

Searching Repository on openSUSE.org

The aim is obvious: because of the dynamical structure you have to search through the build service - a static list does not exist. The script mentioned here scans a mirror server (must be ftp because of the crawling) to do so.

```
#!/usr/bin/perl -w

# #####
#
# This script loops through the directories of a
# mirror of the openSUSE build system and tries
# to figure out if the current directory might be
# a repository and if so, for which release.
# If it is, the directory is written to several
# output files: one for smart (to be placed inside
# /etc/smart/channels), one for zypp (to be stored
# inside /etc/zypp/repos.d), and in one of two
# possible channel lists for good old apt :-)
#
# Improvements are always welcome - I'm not
# a Perl programmer, as you will see when reading
# the source code ;-)
#
# written by Werner Flamme <w.flamme@web.de>
#
# #####
use strict;
use warnings;
use Net::FTP;

# set some variables
our $debug      = 0;
my $ftpserver   = 'ftp5.gwdg.de';
our $searchbase = 'pub/opensuse/repositories';
my $verzname    = "ftp://$ftpserver/$searchbase";
our $preface    = 'http://download.opensuse.org/repositories';

# get date and time (used for logging) and print them
my @dots;
my ($sec, $min, $hour, $mday, $mon, $year,
    $wday, $yday, $isdst) = localtime(time);
$year += 1900;
$mon++;
my $datstempel = sprintf("%04d%02d%02d", $year, $mon, $mday);
my $jetztzeit  = sprintf("%02d:%02d:%02d", $hour, $min, $sec);
print "starting script on $datstempel at $jetztzeit.\n";

# where to put the data - logfiles with timestamp in their names
```

```
my $logfilename = ">findrepo$datstempel\_jetztzeit.log";
our %allfiles = ('LOG' => $logfilename);
my @versionen = ('103', '110', '111', 'FAC', 'S09', 'S10');
for my $vers (@versionen) {
    $allfiles{"{$vers}apt"} =
        ">aptrepo_$vers\_datstempel\_jetztzeit.list";
    $allfiles{"{$vers}home"} =
        ">aptrepo_$vers\_datstempel\_jetztzeit\_home.list";
} # for my $vers (@versionen)
mkdir 'zypp103.d';
mkdir 'zypp110.d';
mkdir 'zypp111.d';
mkdir 'zyppFAC.d';
mkdir 'smart103.d';
mkdir 'smart110.d';
mkdir 'smart111.d';
mkdir 'smartFAC.d';

# the real start :-)
our $ftp = Net::FTP->new($ftpserver, Passive => 1)
    or die "Cannot access FTP server $ftpserver: $@";
print "Access to FTP server $ftpserver is possible\n"
    if ($debug & 2);
$ftp->login("anonymous", '-anonymous@')
    or die "Cannot login: ", $ftp->message;
print "anonymous login is possible :-)\n" if ($debug & 1);

# now start the loop!
scanldir($searchbase);

# post work - write out the time again so one can see how long it takes
($sec, $min, $hour, $mday, $mon, $year,
 $yday, $isdst) = localtime(time);
$year += 1900;
$mon++;
$datstempel = sprintf("%04d%02d%02d", $year, $mon, $mday);
$jetztzeit = sprintf("%02d:%02d:%02d", $hour, $min, $sec);
print "script finished on $datstempel at $jetztzeit.\n";

sub scanldir {
    our %allfiles;
    our $searchbase;
    our $preface;
    our $ftp;
    our $debug;
    my ($thisdir) = @_;
    # $thisdir is s.th. like home:/dings/openSUSE_10.3
    print "now working on $thisdir\n" if ($debug & 4);

    # may this be a repository? And for what version?
    my $right7 = substr($thisdir, -7);
```

```

my $right6 = substr($thisdir, -6);
my $right5 = substr($thisdir, -5);
my $right4 = substr($thisdir, -4);
my $right3 = substr($thisdir, -3);
my $forget = length("$searchbase");
my $shorter = substr($thisdir, $forget);
my $legible = $shorter;
my $itsfor = 'xxx';
my $usehome = ($thisdir =~ /home:\\/);
# now snip from the last / onward:
if (length($shorter)) {
    $legible =~ s|(.*)/(.+)|$1|;
    $legible =~ s|/||g;
    $right4 = substr($thisdir, -4);
    $right3 = substr($thisdir, -3);
    $itsfor = '110' if ( ($right4 eq '11.0' ) or
                        ($right3 eq '110' ) );
    $itsfor = '111' if ( ($right4 eq '11.1' ) or
                        ($right3 eq '111' ) );
    $itsfor = '103' if ( ($right4 eq '10.3' ) or
                        ($right3 eq '103' ) );
    $itsfor = 'FAC' if ( $right7 eq 'Factory' );
    $itsfor = 'S09' if ( ($right5 eq 'SLES9' ) or
                        ($right6 eq 'SLES_9' ) );
    $itsfor = 'S10' if ( ($right5 eq 'SLE10' ) or
                        ($right6 eq 'SLES10' ) or
                        ($right6 eq 'SLE_10' ) );
} # if (length($shorter))
$ftp->cwd("/$thisdir")
    or die "Cannot change into working directory $thisdir: ",
        $ftp->message;
my @dircontent = $ftp->dir();
my @unterverz = ();
my $ftpelem;

# walk through the directory
for $ftpelem (@dircontent) {
    # get file info
    my ($rechte, $num, $user, $group, $size,
        $mon, $day, $time, $name) = split / +/, $ftpelem, 9;
    # is it a directory?
    if (substr($rechte, 0, 1) eq 'd') {
        print "guessed directory: '$name' (itsfor=$itsfor, ",
            "shorter=$shorter, legible=$legible)\n" if ($debug & 1);
        # may it be of interest?
        if ((" $itsfor" ne 'xxx') and (" $name" eq 'repodata')) {
            # yes - so log and print and shout and...
            my ($sec, $min, $hour, $mday, $mon, $year,
                $wday, $yday, $isdst) = localtime(time);
            $year += 1900;
            $mon++;

```

```

my $jetztzeit = sprintf("%02d:%02d:%02d",
                        $hour, $min, $sec);
my $MYTOFILE = $allfiles{"LOG"};
open THISHANDLE, $allfiles{"LOG"}
    or die "$jetztzeit: Cannot open logfile";
print THISHANDLE "$jetztzeit: now writing ",
    "$legible for $thisdir\n";
close THISHANDLE;
# if home:/... entry, write into separate apt file
if ($usehome) {
    open THISHANDLE, $allfiles{"${itsfor}home"}
        or die "$jetztzeit: Cannot open apt-home ",
            "output file for $itsfor\n";
    print THISHANDLE "repomd $preface $shorter\n";
    close THISHANDLE;
} # if ($usehome)
# write into normal apt source
my $meld = "Cannot open apt output file >>" .
    $allfiles{"${itsfor}apt"};
open THISHANDLE, $allfiles{"${itsfor}apt"}
    or die "$jetztzeit: $meld<< for: $itsfor
($shorter)\n";
print THISHANDLE "repomd $preface $shorter\n";
close THISHANDLE;
if (($itsfor ne 'S09') and
    ($itsfor ne 'S10')) {
    # write into zypp repository file
    open ZYPPFILE, ">zypp${itsfor}.d/$legible.repo"
        or die "Cannot create file ",
            "zypp${itsfor}.d/$legible: $!";
    print ZYPPFILE "[$preface$shorter]\n";
    print ZYPPFILE "name=$legible\n";
    print ZYPPFILE "baseurl=$preface$shorter\n";
    print ZYPPFILE "type=rpm-md\n";
    print ZYPPFILE "enabled=1\n";
    print ZYPPFILE "autorefresh=1\n";
    print ZYPPFILE "gpgcheck=1\n";
    print ZYPPFILE "keeppackages=0\n"
        unless ("${itsfor}" eq '103');
    close ZYPPFILE;
    # write into smart channel file
    open THISHANDLE, ">smart${itsfor}.d/$legible.channel"
        or die "Cannot create file
smart${itsfor}.d",
            "/$legible.channel: $!";
    print THISHANDLE "[$legible]\n";
    print THISHANDLE "name=$legible\n";
    print THISHANDLE "baseurl=$preface$shorter\n";
    print THISHANDLE "type=rpm-md\n";
    close THISHANDLE;
} # if not SLEx

```

```

    } # if (($itsfor ne 'xxx') and (" $name" eq 'repodata'))
else {
    # so, it's not important. Put it on the stack,
    # maybe its content can be of some value ;-)
    print "adding $thisdir/$name to unterverz\n"
    if ($debug & 2);
    @unterverz = (@unterverz, "$thisdir/$name")
    if (" $name" ne 'lost+found');
} # else: if (($itsfor ne 'xxx') and (" $name" eq 'repodata'))
} # if (substr($rechte, 0, 1) eq 'd')
} # foreach $elem (@dircontent)

# well, what did we find? Let's loop at the stack...
if ((my $unterverz = @unterverz) > 0) {
    for my $untvz (@unterverz) {
        # yes, we are using recursion here
        scandir($untvz);
    } # for $untvz (@unterverz)
} # if (($unterverz = @unterverz) > 0)
1;
} # sub scandir

```

The script runs on my old IBM T23 every day and takes about 5 hours to complete – thanks to my “extremely fast” internet connection. In the office, it takes about 1 hour...

The gzip'ed lists of 2009-01-17 you find here:

Version	apt	smart	zypper
openSUSE 10.3	apt-103-list.gz apt-103-home-list.gz	smart103.d.tgz	zypp103.d.tgz
openSUSE 11.0	apt-110-list.gz apt-110-home-list.gz	smart110.d.tgz	zypp110.d.tgz
openSUSE 11.1	apt-111-list.gz apt-111-home-list.gz	smart111.d.tgz	zypp111.d.tgz
openSUSE Factory	apt-fac-list.gz apt-fac-home-list.gz	smartfac.d.tgz	zyppfac.d.tgz
SLES 9	apt-s09-list.gz apt-s09-home-list.gz	n. a.	n. a.
SLES/SLED 10	apt-s10-list.gz apt-s10-home-list.gz	n. a.	n. a.

The -home- lists contain the /home:/... repositories only; these repos are included in the “main” list.

From:

<http://www.wernerflamme.net/> - **Werners Wiki**

Permanent link:

<http://www.wernerflamme.net/doku.php?id=comp:en:getrepolist3>

Last update: **2009-01-18 0044**

