.profile and user setup

(the keys to the kingdom)

Michel Valin
CRB / HPCS

General user setup

- The basic components
  - shells and management software
    - supported login shells (/bin/sh, /bin/ksh, /bin/bash)
    - 100% userland software package management tool (SSM) Simple Software Manager
  - user configuration setup and customization files
    - .profile (soft link to a file)
    - .profile.d/ (directory)
    - Ecssm (soft link to a directory)
- Provides user with
  - Basic tools for further software configuration
  - TMPDIR / BIGTMPDIR environment variables
  - Cleanup at entry/exit of session / job for temporary directories

The profile setup elements

- .profile
  - should be a soft link to ECssm/multi/bin/.profile
- ECssm
  - soft link to setup domain
  - /home/ordenv/ssm-domains/ssm-setup (for now)
- .profile.d
  - directory containing the user's setup customization files
  - automatically created at first login with this .profile setup
  - custom setup files come in 4 flavours
    - plain (.xxxx_profile)
    - plain.OS (.xxxx_profile.Linux)
    - plain.processor (.xxxx_profile.i686)
    - plain.OS.processor (.xxxx_profile.Linux.i686)

The .profile.d directory contents

- pre profile
  - .pre_profile[.OS][.cputype]
  - .pre_profile[.${TRUE_HOST}]
- group profiles
  - .group[0123]_profile[.OS][.cputype]
  - .group[0123]_profile[.${TRUE_HOST}]
- user profile
  - .user_profile[.OS][.cputype]
  - .user_profile[.${TRUE_HOST}]
- host profile (or cluster profile)
  - .${TRUE_HOST}_profile

The .profile.d directory contents

- interactive profile
  - .interactive_profile[.OS][.cputype]
  - .interactive_profile[.${TRUE_HOST}]
- XWindows profile
  - .Xwindows_profile[.OS][.cputype]
  - .Xwindows_profile[.${TRUE_HOST}]

The .profile.d directory contents

- ssh profile
  - .ssh_profile[.OS][.cputype]
  - .ssh_profile[.${TRUE_HOST}]
- batch profile
  - .batch_profile[.OS][.cputype]
  - .batch_profile[.${TRUE_HOST}]
- OS = uname -s
- cputype = uname -m (Linux) or uname -p (AIX)

The pre profile (s)

- Sourced immediately after base ssm domain setup
- Contains control elements for the rest of the setup
  - Environment variables
    - BATCH_LIKE_INTERACTIVE (active if non null)
    - SETUP_MSG (verbosity level, 0 to 10)

The group profile(s)

- Sourced
  - After the pre profile
  - Before the user/interactive profile
  - Only for interactive sessions
  - Unless the user defined BATCH_LIKE_INTERACTIVE in which case they are also sourced for batch jobs

The user profile (s)

- Sourced
  - After the group profile(s)
  - Before the batch/interactive profile
  - Only for interactive sessions
  - Unless the user defined BATCH_LIKE_INTERACTIVE in which case they are also sourced for batch jobs

The batch profile (s)

- Sourced
  - After the pre profile
  - Only for batch jobs (tty -s is false)
  - Unless the user defined BATCH_LIKE_INTERACTIVE in which case it is not sourced and the group/user profiles are sourced instead

The ssh profile(s)

- Sourced
  - After the pre profile
  - Only for ssh remote execution (tty -s is false)
  - Nothing else is sourced in this case

The interactive profile (s)

- Sourced after group and user profiles
- Only if the session is interactive (tty -s is true)

The XWindows profile (s)

- Sourced as the last step of .profile processing after group, interactive(if tty -s is true) and user profiles
- Only if the session is XWindows (${DISPLAY} is defined) (at window manager login, tty -s is false but DISPLAY is defined)

switching profile at login / start of job

- .profile - > .profile_switch
- .profile_switch - > ECssm/multi/bin/.profile_switch
- .profile_default - > profile by default
- .profile_alternate - > alternative profile

- There will be a question at login time
  use alternate .profile ? y/n/name [n]
  if no answer within 3 seconds, n will be the answer
  answer may be y / n / somename
  if answer is somename, .profile_somename will be used
  if answer is n, .profile_default will be used
  if answer is y, .profile_alternate will be used

switching profile at login / start of job

- Example

  .profile -> .profile_switch
  .profile_alternate -> /usr/local/env/env_univ/005/.profile
  .profile_default -> ECssm/multi/bin/.profile
  .profile_switch -> Ecssm/multi/bin/.profile_switch
  .profile_plus -> /usr/local/env/env_univ/005/.profile+

- Possible answers would be
  - y
  - n
  - plus
  - default
  - alternate
  - switch

What you get for the price

- basic environment
  - access to basic utilities and environment variables
  - a list of possible config file names (.profile.d/NAMES.list)
  - useful environment variables
    - TMPDIR (private for the session/job)
    - EC_ARCH (full architecture marker)
    - etc ...
- basic utilities
  - ssm (THE package management tool)
  - s.ssmuse.dot / r.ssmuse.dot
    (add ssm managed software components to the available pool)
  - Soumet / ord_soumet (submit a batch job to any of the site's batch capable machines / clusters)

What you get for the price

- basic utilities
  - true_path (get the fully resolved path of files after all soft links have been resolved)
  - r.which/s.which
    (where does this executable / dotted file can / will come from ?)
  - s.echo
    (verbosity controlled messages to stderr)
  - cclargs
    (THE script argument parser)
  - s.kwik-parse
    (a quick gnu like script argument parser)

ssm

- ssm basic terms (in a very small nutshell)
  - domain
    - a collection of packages
      - installed
      - published (available when user subscribes to the domain)
    - can be subscribed to by a user
    - can be registered as a subdomain of another domain
      - subscribing to a domain automatically subscribes to all registered subdomains
  - package (a collection of related software elements)
    - a model
    - a set of graphical utilities
    - a compiler version
  - repository
    - where packages are stored

s.which/r.which

averroes 500% r.which -h

Usage: s.which [-a] [-v] [-d] [-x] [-l] [-e] args

args may contain shell file pattern(s)

-l : ls -l matching item(s)

-e : view matching item(s)

-v : verbose mode

-d : only print dottable files

-x : only print executable files

-a : print all occurrences of each arg (default is first occurrence only)

true_path

```
averroes 511% true_path -h
Usage: true_path [-n] path
     output on stdout will be the true_path with links resolved
Usage: true_path [-n] path1 .... pathn
     output on stderr will be the true_paths with links resolved
averroes 512%
```

s.ssmuse.dot (subscribe to a domain and more)

- Syntax is
  - . s.ssmuse.dot path_to_domain (plain subscribe operation)
  - . s.ssmuse.dot shortcut_name (this may do more)
- Examples
  - . s.ssmuse.dot /home/ordenv/ssm-domains0/ssm-sun12
  - . s.ssmuse.dot sun12
- Shortcut types (name can be dir0/dir1/...../dirn/name)
  - name - > path_to_a_domain (soft link to a domain)
  - name.sh (a script)
  - name.bndl (a file containing a list of shortcuts on 1 line)
- User's own shortcuts from
  - $HOME/my_ssm_domains

s.ssmuse.dot shortcuts (051)

- Shortcut examples
  - Direct link

    averroes 534% ls -al cmd?
    lrwxrwxrwx 1 asphlib hpcs 42 Feb 3 23:30 cmda -> /home/cmss/afsm/adm/ssm_access_points/cmda
    lrwxrwxrwx 1 asphlib hpcs 42 Feb 3 23:30 cmdd -> /home/cmss/afsm/adm/ssm_access_points/cmdd
    lrwxrwxrwx 1 asphlib hpcs 42 Feb 3 23:30 cmds -> /home/cmss/afsm/adm/ssm_access_points/cmds
    lrwxrwxrwx 1 asphlib hpcs 42 Feb 3 23:30 cmdw -> /home/cmss/afsm/adm/ssm_access_points/cmdw

    . s.ssmuse.dot cmdw
  - Script

    averroes 535% cat ovbin.sh
    export PATH=$HOME/ovbin/$EC_ARCH:$HOME/ovbin:$PATH:$HOME/bin/$EC_ARCH:$HOME/bin

    . s.ssmuse.dot ovbin
  - Bundle (a collection of names)

    averroes 518% cat cmd_all.bndl
    cmda cmdd cmds cmdw

    . s.ssmuse.dot cmd_all

s.ssmuse.dot shortcuts (101)

- group + personal shortcuts
- in $HOME/my_ssm_domains
  - bd1.bndl
  - kwik.sh
  - my_group - > path_to_my_group_shortcuts
- in path_to_my_group_shortcuts
  - gbd1.bndl
  - dom1 - > path_to_ssm_domain_1
  - dom2 - > path_to_ssm_domain_2
  - machin.sh
- . s.ssmuse.dot bd1 (my own shortcuts)
- . s.ssmuse.dot my_group/dom1 (my group's shortcuts)

s.ssmuse.dot shortcuts (201)

- Get list of available shortcuts
  s.list_ssm_shortcuts

s.list_ssm_shortcuts -h
usage: s.list_ssm_shortcuts [--help] [--subset=xxx] [--filter=file_pattern] [--doc]
    --subset= subdirectory of interest
    --filter= grep style pattern used to filter out items
    --doc lists the documentation associated with item (ssm listd if a domain)

```
s.list_ssm_shortcuts --subset=MPI
FROM: . . . /ssm-setup-1.0/dot-profile-setup_1.0_multi/notshared/data/ssm_domains/MPI/
MPI/mpi6shared       MPI/mpich2-121       MPI/mpich2-131
MPI/mpich2-verbs-131 MPI/openmpi
```

```
s.list_ssm_shortcuts --filter=Xlf12
FROM: /users/dor/armn/mfv/my_ssm_domains/

FROM: . . . /ssm-setup-1.0/dot-profile-setup_1.0_multi/notshared/data/ssm_domains/
Xlf12.1              Xlf12.104            Xlf12.107
Xlf12.110            Xlf12.111            Xlf12
```

s.ssmuse.dot shortcuts (201)

- Get list of available shortcuts
  s.list_ssm_shortcuts

```
s.list_ssm_shortcuts --filter=Xlf12$ --doc
FROM: /users/dor/armn/mfv/my_ssm_domains/
FROM:  . . . /ssm-setup-1.0/dot-profile-setup_1.0_multi/notshared/data/ssm_domains/
=== Xlf12 ===
## xlf version 12.1.0.4
```

```
s.list_ssm_shortcuts --filter=sms --doc
FROM: /users/dor/armn/mfv/my_ssm_domains/
FROM:  . . . /ssm-setup-1.0/dot-profile-setup_1.0_multi/notshared/data/ssm_domains/
=== Domain /home/dormrb02/ssm-SMS ===
State     Package Name                        Title                          Domain
-----     ------------                        -----                          ------
installed sms_4.4.5_irix65-mips-n32           Supervisor Monitor Scheduler   sms
installed sms_4.4.5_linux26-i386              Supervisor Monitor Scheduler   sms
```

ord_soumet

- Integrated tool, supports OCM, Maestro, straight batch. (ord stands for Operations/Research/Development)
- Swiss army knife utility to submit a batch job to any batch capable Dorval system (LoadLeveler, SunGridEngine, ...)
  - IBM clusters (LoadLeveler saiph/zeta)
  - Linux clusters (SGE dorval-ib/alef)
  - Special systems (SGE idl, …)
  - Workstations (SGE)
  - Pseudo hosts (allows redirection, e.g. erg->alef)
- https://wiki.cmc.ec.gc.ca/wiki/Soumet_:_travaux_par_lots_/_batch_jobs
- Smarter and meaner successor to the long lived soumet (now capable of standalone operation)

ord_soumet job_file [options]

- some popular options
  - -mach machine/cluster/pseudo machine job will be sent to
  - -cm amount of memory needed for job
  - -t WALL CLOCK time needed for job
  - -jn job name (default is job_file)
  - -cpus cpu configuration needed (e.g. 2x3x4)
  - -mpi this job uses MPI
  - -nosubmit (prepare job, put it into lajob.tar, do not submit)
  - -q use a specific batch queue
  - -xterm open an xterm session from the job when running (will target $DISPLAY)
  - -v verbose operation
  - -immediate run job in immediate mode (ssh launch and forget)