CONFIGS

A host configuration facility

by

Simon J. Gerraty <sjg@zen.void.oz.au>

Contents

Configuration management

The ability to easily reproduce a host configuration, facilitates:

- System installation.
- System cloning.
- Disaster recovery.

config.sh

CONFIGS/`config.sh` processes *CONFIGS*/`config.d`
It handles:

- Selecting directories within *CONFIGS* to search (*DIR_LIST*).
- Installs files.
- Installs archives (tar or cpio).
- Installs vendor patches (Solaris).
- Removes files.
- Sets permissions.
- Search *DIR_LIST* for other config.d's and process them.
- Reboot system if needed.

CONFIGS Tree

The *CONFIGS* tree supports multiple operating systems, releases and hardware types. The following directories are added to DIR_LIST if they exist:

CONFIGS/SITE/HOST

CONFIGS/SITE/OS/OSREL/MACHINE

CONFIGS/SITE/OS/OSREL

CONFIGS/SITE/OS/OSMAJOR.`X`/*MACHINE*

CONFIGS/SITE/OS/OSMAJOR.`X`

CONFIGS/SITE/OS/MACHINE

CONFIGS/SITE/OS

CONFIGS/OS/OSREL/MACHINE

CONFIGS/OS/OSREL

CONFIGS/OS/OSMAJOR.`X`/*MACHINE*

CONFIGS/OS/OSMAJOR.`X`

CONFIGS/OS/MACHINE

CONFIGS/OS

CONFIGS/SITE

CONFIGS

Example

The following are *some* of the directories in the author's *CONFIGS* tree:

```
configs/NetBSD/etc/daily.d
configs/NetBSD/etc/rc.d
configs/NetBSD/etc/rc_local.d
configs/NetBSD/i386/usr/local/bin
configs/NetBSD/sparc/usr/local/bin
configs/SunOS/4.1.3_U1/sun4c
configs/SunOS/4.1.4/sun4c
configs/SunOS/4.X/config.d
configs/SunOS/4.X/etc/hourly.d
configs/SunOS/4.X/sun4m/usr/local/bin
configs/SunOS/4.0/usr/share/mk
configs/SunOS/4.X/usr/share/mk
configs/SunOS/5.X/config.d
configs/SunOS/5.X/etc
configs/SunOS/5.X/sun4m/usr/local/bin
configs/Void/zen/etc/rc_d
configs/Void/zen/usr/local/bin
configs/bin
configs/config.d
configs/etc/rc_d
configs/etc/rc_local.d
configs/usr/local/share/bin
configs/usr/share/mk
```

Install lists

Most of the work of `config.sh` is done by *CONFIGS*/`config.d/install_files` which processes install lists that contain entries like:

```
# $Header: /local/src/master/configs/install_files.list,v \
 1.5 1995/09/10 06:40:57 sjg Exp $
/.profile 0500 root wheel file no
/.kshrc 0755 root wheel file no
/etc/myname  0644 root wheel file no
/etc/aliases 0644 root wheel file no
/etc/exports 0664 root wheel file no
/etc/named.boot 0644 root wheel file yes
/etc/resolv.conf 0444 root bin file yes
/etc/ksh.kshrc 0444 root wheel file no
/etc/ksh.cdhist 0444 root wheel file no
/etc/ntp.conf 0644 root sys file yes
/etc/profile 0444 root wheel file no
/etc/profile.site 0644 root wheel file no
/etc/add_path.sh 0555 root wheel file no
/etc/csh.cshrc 0444 root wheel file no
/etc/csh.login 0444 root wheel file no
/etc/csh.login.site 0644 root wheel file no
/etc/rc.sh 00555 root wheel file no
/etc/rc_d/os.sh 00444 root wheel file no
...
/etc/rc_local.sh 00775 root wheel symlink -> rc.sh
...
/etc/daily.sh 00775 root wheel symlink -> rc.sh
...
/ 0755 root wheel cpio from /sm8-cfg.C.Z
/usr/local/mail/lib/xaliases 0444 root wheel file no
# this makes sure that make is run in /usr/local/mail/lib
/usr/local/mail/lib/genericfrom 0444 root wheel touch make
...
/var/msgs/bounds 0664 root daemon touch \
 shell test -s $path || rm -f $path
```

Usage

The *CONFIGS* tree can be shared via NFS (exported read-only). Information gleaned from the host running **config.sh** is used to build *DIR_LIST*. For example:

```
not# uname -a
NetBSD not.void.oz.au 1.0A.95.08.18
not# mount -r zen:/d2 /d2
not# /d2/configs/config.sh
Using: CONFIGS=/d2/configs, SITE=local, PATCHES=, \
 DIR_LIST=local/not local/NetBSD NetBSD/i386 NetBSD \
 local /d2/configs
Checking /etc/rc_local.d/S20pcmcia_off - OK
Checking /etc/neton - OK
Checking /etc/netoff - OK
Checking /etc/net.d/S60nfs_mounts - OK
...
```

Examples

If run with the argument **show** `config.sh` will simply report what would be done.

```
rex# uname -a
SunOS rex 4.0  sun
rex# /zen/d2/configs/config.sh show
Using: CONFIGS=/zen/d2/configs, SITE=local, PATCHES=, \
 DIR_LIST=local/rex SunOS/4.0 SunOS/4.X SunOS/sun386 SunOS \
 local /zen/d2/configs
Checking /usr/local/man/man1/ksh.1 - New [From \
 /zen/d2/configs/usr/local/man/man1/ksh.1]
...
```

Examples

The script `update_file` is a front-end to `config.sh` which can be used to force only a few files to be checked and/or updated.

```
zen# /d2/configs/update_file funcs fix_hom
Using: CONFIGS=/d2/configs, SITE=local, PATCHES=, \
 DIR_LIST=local/zen local/NetBSD NetBSD/i386 NetBSD \
 local /d2/configs
Checking /etc/rc_d/funcs.sh - Updated [From /d2/configs\
 /etc/rc_d/funcs.sh]
Checking /etc/rc_d/fix_homes - New [From /d2/configs\
 /etc/rc_d/fix_homes]
zen#
```

A paper (**configs.ps**) describing this tool in more detail as well as the tool (**config-sh.tar.gz**) and a sample config tree (**configs-example.tar.gz**) are available from:

ftp://ftp.quick.com.au/pub/unix/

Which is mirrored at:

ftp://ftp.telstra.com.au/mirror/ftp.quick.com.au/unix/

The example tree is just that, an example. It does however contain many useful shell scripts.

Questions: **<support@quick.com.au>**