How to Install the Netops Toolkit

Contents

Overview

This document describes how to install the Netops Toolkit. Unless otherwise mentioned, source files can be found at http://www.skendric.com/nmgmt

Install Supporting Code

Install a Local Perl

If you have significant Perl expertise, you might decide to use your operating system's native Perl installation to support your use of the Netops Toolkit. In that case, you can skip this section. But I predict that most users will be happier if they create a locally compiled version of this tool.

Build a directory tree

I recommend creating a directory tree to contain locally compiled code, rather than overwriting your operating system's native copies. Something like this:

```
gnat% mkdir /opt/vdops
gnat% mkdir /opt/vdops/etc
gnat% mkdir /opt/vdops/etc/snmp
gnat% mkdir /opt/vdops/archive
gnat% mkdir /opt/vdops/src
```

Alternatively, you might change ‘vdops’ to a department or project name ('nss' for Network Support Services or 'local' or 'netops' ... whatever). Or, you might house all of this in your home directory, changing '/opt/vdops’ in the above example to '~'.

Compile Perl

Acquire a copy of *perl-5.12.2.tar.gz* from http://www.cpan.org and store it in /opt/vdops/archive.

```
gnat% cd /opt/vdops/src
gnat% tar xvfz ../archive/perl-5.12.2.tar.gz
[...]
gnat% cd perl-5.12.2
gnat% rm -f config.sh Policy.sh[1]
gnat% ./Configure -de -Dprefix=/opt/vdops -Dscriptdir=/opt/vdops/bin -
Dusethreads -Duseshrplib

[...]
gnat% make
[...]
gnat% make test
[...]
```

[1] Files may not exist in your install; delete ‘em if they do exist

```
gnat% make install
[...]
gnat% cd /usr/include
gnat% /opt/vdops/bin/h2ph * sys/*
[...]
gnat%
gnat% /opt/vdops/bin/perl -v
This is perl, v5.12.1 built for i686-linux-thread-multi
[...]
gnat%
```

Install makepatch

```
gnat% pwd
/opt/vdops/src
gnat% tar xvfz /opt/vdops/archive/makepatch*
[…]
gnat% cd makepatch-2.04/
gnat% perl Makefile.PL
[…]
Checking if your kit is complete...
Looks good
Writing Makefile for makepatch
gnat% make
[…]
gnat% make install
[…]
gnat%
```

Configure cpan

'cpan' is a script which automates the process of finding, downloading, building, and installing supplementary Perl modules. The first time you run it, it will want to be configured.

```
gnat% pwd
/home/netops
gnat% rm -rf .cpan
gnat% ls /opt/vdops/lib/perl5/5.12.2/CPAN/Config.pm
/opt/vdops/lib/perl5/5.12.2/CPAN/Config.pm
gnat% rm /opt/vdops/lib/perl5/5.12.2/CPAN/Config.pm*
gnat% ls /opt/vdops/lib/perl5/5.12.2/CPAN/Config.pm
ls: /opt/vdops/lib/perl5/5.12.2/CPAN/Config.pm: No such file or directory
gnat% cpan


vishnu> cpan

CPAN is the world-wide archive of perl resources. It consists of about
300 sites that all replicate the same contents around the globe. Many
countries have at least one CPAN site already. The resources found on
CPAN are easily accessible with the CPAN.pm module. If you want to use
CPAN.pm, lots of things have to be configured. Fortunately, most of
them can be determined automatically. If you prefer the automatic
```

```
configuration, answer 'yes' below.

If you prefer to enter a dialog instead, you can answer 'no' to this
question and I'll let you configure in small steps one thing after the
other. (Note: you can revisit this dialog anytime later by typing 'o
conf init' at the cpan prompt.)

Would you like me to configure as much as possible automatically? [yes] no



The following questions are intended to help you with the
configuration. The CPAN module needs a directory of its own to cache
important index files and maybe keep a temporary mirror of CPAN files.
This may be a site-wide or a personal directory.



I see you already have a  directory
    /home/skendric/.cpan
Shall we use it as the general CPAN build and cache directory?

 <cpan_home>
CPAN build and cache directory? [/home/skendric/.cpan]

Unless you are accessing the CPAN on your filesystem via a file: URL,
CPAN.pm needs to keep the source files it downloads somewhere. Please
supply a directory where the downloaded files are to be kept.

 <keep_source_where>
Download target directory? [/home/skendric/.cpan/sources]

 <build_dir>
Directory where the build process takes place? [/home/skendric/.cpan/build]

Until version 1.88 CPAN.pm never trusted the contents of the build_dir
directory between sessions. Since 1.88_58 CPAN.pm has a YAML-based
mechanism that makes it possible to share the contents of the
build_dir/ directory between different sessions with the same version
of perl. People who prefer to test things several days before
installing will like this feature because it safes a lot of time.

If you say yes to the following question, CPAN will try to store
enough information about the build process so that it can pick up in
future sessions at the same state of affairs as it left a previous
session.

 <build_dir_reuse>
Store and re-use state information about distributions between
CPAN.pm sessions? [no] yes

CPAN.pm can store customized build environments based on regular
expressions for distribution names. These are YAML files where the
default options for CPAN.pm and the environment can be overridden and
```

```
dialog sequences can be stored that can later be executed by an
Expect.pm object. The CPAN.pm distribution comes with some prefab YAML
files that cover sample distributions that can be used as blueprints
to store one own prefs. Please check out the distroprefs/ directory of
the CPAN.pm distribution to get a quick start into the prefs system.

 <prefs_dir>
Directory where to store default options/environment/dialogs for
building modules that need some customization? [/home/skendric/.cpan/prefs]

Normally CPAN.pm keeps config variables in memory and changes need to
be saved in a separate 'o conf commit' command to make them permanent
between sessions. If you set the 'auto_commit' option to true, changes
to a config variable are always automatically committed to disk.

 <auto_commit>
Always commit changes to config variables to disk? [no] yes

CPAN.pm can limit the size of the disk area for keeping the build
directories with all the intermediate files.

 <build_cache>
Cache size for build directory (in MB)? [100]

The CPAN indexes are usually rebuilt once or twice per hour, but the
typical CPAN mirror mirrors only once or twice per day. Depending on
the quality of your mirror and your desire to be on the bleeding edge,
you may want to set the following value to more or less than one day
(which is the default). It determines after how many days CPAN.pm
downloads new indexes.

 <index_expire>
Let the index expire after how many days? [1]

By default, each time the CPAN module is started, cache scanning is
performed to keep the cache size in sync. To prevent this, answer
'never'.

 <scan_cache>
Perform cache scanning (atstart or never)? [atstart]

To considerably speed up the initial CPAN shell startup, it is
possible to use Storable to create a cache of metadata. If Storable is
not available, the normal index mechanism will be used.

Note: this mechanism is not used when use_sqlite is on and SQLLite is
running.

 <cache_metadata>
Cache metadata (yes/no)? [yes]

CPAN::SQLite is a layer between the index files that are downloaded
from the CPAN and CPAN.pm that speeds up metadata queries and reduces
memory consumption of CPAN.pm considerably.
```

```
 <use_sqlite>
Use CPAN::SQLite if available? (yes/no)? [no]

The CPAN module can detect when a module which you are trying to build
depends on prerequisites. If this happens, it can build the
prerequisites for you automatically ('follow'), ask you for
confirmation ('ask'), or just ignore them ('ignore').  Choosing
'follow' also sets PERL_AUTOINSTALL and PERL_EXTUTILS_AUTOINSTALL for
"--defaultdeps" if not already set.

Please set your policy to one of the three values.

 <prerequisites_policy>
Policy on building prerequisites (follow, ask or ignore)? [follow]

When a module declares another one as a 'build_requires' prerequisite
this means that the other module is only needed for building or
testing the module but need not be installed permanently. In this case
you may wish to install that other module nonetheless or just keep it
in the 'build_dir' directory to have it available only temporarily.
Installing saves time on future installations but makes the perl
installation bigger.

You can choose if you want to always install (yes), never install (no)
or be always asked. In the latter case you can set the default answer
for the question to yes (ask/yes) or no (ask/no).

 <build_requires_install_policy>
Policy on installing 'build_requires' modules (yes, no, ask/yes,
ask/no)? [yes]

CPAN packages can be digitally signed by authors and thus verified
with the security provided by strong cryptography. The exact mechanism
is defined in the Module::Signature module. While this is generally
considered a good thing, it is not always convenient to the end user
to install modules that are signed incorrectly or where the key of the
author is not available or where some prerequisite for
Module::Signature has a bug and so on.

With the check_sigs parameter you can turn signature checking on and
off. The default is off for now because the whole tool chain for the
functionality is not yet considered mature by some. The author of
CPAN.pm would recommend setting it to true most of the time and
turning it off only if it turns out to be annoying.

Note that if you do not have Module::Signature installed, no signature
checks will be performed at all.

 <check_sigs>
Always try to check and verify signatures if a SIGNATURE file is in
the package and Module::Signature is installed (yes/no)? [no] yes

The goal of the CPAN Testers project (http://testers.cpan.org/) is to
```

```
test as many CPAN packages as possible on as many platforms as
possible.  This provides valuable feedback to module authors and
potential users to identify bugs or platform compatibility issues and
improves the overall quality and value of CPAN.

One way you can contribute is to send test results for each module
that you install.  If you install the CPAN::Reporter module, you have
the option to automatically generate and email test reports to CPAN
Testers whenever you run tests on a CPAN package.

See the CPAN::Reporter documentation for additional details and
configuration settings.  If your firewall blocks outgoing email,
you will need to configure CPAN::Reporter before sending reports.

 <test_report>
Email test reports if CPAN::Reporter is installed (yes/no)? [no]

When a distribution has already been tested by CPAN::Reporter on
this machine, CPAN can skip the test phase and just rely on the
test report history instead.

Note that this will not apply to distributions that failed tests
because of missing dependencies.  Also, tests can be run
regardless of the history using "force".

 <trust_test_report_history>
Do you want to rely on the test report history (yes/no)? [no]

At the time of this writing (2009-03) there are three YAML
implementations working: YAML, YAML::Syck, and YAML::XS. The latter
two are faster but need a C compiler installed on your system. There
may be more alternative YAML conforming modules. When I tried two
other players, YAML::Tiny and YAML::Perl, they seemed not powerful
enough to work with CPAN.pm. This may have changed in the meantime.

 <yaml_module>
Which YAML implementation would you prefer? [YAML]

Warning (maybe harmless): 'YAML' not installed.
Both YAML.pm and YAML::Syck are capable of deserialising code. As this
requires a string eval, which might be a security risk, you can use
this option to enable or disable the deserialisation of code via
CPAN::DeferredCode. (Note: This does not work under perl 5.6)

 <yaml_load_code>
Do you want to enable code deserialisation (yes/no)? [no]



The CPAN module will need a few external programs to work properly.
Please correct me, if I guess the wrong path for a program. Don't
panic if you do not have some of them, just press ENTER for those. To
disable the use of a program, you can type a space followed by ENTER.
```

```
 <bzip2>
Where is your bzip2 program? [/usr/bin/bzip2]

 <gzip>
Where is your gzip program? [/bin/gzip]

 <tar>
Where is your tar program? [/bin/tar]

 <unzip>
Where is your unzip program? [/usr/bin/unzip]

 <make>
Where is your make program? [/usr/bin/make]

 <curl>
Where is your curl program? [/usr/bin/curl]

 <lynx>
Where is your lynx program? [/usr/bin/lynx]

 <wget>
Where is your wget program? [/usr/bin/wget]

 <ncftpget>
Where is your ncftpget program? [/usr/bin/ncftpget]

 <ftp>
Where is your ftp program? [/usr/kerberos/bin/ftp]

 <gpg>
Where is your gpg program? [/usr/bin/gpg]

 <patch>
Where is your patch program? [/usr/bin/patch]

 <applypatch>
Where is your applypatch program? [/opt/vdops/bin/applypatch]

 <pager>
What is your favorite pager program? [/usr/bin/less]

 <shell>
What is your favorite shell? [/bin/bash]

When CPAN.pm uses the tar command, which switch for the verbosity
shall be used? Choose 'none' for quiet operation, 'v' for file
name listing, 'vv' for full listing.

 <tar_verbosity>
Tar command verbosity level (none or v or vv)? [none]

When CPAN.pm loads a module it needs for some optional feature, it
usually reports about module name and version. Choose 'v' to get this
```

```
message, 'none' to suppress it.

 <load_module_verbosity>
Verbosity level for loading modules (none or v)? [none]

When CPAN.pm extends @INC via PERL5LIB, it prints a list of
directories added (or a summary of how many directories are
added).  Choose 'v' to get this message, 'none' to suppress it.

 <perl5lib_verbosity>
Verbosity level for PERL5LIB changes (none or v)? [none]

When the CPAN shell is started it normally displays a greeting message
that contains the running version and the status of readline support.

 <inhibit_startup_message>
Do you want to turn this message off? [no]

When you have Module::Build installed and a module comes with both a
Makefile.PL and a Build.PL, which shall have precedence?

The main two standard installer modules are the old and well
established ExtUtils::MakeMaker (for short: EUMM) which uses the
Makefile.PL. And the next generation installer Module::Build (MB)
which works with the Build.PL (and often comes with a Makefile.PL
too). If a module comes only with one of the two we will use that one
but if both are supplied then a decision must be made between EUMM and
MB. See also http://rt.cpan.org/Ticket/Display.html?id=29235 for a
discussion about the right default.

Or, as a third option you can choose RAND which will make a random
decision (something regular CPAN testers will enjoy).

 <prefer_installer>
In case you can choose between running a Makefile.PL or a Build.PL,
which installer would you prefer (EUMM or MB or RAND)? [MB]

Every Makefile.PL is run by perl in a separate process. Likewise we
run 'make' and 'make install' in separate processes. If you have
any parameters (e.g. PREFIX, UNINST or the like) you want to
pass to the calls, please specify them here.

If you don't understand this question, just press ENTER.

Typical frequently used settings:

    PREFIX=~/perl    # non-root users (please see manual for more hints)

 <makepl_arg>
Parameters for the 'perl Makefile.PL' command? []

Parameters for the 'make' command? Typical frequently used setting:

    -j3              # dual processor system (on GNU make)
```

```
 <make_arg>
Your choice: []

Do you want to use a different make command for 'make install'?
Cautious people will probably prefer:

    su root -c make
 or
    sudo make
 or
    /path1/to/sudo -u admin_account /path2/to/make

 <make_install_make_command>
or some such. Your choice: [/usr/bin/make]

Parameters for the 'make install' command?
Typical frequently used setting:

    UNINST=1         # to always uninstall potentially conflicting files

 <make_install_arg>
Your choice: []

A Build.PL is run by perl in a separate process. Likewise we run
'./Build' and './Build install' in separate processes. If you have any
parameters you want to pass to the calls, please specify them here.

Typical frequently used settings:

    --install_base /home/xxx             # different installation directory

 <mbuildpl_arg>
Parameters for the 'perl Build.PL' command? []

Parameters for the './Build' command? Setting might be:

    --extra_linker_flags -L/usr/foo/lib  # non-standard library location

 <mbuild_arg>
Your choice: []

Do you want to use a different command for './Build install'? Sudo
users will probably prefer:

    su root -c ./Build
 or
    sudo ./Build
 or
    /path1/to/sudo -u admin_account ./Build

 <mbuild_install_build_command>
or some such. Your choice: [./Build]
```

```
Parameters for the './Build install' command? Typical frequently used
setting:

    --uninst 1                          # uninstall conflicting files

 <mbuild_install_arg>
Your choice: []

Sometimes you may wish to leave the processes run by CPAN alone
without caring about them. Because the Makefile.PL or the Build.PL
sometimes contains question you're expected to answer, you can set a
timer that will kill a 'perl Makefile.PL' process after the specified
time in seconds.

If you set this value to 0, these processes will wait forever. This is
the default and recommended setting.

 <inactivity_timeout>
Timeout for inactivity during {Makefile,Build}.PL? [0]

This timeout prevents CPAN from hanging when trying to parse a
pathologically coded $VERSION from a module.

The default is 15 seconds.  If you set this value to 0, no timeout
will occur, but this is not recommended.

 <version_timeout>
Timeout for parsing module versions? [15]

Normaly, CPAN.pm continues processing the full list of targets and
dependencies, even if one of them fails.  However, you can specify
that CPAN should halt after the first failure.

 <halt_on_failure>
Do you want to halt on failure (yes/no)? [no]



If you're accessing the net via proxies, you can specify them in the
CPAN configuration or via environment variables. The variable in
the $CPAN::Config takes precedence.

 <ftp_proxy>
Your ftp_proxy? []

 <http_proxy>
Your http_proxy? []

 <no_proxy>
Your no_proxy? []

 <ftp_passive>
Shall we always set the FTP_PASSIVE environment variable when dealing
with ftp download (yes/no)? [yes]
```

```
CPAN.pm changes the current working directory often and needs to
determine its own current working directory. Per default it uses
Cwd::cwd but if this doesn't work on your system for some reason,
alternatives can be configured according to the following table:

    cwd         Cwd::cwd
    getcwd      Cwd::getcwd
    fastcwd     Cwd::fastcwd
    backtickcwd external command cwd

 <getcwd>
Preferred method for determining the current working directory? [cwd]

The prompt of the cpan shell can contain the current command number
for easier tracking of the session or be a plain string.

 <commandnumber_in_prompt>
Do you want the command number in the prompt (yes/no)? [yes]

When using Term::ReadLine, you can turn ornaments on so that your
input stands out against the output from CPAN.pm.

 <term_ornaments>
Do you want to turn ornaments on? [yes]

When you have Term::ANSIColor installed, you can turn on colorized
output to have some visual differences between normal CPAN.pm output,
warnings, debugging output, and the output of the modules being
installed. Set your favorite colors after some experimenting with the
Term::ANSIColor module.

 <colorize_output>
Do you want to turn on colored output? [no]

The next option deals with the charset (aka character set) your
terminal supports. In general, CPAN is English speaking territory, so
the charset does not matter much but some CPAN have names that are
outside the ASCII range. If your terminal supports UTF-8, you should
say no to the next question. If it expects ISO-8859-1 (also known as
LATIN1) then you should say yes. If it supports neither, your answer
does not matter because you will not be able to read the names of some
authors anyway. If you answer no, names will be output in UTF-8.

 <term_is_latin>
Your terminal expects ISO-8859-1 (yes/no)? [yes]

If you have one of the readline packages (Term::ReadLine::Perl,
Term::ReadLine::Gnu, possibly others) installed, the interactive CPAN
shell will have history support. The next two questions deal with the
filename of the history file and with its size. If you do not want to
set this variable, please hit SPACE RETURN to the following question.

If you have one of the readline packages (Term::ReadLine::Perl,
```

```
Term::ReadLine::Gnu, possibly others) installed, the interactive CPAN
shell will have history support. The next two questions deal with the
filename of the history file and with its size. If you do not want to
set this variable, please hit SPACE RETURN to the following question.

 <histfile>
File to save your history? [/home/skendric/.cpan/histfile]

 <histsize>
Number of lines to save? [100]

The 'd' and the 'm' command normally only show you information they
have in their in-memory database and thus will never connect to the
internet. If you set the 'show_upload_date' variable to true, 'm' and
'd' will additionally show you the upload date of the module or
distribution. Per default this feature is off because it may require a
net connection to get at the upload date.

 <show_upload_date>
Always try to show upload date with 'd' and 'm' command (yes/no)? [no]

During the 'r' command CPAN.pm finds modules without version number.
When the command finishes, it prints a report about this. If you
want this report to be very verbose, say yes to the following
variable.

 <show_unparsable_versions>
Show all individual modules that have no $VERSION? [no] yes

During the 'r' command CPAN.pm finds modules with a version number of
zero. When the command finishes, it prints a report about this. If you
want this report to be very verbose, say yes to the following
variable.

 <show_zero_versions>
Show all individual modules that have a $VERSION of zero? [no] yes

If you have never defined your own C<urllist> in your configuration
then C<CPAN.pm> will be hesitant to use the built in default sites for
downloading. It will ask you once per session if a connection to the
internet is OK and only if you say yes, it will try to connect. But to
avoid this question, you can choose your favorite download sites once
and get away with it. Or, if you have no favorite download sites
answer yes to the following question.

 <connect_to_internet_ok>
If no urllist has been chosen yet, would you prefer CPAN.pm to connect
to the built-in default sites without asking? (yes/no)? [no] yes


commit: wrote '/opt/vdops/lib/perl5/5.12.2/CPAN/Config.pm'
Terminal does not support AddHistory.

cpan shell -- CPAN exploration and modules installation (v1.9456)
```

```
Enter 'h' for help.

cpan[1]>
```

Install Generally Useful Modules

Hit Enter whenever the install stops and asks you a question.

```
cpan[1]> install YAML LWP Bundle::CPAN CPAN::Reporter CPAN::SQLite
Module::Signature Test::Pod Test::Pod::Coverage PadWalker
Digest::BubbleBabble Test::Distribution Test::Portability::Files Test::Output
Test::Signature

[...]

cpan[2]>
```

Upgrade Modules

Upgrade the installed modules to the latest version. Ignore errors. This takes a while.

```
cpan[3]> upgrade
CPAN: Storable loaded ok (v2.18)
Going to read /home/netops/.cpan/Metadata
  Database was generated on Tue, 02 Sep 2008 00:02:48 GMT
CPAN: LWP::UserAgent loaded ok (v5.814)
CPAN: Time::HiRes loaded ok (v1.9715)
[...]
Package namespace         installed    latest  in CPAN file
Apache2::Reload                0.09      0.10  PHRED/Apache-Reload-0.10.tar.gz
B                            1.09_01      1.17  RGARCIA/perl-5.12.2.tar.gz
B::Debug                       1.09      1.11  RURBAN/B-Debug-1.11.tar.gz
ModPerl::MethodLookup         undef     0.009  GOZER/mod_perl-2.0.4.tar.gz
Net::Nslookup                  1.16      1.18  DARREN/Net-Nslookup-1.18.tar.gz
[...]
Running install for module 'Math::BigRat'
Running make for L/LE/LETO/Math-BigRat-0.24.tar.gz
  Did not pass the signature test.
Running make test
  Make had some problems, won't test
Running make install
  Make had some problems, won't install

cpan[4]>
```

Install Netops-Related Modules

The Netops Toolkit relies on a selection of modules from CPAN. Install them.

```
cpan[4]>  install autovivification Array::Slice Array::Utils Carp
Compress::Zlib Config::INIPlus Contextual::Return Crypt::SSLeay Cwd
Data::Dumper Data::Dump::Streamer Data::Validate::IP Date::Calc Date::Manip
DateTime DBI DBD::Pg English Exporter File::Basename::Object File::Copy
File::Path File::Stat File::Tail File::Temp File::Util GD::Simple IO::File
IO::Interactive IO::Socket::SSL IPC::Open3 Lingua::EN::Inflect List::Compare
List::MoreUtils List::Util LWP Mail::Send NetAddr::IP::Find Net::DNS Net::ARP
Net::FTP Net::Ifconfig::Wrapper Net::NBName Net::IP Net::IPAddress
Net::IPv4Addr Net::Netmask Net::LDAP::Express  Net::Ping::External
Net::Nslookup Net::SCP Net::SCP::Expect Net::SNMP Net::SNPP Net::Syslog
Nmap::Parser Perl6::Builtins Perl6::Slurp POE POE::Component::Client::Ping
Proc::ProcessTable Proc::Reliable Readonly Regexp::Common Set::Array
Spreadsheet::ParseExcel String::Similarity String::Util Sys::Hostname
Text::Diff Text::Trim Text::Diff Thread::Running Time::Concise Time::Duration
Time::Local Time::localtime Time::Period Thread::State version

Recursive dependency detected:
    Cwd (have: 3.2501; want: 3.29)
 => SMUELLER/PathTools-3.29.tar.gz
 => ExtUtils::CBuilder (have: 0.21; build_requires: 0)
 => KWILLIAMS/ExtUtils-CBuilder-0.24.tar.gz
 => File::Spec (have: 3.2501; requires: 0)
 => SMUELLER/PathTools-3.29.tar.gz.
Cannot resolve.
Running install for module 'Carp'
[...]
[... hit Return as needed ...]

lo:     UP
        inet 127.0.0.1       mask 255.0.0.0
sit0:   DOWN
====================================
Is Net::Ifconfig::Wrapper info output correct? Y/N:Y
[...]
Failed during this command:
 SMUELLER/PathTools-3.29.tar.gz               : make NO cannot resolve
circular dependency
 KWILLIAMS/ExtUtils-CBuilder-0.24.tar.gz      : make NO cannot resolve
circular dependency
 NWCLARK/perl-5.8.9.tar.gz                    : make NO isa perl
 BSUGARS/IPC-Shareable-0.60.tar.gz            : make_test NO
 DCONWAY/Perl6-Builtins-0.0.3.tar.gz          : make_test NO
 RCLAMP/File-Find-Rule-0.30.tar.gz            : make_test NO
 PETDANCE/Test-Pod-Coverage-1.08.tar.gz       : make_test NO
 SRSHAH/Test-Distribution-2.00.tar.gz         : make_test NO 3 dependencies
missing (Pod::Coverage,File::Find::Rule,Test::Pod::Coverage)
 AVIF/Time-Duration-1.06.tar.gz               : make_test NO

cpan[6] force install Perl6::Builtins Proc::ProcessTable IPC::Shareable
Manifying blib/man3/IPC::Shareable.3
  BSUGARS/IPC-Shareable-0.60.tar.gz
  /usr/bin/make install -j3 -- OK
Failed during this command:
```

```
 DCONWAY/Perl6-Builtins-0.0.3.tar.gz          : make_test FAILED but failure
ignored because 'force' in effect
 BSUGARS/IPC-Shareable-0.60.tar.gz            : make_test FAILED but failure
ignored because 'force' in effect

cpan[7]> exit
gnat%
```

Fix GD

The install routine for the GD module does not always copy one of its files, Group.pm, to the appropriate location. This is a bug.[2] We work around it manually in the following way.

First, determine whether or not this has happened in your installation.

In this example, Group.pm is there; this is good; you can skip ahead to the next step.

```
gnat% cd /opt/vdops/lib/perl5
gnat% find . | grep Group.pm
./site_perl/5.12.2/i686-linux-thread-multi/GD/Group.pm
gnat%
```

However, in the following example, Group.pm is missing; proceed with the rest of this step.

```
gnat% cd /opt/vdops/lib/perl5
gnat% find . | grep Group.pm
gnat%
```

First, figure out where it should have landed.

```
gnat% find . | grep Simple.pm
[...]
./site_perl/5.12.2/x86_64-linux-thread-multi/GD/Simple.pm
[...]
gnat%
```

Then, go find the copy unwrapped by your invocation of cpan.[3]

```
gnat% cd ~/.cpan
gnat% find . | grep Group.pm
./build/GD-2.41-bPErNA/GD/Group.pm
gnat%
```

Copy Group.pm to the appropriate location.

```
cp ./build/GD-2.41-bPErNA/GD/Group.pm
/opt/vdops/lib/perl5/site_perl/5.12.2/x86_64-linux-thread-multi/GD
```

[2] I believe this is fixed now. 2011-03-04 --sk

[3] If it isn't there, force an install, like this: *cpan> force install GD*

Enhance Net::Ping::External

This step is not strictly necessary[4], but it does improve performance of some of the scripts. Some of the scripts set the timeout parameter to varying values; currently, Net::Ping::External under Linux does not yet support this: the man page says it does, but if you are running under Linux, the code ignores whatever you set and reverts to the distro's default, typically 5 seconds. This modification introduces support for the timeout value. I have tested this under SuSE and RedHat.

Edit `/opt/vdops/lib/perl5/site_perl/5.12.2/Net/Ping/External.pm` and change this section:

```
# Debian 2.2 OK, RedHat 6.2 OK
# -s size option available to superuser... FIXME?
sub _ping_linux {
  my %args = @_;
  my $command;
#for next version
  if (-e '/etc/redhat-release' || -e '/etc/SuSE-release') {
    $command = "ping -c $args{count} -s $args{size} $args{host}";
  } else {
    $command = "ping -c $args{count} $args{host}";
  }
  return _ping_system($command, 0);
}
```

until it looks like this:

```
# Debian 2.2 OK, RedHat 6.2 OK
# -s size option available to superuser... FIXME?
sub _ping_linux {
  my %args = @_;
  my $command;
  $command = "ping -c $args{count} -s $args{size} -W $args{timeout}
$args{host}";
  return _ping_system($command, 0);
}
```

Install a Local Net-SNMP

If you have significant Perl expertise, you might decide to use your operating system's native Net-SNMP installation to support your use of the Netops Toolkit. In that case, you can skip this section. But I predict that most users will be happier if they create a locally compiled version of this tool.

[4] Appears to be entirely unnecessary under perl-5.12.2 --sk.

Compile Net-SNMP

Acquire a copy of *net-snmp-5.6.1.tar.gz* from http://net-snmp.sourceforge.net and store it in `/opt/vdops/archive`.

```
gnat% cd /opt/vdops/src
gnat% tar xvfz ../../archive/net-snmp-5.6.1.tar.gz
[...]
gnat% cd net-snmp-5.6.1
gnat% ./configure --prefix=/opt/vdops --with-
mibdirs="/opt/vdops/share/snmp/mibs" --with-persistent-
directory="/opt/vdops/var/snmp" --with-sys-contact=root --with-
logfile="/opt/vdops/var/log/net-snmp" --with-default-snmp-version="2" --with-
out-transports="TCP" --with-perl-modules --with-mib-modules="ucd-snmp/diskio
ucd-snmp/lmSensors smux --disable-embedded-perl"
[...]
gnat% make
[...]
gnat% make test
[...]
gnat% make install
[...]
gnat%
gnat% /opt/vdops/bin/snmptranslate -On -IR sysDescr.0
.1.3.6.1.2.1.1.1.0[5]
gnat%
```

Install Enterprise MIB Files

Acquire the Netops enterprise MIB file collection *Netops-MIBs.tar.bz2* and store it in `/opt/vdops/archive`.[6] Uncompress/untar them to wherever you store MIB files: typically `/usr/share/snmp/mibs` if you are using your operating systems native Net-SNMP installation. Or `/opt/vdops/share/snmp/mibs` if you are employing a locally compiled Net-SNMP installation.

```
gnat% cd /opt/vdops/share/snmp
gnat% rm -rf *
gnat% bunzip2 -c /opt/vdops/archive/Netops-MIBs.tar.bz2 | tar xvf -
[...]
gnat% ls mibs
gnat% ls
ADIC                chk_dups    HP             Netopia       Sun
AlliedTelesyn       Cisco       IBM            Netscreen     Symbios
APC                 Compaq      IBM-CLEAN      NetSNMP       Tandem
Apple               Crossroads  IETF           Networth      Tektronix
ARCserve            Cyclades    Intel          Nokia         Test
ARCserver-Alarm-MIB Cyclone     Konica         Nortel        TippingPoint
Aruba               DEC         Lantronix      Novell        Transition
Ascend              Dell        Lexmark        Okidata       Unisys
```

[5] May need to 'Acquire *snmp.conf…*' (see below) before this works.
[6] Generally available as /opt/vdops/archive/mibs.tar.gz. In which case, uncompress using the statement: `tar xvfz /opt/vdops/archive/mibs.tar.gz`

```
Availant            Dialogic    Liebert        PF            Uptime
AVAM-SNMPv1         DMTF        Linksys        Platform      Veritas
Axis                DPS         LSI            Polycom       VMWare
bkupexec.mib        Enterasys   McData         Qlogic        WTCS
BlackBerry          ESI         mib_index.txt  Radlan        Wyse
BlueArc             Extreme     Microsoft      Raritan       Xerox
BMC                 Freeradius  Minolta        Seagate       Xyplex
Broadcom            Gnome       mkindex        ServerTech
Brocade             Grabbag     Mylex          SMC
CheckPoint          Hitachi     NetApp         SNMPResearch
gnat%
```

Acquire *snmp.conf* and copy it to /opt/vdops/etc/snmp/snmp.conf

```
gnat% cp ~/snmp.conf.5 /opt/vdops/etc/snmp/snmp.conf
gnat% cat /opt/vdops/etc/snmp/snmp.conf
defVersion 2c
mibAllowUnderline yes
strictCommentTerm no
mibs ALL
mibdirs +/opt/vdops/share/snmp/mibs
mibdirs +/opt/vdops/share/snmp/mibs/Test
mibdirs +/opt/vdops/share/snmp/mibs/3Par
mibdirs +/opt/vdops/share/snmp/mibs/ADIC
mibdirs +/opt/vdops/share/snmp/mibs/AlliedTelesyn
mibdirs +/opt/vdops/share/snmp/mibs/APC
mibdirs +/opt/vdops/share/snmp/mibs/Apple
mibdirs +/opt/vdops/share/snmp/mibs/Aruba
mibdirs +/opt/vdops/share/snmp/mibs/Asante
mibdirs +/opt/vdops/share/snmp/mibs/Ascend
mibdirs +/opt/vdops/share/snmp/mibs/Asentria
mibdirs +/opt/vdops/share/snmp/mibs/Axis
mibdirs +/opt/vdops/share/snmp/mibs/BlackBerry
mibdirs +/opt/vdops/share/snmp/mibs/BlueArc
mibdirs +/opt/vdops/share/snmp/mibs/Broadcom
mibdirs +/opt/vdops/share/snmp/mibs/Brocade
mibdirs +/opt/vdops/share/snmp/mibs/Cabletron
mibdirs +/opt/vdops/share/snmp/mibs/CheckPoint
mibdirs +/opt/vdops/share/snmp/mibs/Cisco
mibdirs +/opt/vdops/share/snmp/mibs/Citrix
mibdirs +/opt/vdops/share/snmp/mibs/Compaq
mibdirs +/opt/vdops/share/snmp/mibs/Crossroads
mibdirs +/opt/vdops/share/snmp/mibs/Cyclades
mibdirs +/opt/vdops/share/snmp/mibs/Dell
mibdirs +/opt/vdops/share/snmp/mibs/DLink
mibdirs +/opt/vdops/share/snmp/mibs/DMTF
mibdirs +/opt/vdops/share/snmp/mibs/DPS
mibdirs +/opt/vdops/share/snmp/mibs/ESI
mibdirs +/opt/vdops/share/snmp/mibs/Enterasys
mibdirs +/opt/vdops/share/snmp/mibs/ESI
mibdirs +/opt/vdops/share/snmp/mibs/Extreme
mibdirs +/opt/vdops/share/snmp/mibs/FibreChannel
#mibdirs +/opt/vdops/share/snmp/mibs/Fore
```

```
mibdirs +/opt/vdops/share/snmp/mibs/Freeradius
mibdirs +/opt/vdops/share/snmp/mibs/Gnome
mibdirs +/opt/vdops/share/snmp/mibs/HP
mibdirs +/opt/vdops/share/snmp/mibs/IBM-CLEAN
mibdirs +/opt/vdops/share/snmp/mibs/IETF
mibdirs +/opt/vdops/share/snmp/mibs/Intel
mibdirs +/opt/vdops/share/snmp/mibs/Isilon
mibdirs +/opt/vdops/share/snmp/mibs/Juniper
mibdirs +/opt/vdops/share/snmp/mibs/Konica
mibdirs +/opt/vdops/share/snmp/mibs/Lantronix
mibdirs +/opt/vdops/share/snmp/mibs/Lexmark
mibdirs +/opt/vdops/share/snmp/mibs/Liebert
mibdirs +/opt/vdops/share/snmp/mibs/Linksys
mibdirs +/opt/vdops/share/snmp/mibs/Marconi
mibdirs +/opt/vdops/share/snmp/mibs/McData
mibdirs +/opt/vdops/share/snmp/mibs/Microsoft
mibdirs +/opt/vdops/share/snmp/mibs/Minolta
mibdirs +/opt/vdops/share/snmp/mibs/NetApp
mibdirs +/opt/vdops/share/snmp/mibs/NetGear
mibdirs +/opt/vdops/share/snmp/mibs/Netopia
mibdirs +/opt/vdops/share/snmp/mibs/Netscreen
mibdirs +/opt/vdops/share/snmp/mibs/NetSNMP
mibdirs +/opt/vdops/share/snmp/mibs/Nokia
mibdirs +/opt/vdops/share/snmp/mibs/Nortel
mibdirs +/opt/vdops/share/snmp/mibs/Novell
mibdirs +/opt/vdops/share/snmp/mibs/Okidata
mibdirs +/opt/vdops/share/snmp/mibs/Pathlight
mibdirs +/opt/vdops/share/snmp/mibs/PF
mibdirs +/opt/vdops/share/snmp/mibs/Polycom
mibdirs +/opt/vdops/share/snmp/mibs/Radware
mibdirs +/opt/vdops/share/snmp/mibs/Raritan
mibdirs +/opt/vdops/share/snmp/mibs/ServerTech
mibdirs +/opt/vdops/share/snmp/mibs/SMC
mibdirs +/opt/vdops/share/snmp/mibs/SNMPResearch
mibdirs +/opt/vdops/share/snmp/mibs/Sun
mibdirs +/opt/vdops/share/snmp/mibs/Symbios
mibdirs +/opt/vdops/share/snmp/mibs/Tektronix
mibdirs +/opt/vdops/share/snmp/mibs/TippingPoint
mibdirs +/opt/vdops/share/snmp/mibs/Transition
mibdirs +/opt/vdops/share/snmp/mibs/Uptime
mibdirs +/opt/vdops/share/snmp/mibs/Veritas
mibdirs +/opt/vdops/share/snmp/mibs/VMWare
mibdirs +/opt/vdops/share/snmp/mibs/WTCS
mibdirs +/opt/vdops/share/snmp/mibs/Wyse
mibdirs +/opt/vdops/share/snmp/mibs/Xerox
mibdirs +/opt/vdops/share/snmp/mibs/Xyplex
gnat%
```

Verify that your copy of the net-snmp toolkit can find the enterprise-specific MIB files which you just installed by translating an enterprise-specific Object Value into a fully-qualified Object Value and into an OID (aka 'number').

```
gnat% snmptranslate -Of -IR sysBootedImage.0
```

```
.iso.org.dod.internet.private.enterprises.cisco.workgroup.ciscoStackMIB.syste
mGrp.sysBootedImage.0
gnat%  snmptranslate -IR -On sysBootedImage.0
.1.3.6.1.4.1.9.5.1.1.38.0
gnat%
```

Generally, MIB files contain syntax errors, and *snmptranslate* will discuss these with you in glowing detail. Edit the MIB files and fix the errors.

Install the Netops Perl Code

Install the Netops Modules

Acquire Netops-Modules.tar.gz and store it in /opt/vdops/archive.

```
gnat% cd /opt/vdops/src
gnat% tar xvfz ~/Netops-ToolKit.tar.gz
[…]
gnat% cd netops-toolkit-2.2.4
gnat% mkdir –p /opt/vdops/lib/perl5/site_perl/5.12.2/FHCRC/Netops
gnat% chmod –R 770 /opt/vdops/lib/perl5/site_perl/5.12.2/FHCRC
gnat% cp modules/*.pm /opt/vdops/lib/perl5/site_perl/5.12.2/FHCRC/Netops
gnat% ls / opt/vdops/lib/perl5/site_perl/5.12.2/FHCRC/Netops
APCTools.pm    DellTools.pm  MRTGTools.pm    PingTools.pm  Utilities.pm
CiscoTools.pm  HostTools.pm  NetopsData.pm   SNMPTools.pm
DBTools.pm     IFTools.pm    NetopsTools.pm  SwatchOps.pm
gnat%
```

Install the Netops Scripts

Create the 'netops' user. Override the choices for UID ('9999') and Group ('users') as necessary for your site.

```
gnat# useradd -u 9999 -g users -c "Netops Toolkit" -m netops
gnat#
```

Acquire Netops-Scripts.tar.gz and store it in /opt/local/archive.

```
gnat% cd /home/netops
gnat% mkdir etc bin logs rpts tmp
gnat% cd /opt/vdops/src/netops-toolkit-2.2.4
gnat% cp bin/* /home/netops/bin
gnat% chmod 770 /home/netops/bin/*
gnat% ls bin
apc-battery-status                  apc-mod-config
apc-calibrate                       apc-pdu-alarm
apc-chassis-status                  apc-reset-device
apc-hardware                        apc-reset-passwd-alarm
apc-iem-alarm                       apc-save-files
apc-io-status                       apc-self-test
apc-mgmt-card                       apc-seq-calibrate
```

```
apc-set-stuff                       intel-nic-alarm
apc-software                        inv-hardware
apc-upgrade                         inv-image
apc-ups-alarm                       inv-rom
apc-verify-trapping-alarm           liebert-alarm
apc-watch-calibrate                 mass-ping
asa-alarm                           mod-config
auto-save                           mod-interface
bluearc-alarm                       named-if-alarm
broadcom-nic-alarm                  netapp-alarm
brocade-alarm                       netops-alarm-config
build_backplane_config              netops-global-config
build_firewall_config               nokia-alarm
build_fsx_config                    pinger-report
build_gw_call_stats_config          porter-report
build_host_config                   proxy-ping-alarm
build_inline_power_config           qos-drops-alarm
build_ios_config                    red-reboot
build_ips_config                    redundancy-testing
build_octet_config                  reset-device
build_port_buffer_config            save-config
build_port_error_config             seq-reboot
build_remote_access_config          show-cdp-neighbors
build-swatch-config                 show-standby
build_system_power_config           show-tunnels
build_temperature_config            switch-power-alarm
build_ups_config                    tippingpoint-alarm
catalyst-serial-num                 transition-alarm
ccm-alarm                           transition-hardware
chassis-serial-num                  troubled-interface-report
cisco-alarm                         tunnel-alarm
compaq-alarm                        twiddle-interface
compaq-firmware                     twiddle-module
count-cam-table                     uc-version
count-mail-msgs                     unconf-modules-alarm
daily-syslog-extracts               unity-alarm
dell-alarm                          unsnooped-vlan-alarm
dell-firmware                       upgrade-ios
emit-dhcp-discover                  verify-snmp-access-alarm
examine-ips-logs                    wan-circuit-alarm
find-if-problems                    wan-circuit-details
find-qos-drops                      watch-via-arp
find-span-ports                     write-mem
graph-mass-ping

gnat%
```

Verify that you have all the necessary Perl modules installed.

```
gnat% cd /home/netops/bin
gnat% /opt/vdops/bin/perl -c auto-save
Can't locate Mail/Send.pm in @INC (@INC contains:
/opt/local/lib/perl5/5.8.6/i68
```

```
6-linux-thread-multi /opt/local/lib/perl5/5.8.6
/opt/local/lib/perl5/site_perl/5
.8.6/i686-linux-thread-multi /opt/local/lib/perl5/site_perl/5.8.6
/opt/local/lib/perl5/site_perl .) at
/opt/local/lib/perl5/site_perl/FHCRC/Netops/Netops.pm line 51.
BEGIN failed--compilation aborted at
/opt/local/lib/perl5/site_perl/FHCRC/Netops/
Netops.pm line 51.
Compilation failed in require at inv-image line 89.
BEGIN failed--compilation aborted at inv-image line 89.
gnat%
```

Oops, don't have Mail::Send. Ok, install it

```
gnat% cpan
CPAN: File::HomeDir loaded ok (v0.69)

cpan shell -- CPAN exploration and modules installation (v1.9205)
ReadLine support enabled

cpan[1]>  install Mail::Send
[...]
cpan[2]> quit
Lockfile removed.
gnat% perl -c auto-save
Can't locate Perl6/Builtins.pm in @INC (@INC contains:
/opt/vdops/lib/perl5/5.8.
8/x86_64-linux-thread-multi /opt/vdops/lib/perl5/5.12.2
/opt/vdops/lib/perl5/site
_perl/5.12.2/x86_64-linux-thread-multi /opt/vdops/lib/perl5/site_perl/5.12.2
/opt/vdops/lib/perl5/site_perl .) at /home/netops/bin/build_octet_config line
80.
BEGIN failed--compilation aborted at /home/netops/bin/build_octet_config line
80.
```

Oops, don't have Perl6::Builtins. Ok, install it.

```
gnat% cpan
cpan[1]> install Perl6::Builtins
[...]
Failed 1/5 test programs. 2/32 subtests failed.
make: *** [test_dynamic] Error 2
  DCONWAY/Perl6-Builtins-0.0.3.tar.gz
  /usr/bin/make test -- NOT OK
//hint// to see the cpan-testers results for installing this module, try:
  reports DCONWAY/Perl6-Builtins-0.0.3.tar.gz
Running make install
  make test had returned bad status, won't install without force
Failed during this command:
 DCONWAY/Perl6-Builtins-0.0.3.tar.gz          : make_test NO
```

Dang, I don't like to ignore failed tests, but I'm lazy today, so let's push ahead.

```
cpan[2]> force install Perl6::Builtins
Failed during this command:
 DCONWAY/Perl6-Builtins-0.0.3.tar.gz          : make_test FAILED but failure
ignored because 'force' in effect

cpan[3]> quit
Lockfile removed.
gnat% perl -c auto-save
auto-save syntax OK
gnat%
```

Success!

Configure the Netops Scripts

For the most part, we configure the Toolkit using two external configuration files. Their locations are specified in `/opt/vdops/lib/perl5/site_perl/5.12.2/FHCRC/Netops/NetopsData.pm` If you don't like the defaults (below), edit this file and move them.

```
cd /opt/vdops/src/netops-toolkit-2.2.4
cp etc/netops-global-config /home/netops/etc
cp etc/netops-alarm-config /home/netops/etc
```

Edit each file in turn.

netops-global-config

Here is a description of each parameter.

The [all] section applies to all scripts.

;Company strings	
company_tag	Appears in the Subject line of messages emitted by xxxx-alarm scripts
institution	The name of your company; appears in reports
;Miscellaneous	
grab_hosts	The command to use to produce an /etc/hosts formatted list of devices. Only needed if you want to use the -a or -e command-line options
home_networks	The IP space(s) which belong to you; anything else will be considered 'alien', and the Toolkit will automatically exclude devices whose IP addresses live outside the collection of home_networks
skip_name	When parsing the hosts table looking for devices (the -a or -e command-line options), the Toolkit will skip any device whose name contains one of these strings.
snpp-host	Name of the host running a Simple Network Paging Protocol server (typically qpage) – only needed if you want to emit pages

suffixes — We name our devices using a structured naming convention: yale-rtr, for example, is a router, while yale-esx is an Ethernet switch. If you want to use the -a or -e command-line switches, define your suffixes here.

-agw	Analog Gateway
-ap	Access Point (wireless)
-bat	Battery Management System (APC)
-dgw	Digital Gateway
-emu	Environmental Monitoring Unit (APC)
-fsx	Fibre Channel Switch
-hvac	Heating, Ventilating, and Cooling (Liebert)
-ips	Intrusion Prevention Server (TippingPoint)
-pdu	Power Distribution Unit (APC)
-ups	Uninterruptible Power Supply (APC)
-vpn	VPN server (Cisco ASA)
-vpnhw	ASA 5505
-wism	Wireless Services Module
-wlc	Wireless LAN Controller

temperature_units — F or C (Fahrenheit or Celsius)

;Pause values
; Control various aspects of waiting
long
middle
short

;Ping stuff
ping_count
ping_timeout

; Report stuff
report_queries — Shows up in reports

; SNMP stuff
mib_dirs — Where to look for the MIB file collection
mib_files — Which MIB files to load
snmp_max_msg_size
snmp_max_rep
snmp_non_rep
snmp_read — Can be a list, separate using spaces or commas
snmp_translate — Use the net-snmp toolkit's OID to Object Value translation feature
snmp_port
snmp_retries
snmp_timeout — Measured in milliseconds

; Syslog stuff
syslog_facility

syslog_host
syslog_port
syslog_priority
syslog_socket

; TFTP stuff

tftp_dir	Where to save files via tftp: file system semantics
tftp_path	Where to save files via tftp: TFTP semantics
tftp_wait	How long to wait in seconds for a TFTP copy to complete

Those are the global parameters.

Individually, one can override a few of the global parameters on a per script basis (see NetopsTools.pm:read_config for specifics). In general, we override the following:

[apc-battery-status]

alarm_log	To what file does the script append a record of the problems it finds; only relevant to the xxxx-alarm scripts
report_file	Where do we write the report
report_subject	Appears at the top of the report and in the Subject line of e-mail
report_url	Where to find the HTML version of the output; appears in text reports
skip_name	Skip device names containing these strings; useful if you are using the -a or –e command-line options
snmp_read	List of SNMP read-only strings, separated by spaces or commas
snmp_write	List of SNMP write strings, separated by spaces or commas
suffixes	Useful if you are using the –a or –e command-line options

netops-alarm-config

This config file influences the behavior of the xxxx-alarm scripts, which are scripts typically run from cron nightly. They produce a text report, which they write to the file system. If they detect a problem, they also send e-mail. Here is a description of each parameter.

down_for_maintenance	Devices which we know are broken go here, so that we don't have to see the daily e-mail whining about how broken they are. Separate device names using spaces or commas.
owner	The name of the person who owns responsibility for responding to the alarms identified in this report
owner_backup	The name of the person who owns responsibility for responding to the alarms identified in this report if the 'owner' is out of action
report_recipients	List of e-mail addresses, separated by commas or spaces, identifying who wants to receive this report, if an alarm is detected. Minimally, the e-mail addresses of the 'owner' and 'owner_backup'.

The [all] section allows one to populate down_for_maintenance and report_recipients with elements which apply globally, to all alarm scripts. Saves having to type these into each one individually.

Test

Run a Simple Script

Start with a simple script -- I recommend *inv-image* -- and read the comments at the front of the script. In particular, find the 'Define global variables' section and modify as appropriate for your site.

Minimally:

- change the first line to point to the your 'Netops' ready copy of Perl
- change the definition of $institution
- define where you want the $report_file to land
- change @mib_dirs to reflect where you store your MIB files
- change @snmp_read_strings to reflect your read-only community strings
- change @snmp_write_strings to reflect your write community strings
- ignore @skip_name and @suffixes for now ... these are complex knobs

Try running a script, substituting the name of one of your devices for the string "mp4-esx" below:

```
gnat% ./inv-image -s yes mp4-esx
Beginning ./inv-image
Gathering status ...


Checking target list for errors ...
!

Characterizing target list...
!

Acquiring image names ...
!

# Title:           IOS Image Report
#
# Institution:     Widgets, Inc.
#
# Date of Report:  3-2-2005 at 08:34:15
#
# Description:     This report identifies the hardware and software under
#                  switches and routers
#
# Active:          1
#
# Questions:       If you have questions or comments regarding this
#                  report, please mail them to "sysadmin@company.com".
```

```
#
#
# target          hardware              image
# --------------  --------------------  ------------------------------------
mp4-esx           wsc4006               cat4000-k9.7-6-7.bin


Ending ./inv-image
gnat%
```

Repeat this process with each script of interest to you. Minimally, I recommend testing the following (many of the scripts exercise the same underlying functions ... but the scripts in this list tend to rely on unique capabilities and therefore tend to be the ones which break first).

Exercise Complex Scripts

auto-save
Make a change on a device.
`auto-save -s yes test-esx`
Verify that before and after configs appear in /home/netops/logs/router/{date}/{time}

write-mem
`write-mem -s yes test-esx`
Verify that running-config and startup-config are the same (that your change is now reflected in startup-config)

save-config
Change directories to /home/netops/logs/router/{today's date}/{current time}
`save-config -s yes test-esx`
save-config should perform a 'wr mem' on test-esx and then save before and after copies of startup-config in this directory. Do *not* run save-config from the root level of your home directory (this script messes with current directory permissions) ... if you must run it from somewhere near your home directory, dig yourself a hole first:

```
mkdir -p ~/temp/testing
cd ~/temp/testing
save-config -s yes test-esx
```

mod-config
Create a config file snippet in your tftpboot directory and upload it to test-esx, to undo whatever change you made.
`mod-config -s yes -c snippet test-esx`
Perform a 'wr mem' on the device to save your changes.

apc-save-files
Save the current APC config files to your tftp server.
`apc-save-files -s yes -u {username} -p {password} -t config -a`
Verify that the config files stored on your tftp server have been updated.

seq-reboot

Reboot a couple of devices in sequence.
```
seq-reboot -s yes test1-esx test2-esx
```
Use your own continuous pings to verify that in fact the devices rebooted.

twiddle-interface
Toggle link on an interface.
```
twiddle-interface -s yes -i Gi2/23 -a toggle -h test-esx
```
Watch syslog to verify that in fact link goes down and then comes back up again.

upgrade-ios
Upgrade the image on a device.
```
upgrade-ios -s yes -i cat4500-ipbasek9-mz.122-46.SG.bin test-esx
```

Use Help

```
Consult 'help':

gnat% ./inv-image --version
inv-image v1.7.4
gnat%

gnat% ./inv-image --help
[...]


Typical command-line switches:

'-s' is the "are you serious" flag ... all the scripts require
either 'yes' or 'no' ... 'yes' means "go ahead and do it", while
'no' means "pretend to do it ... but don't actually make any
changes".  The 'no' flag is a good way to exercise much of a
script's functionality without actually doing anything ... a script
may emit SNMP GET requests and ICMP Echos while running under 'no',
but it will never emit an SNMP SET or otherwise change the targets.

'-r' tells the script to write the report to a file, rather than to STDOUT

'-a' tells the script to parse the hosts table (see the definition
of $grab_hosts) and to extract a list of all hosts terminating in
the strings defined in @suffixes.  And to use this list as the
list of targets on which to perform work.

'-e {regex}' tells the script to use that regex to parse the hosts
table, extracting a list of all the hosts which match.  And to use
this list as the list of targets on which to perform work.

'-f {filename}' tells the script to take the following string and to
treat that string as the name of a file to read and to build its target
list from the contents of that file.


Alternatively, if you don't want the script parsing the hosts table
```

```
for suffixes (-a) or parsing the hosts table using a regex (-e expr)
or to read a file (-f filename), you can simply type the names of
the targets, separated by spaces, on the command line, e.g.

gnat% ./inv-image -s yes device1 device2 device3 device4 device5 ...
```

Use Debug

Add the "-d {integer}" flag to invoke debugging. Debug levels typically range from 0 (no debugging) to 9 (maximum debugging). Start off with a debug level of one and increment until you find a sufficiently verbose error message to give you a clue as to why the script isn't behaving the way you would expect it to behave. In general, '4' is about as verbose as you'll want to get.

```
gnat% ./inv-image -s yes –d 1 device1
```

Notes

I've only barely started thinking about Windows compatibility ... if you're feeling brave and want to give this a shot, drop me a line.

Stuart Kendrick
www.skendric.com
sbk {insert @ sign here} skendric.com

[If I don't answer an e-mail query, try digging up my work e-mail address from www.fhcrc.org and sending me e-mail there ... the spam filters on the *skendric.com* account are fierce.]