_options] [operands]

/usr/bin/pkg install [-nvq] [-no-refresh] [-no-index] pkg_fmri...

/usr/bin/pkg uninstall [-nvq] [-no-index] pkg_fmri...



パッケージの確認

```
# pkg contents -r JPCtree
```

```
PATH
```

```
opt
```

```
opt/JPC
```

```
opt/JPC/bin
```

```
opt/JPC/bin/tree
```

```
opt/JPC/share
```

```
opt/JPC/share/man
```

```
opt/JPC/share/man/man1
```

```
opt/JPC/share/man/man1/tree.1
```

```
# pkg info -r JPCtree
```

```
Name: JPCtree
```

```
Summary: tree - A utility which displays a tree view of the contents of directories.
```

```
Category: Applications/System Utilities (org.opensolaris.category.2008)
```

```
State: Installed
```

```
Authority: test-ips
```

```
Version: 1.5.2.2
```

```
Build Release: 5.11
```

```
Branch: 0.101
```

```
Packaging Date: Thu Apr  2 20:17:31 2009
```

```
Size: 102.51 kB
```

```
FMRI: pkg://test-ips/JPCtree@1.5.2.2,5.11-0.101:20090402T201731Z
```

テストサーバで行う!



インストール

```
# pkg install -v JPCtree pkg(4)
```

```
Creating Plan - Before evaluation:
```

```
UNEVALUATED:
```

```
+pkg://test-ips/JPCtree@1.5.2.2,5.11-0.101:20090402T201731Z
```

```
After evaluation:
```

```
None -> pkg://test-ips/JPCtree@1.5.2.2,5.11-0.101:20090402T201731Z
```

```
Actuators:
```

```
None
```

```
PHASE
```

```
Indexing Packages
```

```
DOWNLOAD
```

```
Completed
```

```
ITEMS
```

```
96/96
```

```
PKGS
```

```
1/1
```

```
FILES
```

```
5/5
```

```
XFER (MB)
```

```
0.03/0.03
```

```
PHASE
```

```
Install Phase
```

```
Reading Existing Index
```

```
Indexing Packages
```

```
ACTIONS
```

```
22/22
```

```
9/9
```

```
1/1
```



SVR4 のパッケージからのコンバート

すでにSVR4のパッケージができていたりすると、下記のように簡単に、コンバートができる。

```
# eval `pkgsend -s http://192.168.199.37/ open JPClv@4.5.1,5.11-0.101`  
# pkgsend -s http://192.168.199.37/ import JPClv.pkg  
# pkgsend -s http://192.168.199.37/ add set name=description value="lv - a Powerful  
  Multilingual File Viewer / Grep"  
# pkgsend -s http://192.168.199.37/ close
```

SVR4からのコンバートは、あまり完璧ではなく、情報が微妙に足りないので、addコマンドを利用して先に記したMANIFESTをコピーしなくてはならない。

SVR4パッケージ作成時のpkginfoファイルから変換するソフトはかけそう。

パッケージ作業を体系化する

ソース・レポジトリの作成

tar ballのありか、自動的なダウンロード、パッチ、メタ情報等々を、リビジョン管理システム (cvs、svn、hg) などの管理下に起きたい。

これらを多人数で共有したい。

後続くものにHackを楽にさせたい (どういう環境でBuildしたのか分からないものは、拡張できない)

ビルド環境の統一化

簡単にビルドできるよ。ただし、僕の環境ではね・・・では、困る。

すべてのコミッターが統一した環境下でビルドができる必要がある。

専用ビルドマシンが欲しい

IPSに登録するビルドマシンは1つであることが望ましい。

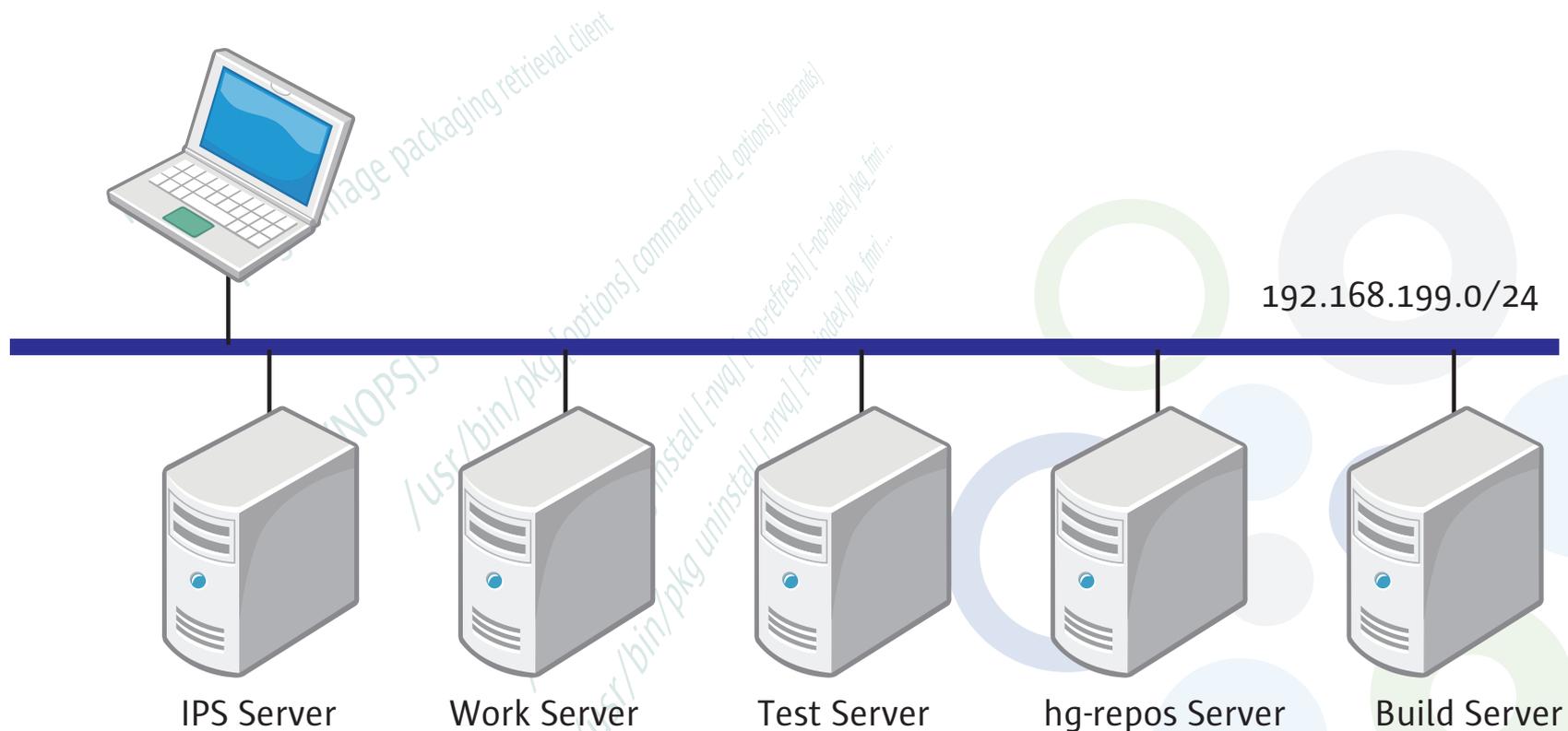
とある担当者がビルド環境に知らずに手を入れていると困るため。

ビルドファームの例

たとえばこのような形のマシンが必要になります。

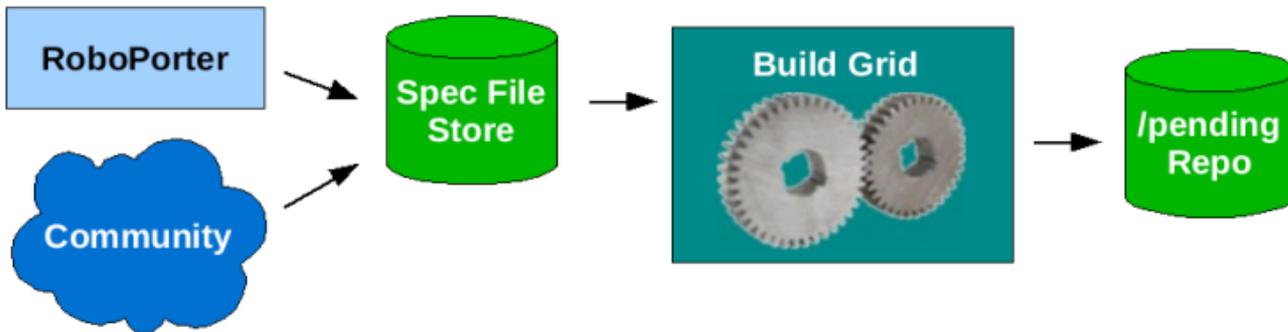
Work Server、Test Serverはコミッタの数だけあると便利です。

zone+zfsってすてきですね。

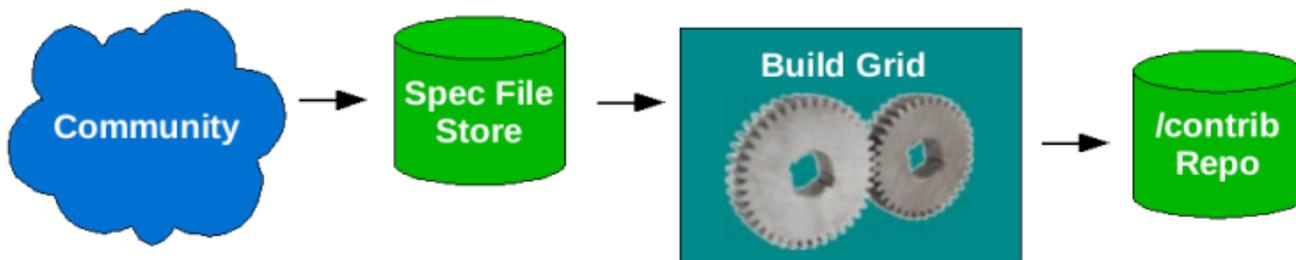


統一したビルドファームを作る

OpenSolaris Pending Repo Build System

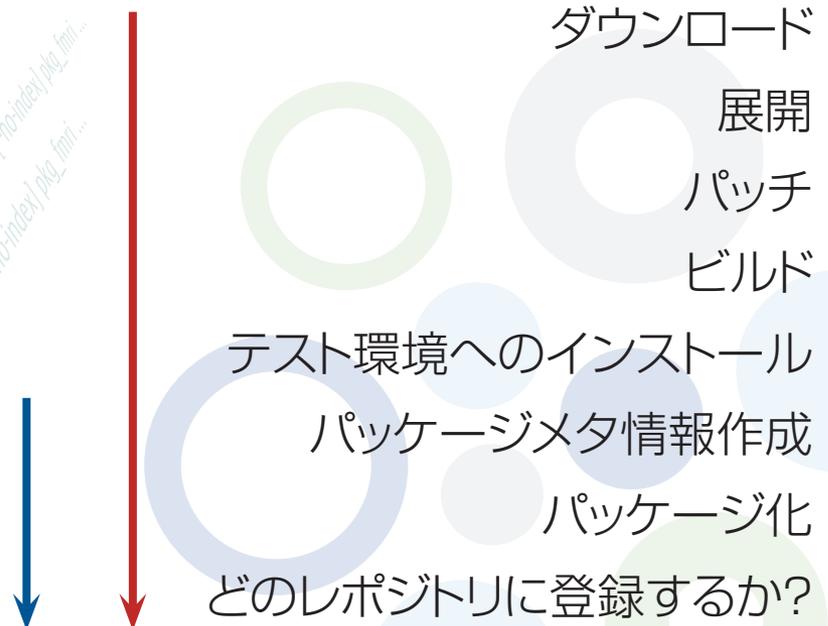


OpenSolaris Contrib Repo Build System



spec ファイルとは・・・

- RedHat Linux系からきている考え方。
- SolarisではSpec File Extra (SFE)という名前のパッケージ体系
- Manifestよりも、さらにバックヤードの考え方を持つ。
 - 配布サイトの場所
 - パッチ
 - ソースのビルド方法
 - パッケージの構成情報



spec ファイルのビルド環境作成 #1

開発環境のインストール

```
pfexec pkg install -v \  
  ss-dev \  
  SUNWgnome-common-devel \  
  SUNWperl-xml-parser \  
  SUNWgnome-xml-root SUNWgnome-xml SUNWgnome-xml-share
```

コンパイラ(開発環境)は、gcc-dev、ss-dev(StudioExpress)、Studio12が選べます。しかし、現時点でON(OS+Network)がStudio12でビルドしてあるため、「私としては」Studio12をおすすめしたいです。

Studio12は、こちらからダウンロードします。

http://www.opensolaris.org/os/community/tools/sun_studio_tools/sun_studio_12_tools/

Studio12のインストール作業

```
pfexec mkdir /opt/SUNWspro/  
cd /opt/SUNWspro/  
pfexec gtar zxvf sunstudio12-ii-20081010-sol-x86.tar.gz
```

spec ファイルのビルド環境作成 #2

JDS CBE (Common Build Environment) 1.7+をインストール

まず、オペレーションユーザにPrimary Administratorの権限をつけておきます。この作業をした後は、シェルの再起動をする必要があります。

```
pfexec usermod -P 'Primary Administrator' ユーザ名
```

JDS CBEをインストール

このプログラムはSVR4のパッケージでインストールされます。

```
wget \  
http://dlc.sun.com/osol/jds/downloads/cbe/test/desktop-cbe-1.7.0-rc1-x86.tar.bz2  
gtar jxvf desktop-cbe-1.7.0-rc1-x86.tar.bz2
```

spec ファイルのビルド環境作成 #3

```
% cd desktop-cbe-1.7.0-rc1
% ./cbe-install
Desktop Common Build Environment (CBE) Installer version 1.7.0-rc1
~以下、質問に答えます。とても長いので割愛~
途中、ブラウザが開いてレジストするよう促されるので、注意しましょう。
```

pkgbuildをダウンロード

cbeに含まれているpkgbuildが今は古いので、最新版を<http://sourceforge.net/projects/pkgbuild/>からダウンロードします。

1.3.1が入っているのでアンインストール

```
pfexec pkgrm SFpkgbuild
手コンパイルでインストール
```

```
cd pkgbuild-1.3.98
./configure --prefix=/opt/dtbld/
make
pfexec make install
```

spec ファイルのビルド環境作成 #4

ACLOCALについて

/opt/dtbuild/share/aclocal以下に、CBE m4のファイルが見つからないと、コンパイル時にACLOCALでエラーが出ます。この場合、/usr/share/aclocal/dirlistに、下記のものがあるか確認する必要があります。

```
/usr/sfw/share/aclocal  
/opt/dtbuild/share/aclocal
```

aclocalという名で実行できないのでシンボリックリンクで回避します。

```
ln -s /usr/bin/aclocal-1.10 /usr/bin/aclocal
```

SPECファイルのインストール

workディレクトリを決めて、svnからチェックアウトしてきます。

```
svn co svn+ssh://anon@svn.opensolaris.org/svn/jds/spec-files/trunk spec-files-trunk  
svn co svn+ssh://anon@svn.opensolaris.org/svn/jds/spec-files/branches/gnome-2-24 spec-files-2-24
```

pkgtoolを使う前に、JDSのコンポーネントをインストールします。manページとpo-sunです。

```
% cd spec-files-2-24/manpages  
% make  
% cd ../po-sun  
% make
```



spec ファイルのビルド環境作成 #5

環境変数の設定

```
./opt/dtbld/bin/env.sh  
Cシェルの場合
```

```
source /opt/dtbld/bin/env.csh  
LC_ALL=C ; export LC_ALL  
LANG=C ; export LANG
```

コンパイル

```
cd spec-files-2-24  
/opt/dtbld/bin/pkgtool --download --ips build-only SUNWTiff.spec
```

このようにすると、コンパイル後、IPSに自動的に登録します。

コンパイルの中間ファイルや結果は、~/packages/以下に保存されます。

--ipsではなく、--svr4にした場合、従来のSVR4のパッケージを作成します。

このようにすることで、同じメタ情報とソースコードから、IPSとSVR4の両方のレポジトリの作成が可能です。

IPS サーバのニーズ (LAN 内)

サーバシステムの同期用

- phpやperlで書かれたシステムの同期用
- glassfishなどでは標準用でデプロイ用のサーバがありますが、phpにはありません。
- L/Bの下に複数台のphpをおくときに便利かもしれません。

ステージサーバとして

- サービスプログラムを公開用のサーバにアップロードするとき、ISPサーバに一式をコミットして、あとは同期をかける。

LAN用のIPSは非常にたてやすいので、利用用途は高い。

公開用 IPS サーバ

IPSサーバのオプションは、`svccfg`を利用して設定します。

次のオプションがあります(参考:`man pkg.depotd`)

`pkg/content_root` ドキュメントルート。デザインを変更するときのコンポーネントはここに保存 (IPSは、`httpd`として起動している)。デフォルトは、`/usr/share/lib/pkg`

`pkg/inst_root` レポジトリの実体が入ります。レポジトリデータはこの場所にあるので、ここをコピーすることにより、レポジトリの複製が可能です。デフォルトは`/var/pkg/repo`。

`pkg/log_access` アクセスログ。

`pkg/log_errors` エラーログ。

`pkg/port` LISTENポート番号。

`pkg/proxy_base` Proxyサーバを経由したとき、実際にはどのURLになるのか。

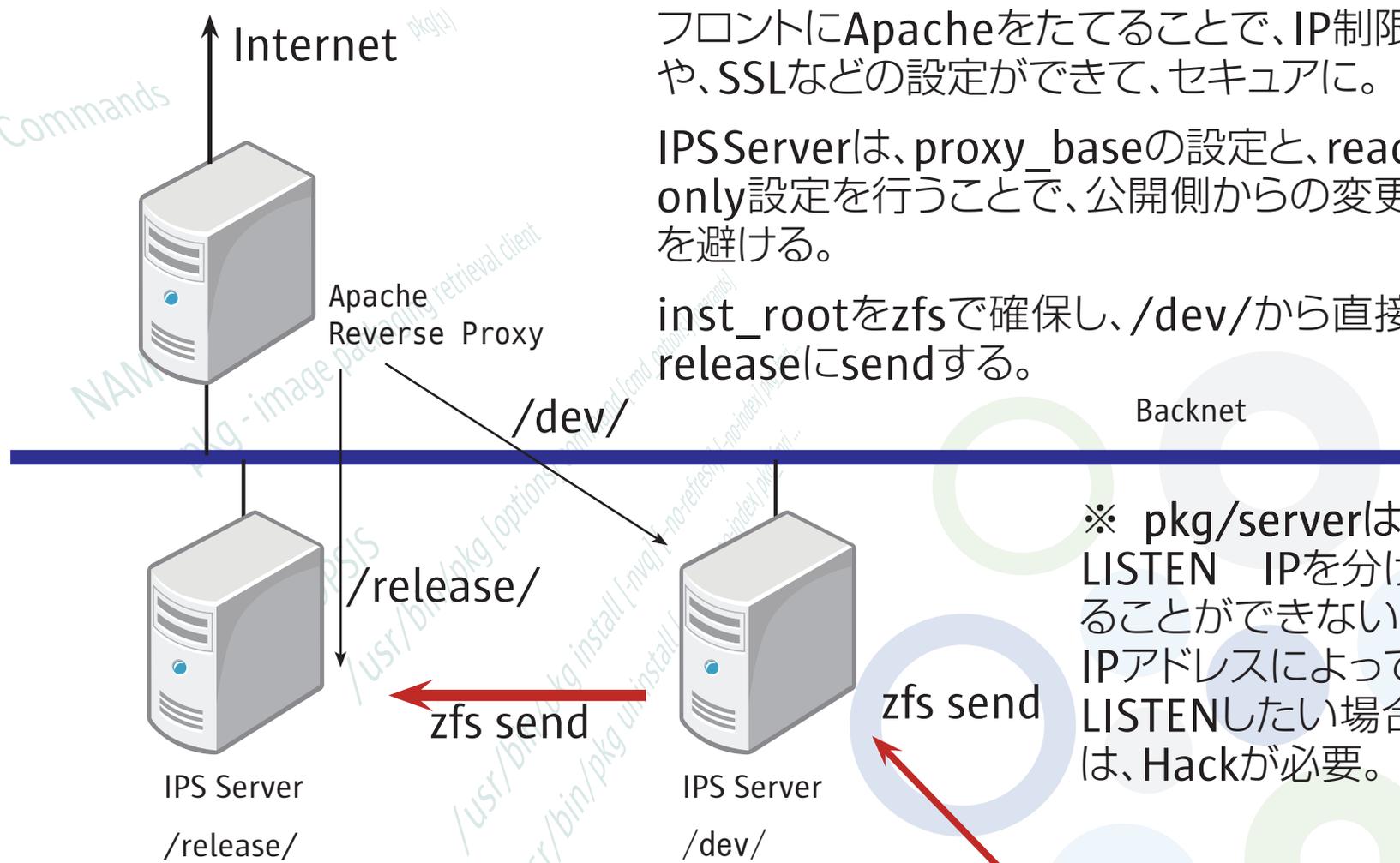
`pkg/readonly` リードオンリー。`true`にすると、`pkgsend`できなくなる。

`pkg/threads` スレッドの本数。デフォルトは10なので最大同時10人しかアクセスできない。

設定変更例

```
pfexec svccfg -s pkg/server "setprop pkg/port=10000"  
pfexec svcadm refresh pkg/server  
pfexec svcadm restart pkg/server
```

公開用 IPS ネットワーク図



フロントにApacheをたてることで、IP制限や、SSLなどの設定ができて、セキュアに。

IPSServerは、proxy_baseの設定と、read only設定を行うことで、公開側からの変更を避ける。

inst_rootをzfsで確保し、/dev/から直接releaseにsendする。

※ pkg/serverは、LISTEN IPを分けることができない。IPアドレスによってLISTENしたい場合は、Hackが必要。

開発用IPSより

IPS Q&A

ご静聴ありがとうございました。

OpenSolaris Users Group

ジャストプレイヤー株式会社

代表取締役社長 瀧 康史

