Notes on Perl

- General Notes
- Perl on Windows
- Perl on VMS
- Perl on NonStop
- Required Perl modules
- Notes on installing a Perl module

General Notes

For our scripts, we rely quite a bit on Perl, and increasingly on some core Perl modules. These Perl modules are part of the Perl source, so if you build Perl on your own, you should be set.

However, if you install Perl as binary packages, the outcome might differ, and you may have to check that you do get the core modules installed properly. We do not claim to know them all, but experience has told us the following:

- on Linux distributions based on Debian, the package `perl` will install the core Perl modules as well, so you will be fine.
- on Linux distributions based on RPMs, you will need to install `perl-core` rather than just `perl`.

You MUST have at least Perl version 5.10.0 installed. This minimum requirement is due to our use of regexp backslash sequence \R among other features that didn't exist in core Perl before that version.

Perl on Windows

There are a number of build targets that can be viewed as "Windows". Indeed, there are `VC-*` configs targeting VisualStudio C, as well as MinGW and Cygwin. The key recommendation is to use a Perl installation that matches the build environment. For example, if you will build on Cygwin be sure to use the Cygwin package manager to install Perl. For MSYS builds use the MSYS provided Perl. For VC-* builds we recommend Strawberry Perl, from http://strawberryperl.com. An alternative is ActiveState Perl, from http://www.activestate.com/ActivePerl for which you may need to explicitly select the Perl module Win32/Console.pm available via https://platform.activestate.com/ActiveState.

Perl on VMS

You will need to install Perl separately. One way to do so is to download the source from http://perl.org/, unpacking it, reading `README-VMS.md` and follow the instructions. Another way is to download a `.PCSI` file from http://www.vmsperl.com/ and install it using the POLYCENTER install tool.

Perl on NonStop

Perl is installed on HPE NonStop platforms as part of the Scripting Languages package T1203PAX file. The package is shipped as part of a NonStop RVU (Release Version Updates) package. Individual SPRs (Software Product Release) representing fixes can be obtained from the Scout website at https://h22204.www2.hpe.com/NEP. Follow the appropriate set of installation instructions for your operating system release as described in the Script Language User Guide available from the NonStop Technical Library.

Required Perl modules

We do our best to limit ourselves to core Perl modules to keep the requirements down. There are just a few exceptions.

- Text::Template this is required *for building*

  To avoid unnecessary initial hurdles, we include a copy of this module in the source. It will work as a fallback if the module isn't already installed.

- `Test::More` this is required *for testing*

  We require the minimum version to be 0.96, which appeared in Perl 5.13.4, because that version was the first to have all the features we're using. This module is required for testing only! If you don't plan on running the tests, you don't need to bother with this one.

Notes on installing a Perl module

There are a number of ways to install a perl module. In all descriptions below, `Text::Template` will serve as an example.

1. for Linux users, the easiest is to install with the use of your favorite package manager. Usually, all you need to do is search for the module name and to install the package that comes up.

   On Debian based Linux distributions, it would go like this:

   ```
   $ apt-cache search Text::Template
   ...
   libtext-template-perl - perl module to process text templates
   $ sudo apt-get install libtext-template-perl
   ```

   Perl modules in Debian based distributions use package names like the name of the module in question, with "lib" prepended and "-perl" appended.

2. Install using CPAN. This is very easy, but usually requires root access:

   ```
   $ cpan -i Text::Template
   ```

   Note that this runs all the tests that the module to be installed comes with. This is usually a smooth operation, but there are platforms where a failure is indicated even though the actual tests were successful. Should that happen, you can force an installation regardless (that should be safe since you've already seen the tests succeed!):

   ```
   $ cpan -f -i Text::Template
   ```

   Note: on VMS, you must quote any argument that contains upper case characters, so the lines above would be:

   ```
   $ cpan -i "Text::Template"
   ```

   and:

   ```
   $ cpan -f -i "Text::Template"
   ```