CRUX 3.4 Commonwealth Manual

1. Preface and Introduction

This manual documents the Commonwealth Extension[1] (specifically for instance `CCIA_0.0`) of the GNU/Linux distribution named **CRUX**. Both CRUX and the Extension maintain the KISS (Keep It Simple, Stupid) design concept as the primary design goal. While these days that can seem highly unorthodox for a GNU/Linux distribution, it **is** after all the core of the UNIX Philosophy as described by Eric S. Raymond in *The Art of UNIX Programming* (specifically *The Unix Philosophy in One Lesson* chapter).

A highly significant derived design goal for CRUX has emerged in recent years—`systemd` has no place in a CRUX *system*[2]. That amazingly popular alternative to the idea of the simple ***`init`*** dæmon, a feature of UNIX from its early years (the relatively new `sysvinit` package originated ~1980), is staunchly considered egregiously flaunting of the UNIX Philosophy by CRUX developers. See the other chapters of Raymond's treatise for the important details.

The KISS principle has been applied to CRUX software maintenance from its inception. For instance, CRUX does not repackage or even build the kernel. Except for providing an `ISO` kernel with its source and `.config` file (see Section **6.1** for details), CRUX *ignores* kernel maintenance, leaving it to the sysadmin to decide what to do using the kernel's `README` file and `kernel.org` announcements.

The Extension adds much complexity to a CRUX infrastructure but as KISSfully as possible. It is likely not for everyone but then it is entirely optional.

1.1. What Is CRUX and Why Use It? (from the CRUX 3.4 Handbook)

CRUX is a lightweight Linux distribution for the x86-64 architecture targeted at experienced Linux users. The primary focus of this distribution is "keep it simple", which it reflects in a simple `tar.gz`-based package system, BSD-style initscripts, and a relatively small collection of trimmed packages. The secondary focus is utilization of new Linux features and recent tools and libraries. CRUX also has a ports system which makes it easy to install and upgrade applications.

There are many Linux distributions out there these days, so what makes CRUX any better than the others? The choice of distribution is a matter of taste, really. Here are a few hints about the tastes and goals of the people behind CRUX. CRUX is made with simplicity in mind from beginning to end. Making it easy to create new and update old packages is essential; updating a package in CRUX is often just a matter of typing ***`pkgmk -d -u`***. The usage of ports helps keep your packages up-to-date; not the latest bleeding-edge-alpha version, but the latest stable version. Other features include creating packages optimized for your processor; e.g., by compiling with `-march=x86-64`, and avoiding cluttering the filesystem with files you'll never use like `/usr/doc/*`, etc. If you need more information about a specific program, other than information found in the man-page, Google usually knows all about it. Finally, it strives to use new features as they become available, as long as they are consistent with the rest of the goals.

In short, CRUX might suit you very well if you are:

- A somewhat experienced Linux user who wants a clean and solid Linux distribution as the foundation of your installation.
- A person who prefers editing configuration files with an editor to using a GUI.
- Someone who does not hesitate to download and compile programs from the source.

1.2. What is the Commonwealth Extension of CRUX?

The canonical CRUX distribution is not intended for widespread deployment within an organization, as part of its simplicity requires each CRUX system to effectively be installed and maintained in isolation from all other organization CRUX systems. In other words, canonical CRUX system administration is not designed to scale out of the box. The Commonwealth Extension to CRUX was developed with the goal of providing "industrial-strength" system software maintenance for CRUX to make feasible wide-scale deployment and maintenance of CRUX platforms within an organization. However, it was equally important to not change canonical CRUX in the process; in particular, the enhancement supports the same package ports and associated infrastructure of canonical CRUX. The `prtpkg` architecture introduces new concepts to canonical CRUX, especially *commonwealth*—a group of CRUX systems that have a shared `prtpkg` software maintenance infrastructure. Refer to Section **3** for more detail about the Extension.

1.3. Why Use the Commonwealth Extension of CRUX?

- You are truly a GNU/Linux sysadmin rockstar. If you have not yet tried using canonical CRUX, you are well-advised to defer diving into the Extension until you are very informed about and comfortable with the CRUX way of doing GNU/Linux.
- You really like what CRUX could do for your enterprise and want to make it a permanent component of your IT infrastructure, but you need to reduce the sysadmin workload many canonical CRUX platforms will require—congratulations! This Extension was developed for people like you.

- You would like to use the Extension to better track the software and files installed on your CRUX system(s). Take a look at what prtpkg can do beyond what the canonical package handling tools can provide even when there is only one computer in your commonwealth.
- You are not intimidated by the additional effort working with prtpkg will require:
    - Comprehending the body of prtpkg documentation (mostly within the document you are reading),
    - Studying and possibly modifying the software's ***bash***, ***dash***, and ***gawk*** code,
    - Planning your organization's commonwealth(s); e.g., disaster site, and
    - Designing, implementing, and documenting the CRUX maintenance policies your enterprise will need; e.g., business continuity plans.
- You are certain the possible benefits of using prtpkg justify at least an experience-garnering pilot project.

2. Historical Software Maintenance in CRUX

This section entirely deals with canonical CRUX—the Extension uses everything described herein without any modifications. Using the prtpkg software requires a solid grounding in this section’s information. Actually, that is true for performing canonical CRUX software maintenance as well.

2.1. In The Beginning, pkgutils

In the early days of CRUX, founding developer Per Lidén created the heart of CRUX software maintenance, the pkgutils package that facilitates automating the standard procedure for introducing FOSS packages into a system:

1. *download* the package source code tarball(s),
2. *extract* the tarball content into a usually temporary build (aka work) directory,
3. *configure* the build directory for building usually via a ***configure*** script,
4. *build* (compile and link, etc.) the package using a ***make*** command, and
5. *install* the built results into the system, usually using a ***make install*** command.

This final step must be performed by the *superuser* who is responsible for the availability and integrity of the platform, and therefore has the highest authorities of the platform in order to ensure said availability and integrity.

With `pkgutils`, the entirety of the standard procedure is within the purview of the ***pkgmk*** command with a twist: the final phase does not "install" the package into the system doing the building; i.e., the root filesystem *tree*[3]. Instead, an alternative tree is defined, usually the build directory itself, via a `DESTDIR` parameter supported by the ***make install*** configuration for the package. The final task of ***pkgmk*** processing, which is outside the standard procedure, is to create a CRUX *package file* (aka *package tarball*) that is intended to be extracted into the real root tree(s) of the target installation system(s) by the superuser. A security advantage of this approach to ***make install*** is that full superuser authority is not necessary. Instead, the `fakeroot` package can be used to define privileged *inode* metadata in a tar, not in the logical root filesystem, without allowing any superuser authority. CRUX encourages using ***fakeroot*** with a no-login user for *everything* ***pkgmk*** does. A foundational concept of `pkgutils` maintenance, then, is building binary executables from source must be separate from installing them into a system and that the build results should be encapsulated in a binary-code package file that can be copied to and installed into one or more suitable CRUX platforms without requiring the source first be built on each platform.

A CRUX package (also called a *port*) is a tree branch developed and maintained by a CRUX developer. Its trunk directory is called the *package directory* (also called a port on occasion). A CRUX package contains all the package-specific data ***pkgmk*** needs to do its job except the actual source tarballs involved (but does contain all the data needed to obtain the source tarballs if necessary). The port's `Pkgfile` text file is central. In fact it is sourcable ***bash*** code (not directly executable) that contains a ***build()*** function (that is invoked by ***pkgmk***, itself a fully executable ***bash*** script) to perform steps 3–5 of the standard procedure (not including the actual installation into a platform as previously discussed).

The ***pkgadd*** command of the `pkgutils` package is run to install/update CRUX package files into the running or mounted target system, which of course is only permitted for that system's superuser. Uninstalling CRUX packages and their installed inodes is the job of the ***pkgrm*** command. Both programs manipulate the flat file `/var/lib/pkg/db` that contains the inventory of packages installed on that system, including their versions and all files installed on their behalf. Use ***rejmerge*** to interactively resolve any file update rejections encountered by ***pkgadd*** invocations.

Note the `pkgutils` methodology remains fundamental to CRUX software maintenance, and has not changed dramatically for almost two decades. However, it is completely ambivalent regarding just what it is maintaining. ***Pkgmk*** learns nothing outside the package directory in which it is invoked save its configuration as defined via command parameters and a configuration file. It does not even care to know the path of the package directory. CRUX developers still highly esteem `pkgutils`' aloofness from package metadata and its abhorrence of activity logging. Thus, `pkgutils` is of next to no use when it comes to distinguishing multiple ports of any particular package from each other and selecting the ports to be processed by `pkgutils`.

2.2. Supporting Different Package Configurations: ports, prt-get

CRUX had grown to the point of organizing package ports with similar characteristics together into directories of ports called *collections*. The officially supported `core`, `opt`, `compat-32`, `xorg`, and quasi-supported `contrib` collections are simply divisions of the packages considered canonical to CRUX—there is only one port per package in this set of collections.

The ***`ports`*** command (and package) was introduced by Per Lidén to assist with maintenance of entire collections. Its biggest feature is provided via the ***`-u`*** flag, which causes updating the specific (or by default all activated) local mirror collections with any changes made to the master collections available via the Internet. Thus, collection maintenance in CRUX is undertaken on a pull basis. Also, the ***`rsync`*** model is used, so local collection mirrors have ports deleted when their Internet counterparts are deleted. Consequently, only the known current version of any port is ever available in any particular collection, or not—mirrored collections alone can contain back-level or dropped ports, but only as long as they remain unsynchronized with their masters.

However, as expected, folks who installed CRUX on their computers often found they had a need to modify the canonical port of a package in order to accommodate capability requirements incongruous with CRUX canon. The solution was and is to fork the canonical port by copying it (***`cp -a`***) into a *private collection*, modifying its `Pkgfile` and anything else in the port as needed, and using the `pkgutils` commands on that port instead of the canonical port. Private collections were (and are) named according to the sysadmin's fancy as long as they didn't/don't conflict with any other collection's name, and these began to be shared with other CRUXers via public *repositories* (just another word for collections). Port management (upstream packages are not static and neither are their ports) was increasingly a chore for everybody.

In response to that itch, Johannes Winkelmann scratched out the next layer of CRUX software management infrastructure from inspiration based in the Debian distribution's `apt-get` facility. The ***`prt-get`*** command uses a configuration file of important options to control its processing, especially the `prtdir` statements that (1) describe what local collections the tool knows about and (2) define how one of many available ports for a particular package is selected for processing. According to the specific ***prt-get*** subcommand the maintainer issues, the utility can invoke ***pkgmk***, ***pkgadd***, and/or ***pkgrm*** as needed to bring about that ***prt-get*** subcommand's objective. In many cases, `pkgutils` commands were no longer used directly to routinely perform **CRUX** software maintenance.

The ***prt-get*** subcommands ***sysup***, ***depinst***, and ***grpinst*** in particular greatly simplified software maintenance by supporting a new wrinkle, *dependency handling*, another important feature that became canonical to **CRUX** like `prt-get` itself. The format of the `Pkgfile` was extended to allow inclusion of a `# Depends on:` statement that specifies a blank-or-comma-separated list of packages that need to be pre-installed to build and/or run the package having the statement. Although it is much more simple than other distributions' dependency capabilities, and not always correct as a consequence (usually when non-canonical ports are involved), it serves as a crucial starting point for identifying the order in which packages need to be processed.

The `prt-get` package also provisioned two new optional scripts within package directories: `pre-install` and/or `post-install`, which may be invoked by some ***prt-get*** subcommands before they run ***pkgmk*** and/or after they run ***pkgadd*** for the package. There have been other extensions, some canonical. An inadequately documented facility to *alias* depended-upon package names was added to `prt-get` to provide a rudimentary *package provides generic package* and *package depends on generic package* capacity. The `/var/lib/pkg/prt-get.aliases` flat file, which may be customized as needed by an installation, defines such relationships of the first kind. Also, some **CRUX**ers use the `opt/mpup` package to help manage forked ports—see its `README` file for the details (`prtpkg` does not yet directly support that software). More recently `onodera` announced a new ports/prt-get work-alike tool on `#crux`. While I have not yet had the opportunity to deeply look into it, it is my expectation it can be fit into the `prtpkg` framework without too much difficulty.

2.3. Official Website For Collections

`https://crux.nu/portdb` is the canonical resource for all known **CRUX** collections, particularly for downloading the `/etc/ports` *driver config* files for the collections, which are read by ***ports -u*** to know how to access the collections' master repositories. Changes to the set of master repositories and the contents of driver config files usually occur unannounced. The site normally maintains and supports a single release. Access to the release in development is limited until its announcement opens a 2-4 week window when both releases are supported. When the window silently closes, support for the old release ends and later it silently ceases to be available. All the canonical collections are release-versioned. Non-canonical collections can silently be changed between versioned and unversioned.

3. What the Commonwealth Extension Adds to CRUX Software Maintenance

The Enhancement expands canonical port and package maintenance by providing the following missing pieces in a standardized and hopefully easily adaptable and highly flexible manner for any enterprise deploying one or more CRUX platforms.

3.1. In a Nutshell

3.1.1. Shared Maintenance Within or Between Systems

The Commonwealth Extension supports multiple CRUX distribution releases on a single system and/or supports multiple systems, allowing designated systems to provide `prtpkg` *services*:

- `portdb`—synchronize `/etc/ports` with `crux.nu/portdb`,
- `portsu`-–synchronize `/usr/ports` with Internet repositories,
- `prtpkg`—build packages, other maintenance, and
- `deploy`—***pkgadd***/***pkgrm***—defined for now as self-service.

This includes tracking what specific systems and groups of systems are running what releases and are served by what servers. Such a configuration of multiple releases and/or systems sharing a common `prtpkg` maintenance infrastructure is termed a *commonwealth*.

3.1.2. Maintenance Activity Tracking

The Extension supports maintenance logging, especially the ability to *know* what ports are currently installed (not just what packages are installed), when and how they were built and installed, and true chronological logging that facilitates easy commenting upon the maintenance process directly in the logs produced (as *What Mother Never Told You About VM Software Service* clearly stated in March 1983, “Rule Number Four: Always Leave Tracks”). This includes taking snapshots into a compressed tarball of the build environment for ***pkgmk*** including the package directory tree, currently installed ports (implicitly pointing to their build environment snapshots), the kernel’s `.config` file, etc.

3.1.3. Building Configurations

The Extension also supports multiple building configurations, enabling you to both define and know exactly *how* a port will be/was built; e.g., `PATH` and `CFLAG` settings inherited by and possibly not overridden by ***pkgmk*** or the package being processed. This facilitates (1) designating what `builddefs` you want to permit for any set of ports and/or systems within the commonwealth, and (2) selecting the `builddef` variety (`prtpkg` provides `default` and `debug`, or use your own) of any or all packages you wish to deploy on any CRUX system, in a commonwealth or not.

3.1.4. Maintenance Policies

This grouping of capabilities deals with defining the way software maintenance and system/network administration is done and not done from the enterprise level down to the level of individual systems, and implementing protections against actions contrary to such policies (to help you avoid shooting yourself in the foot or worse—at the enterprise level such disasters can be, well, disastrous). Standards are defined and implemented that apply to one or more commonwealths, and all platforms must adhere thereto (refer to Section **4.2** for further explanation of the Extension's approach to such provisioning).

3.2. Rationale

Together, these capabilities can be reasonably hailed as industrial-strength CRUX software maintenance. They set the stage for enterprise-wide automated maintenance roll-outs as well as cross-architectural package builds in the future.

The state-of-the-art for maintaining CRUX is geared to one or two completely independent platforms. The concept of an organization maintaining hundreds (let alone thousands) of CRUX hosts is a dream, or more likely a nightmare, for CRUX developers. At this time it *does* seem the probability of an enterprise needing a team of CRUX sysadmins is microscopic. However, as appealing to an organization as CRUX might be vis-à-vis most GNU/Linux distributions, the problems of scaling canonical software maintenance make the proposition too expensive to pursue. Thus, canonical maintenance is limiting much needed wider acceptance of the distribution, which in turn limits CRUX' mindshare and inhibits expanding the microscopic pool of developer resources.

The challenge for `prtpkg` design is envisioning a one-size-fits-all architecture that can be coded as needed while delivering a subset most likely to be useful to a wide center of the bell curve of potential users. The KISS concept becomes a trifle unclear as new capabilities are added to engineering systems—it seems the system is becoming less simple. First and foremost, new tricks must have compelling justification to be incorporated—there *must* be a crying need for the features within a significant subset of the installed base. Once past that requirement, the design must be straight-forward, clear, near-universal in applicability, easily configured and customized… well, what ESR wrote already. Perhaps this should be termed KIASAPS (Keep It As Simple As Possible, Smartie). But in reality it is part and parcel to KISS, just not so obvious. Simple is not always orthogonal to complex (Exhibit A: the current Linux kernel vis-à-vis version 0.01 released

September 17, 1991). Making simplicity a higher goal than ensuring security, integrity, and perhaps maintainability is unwise.

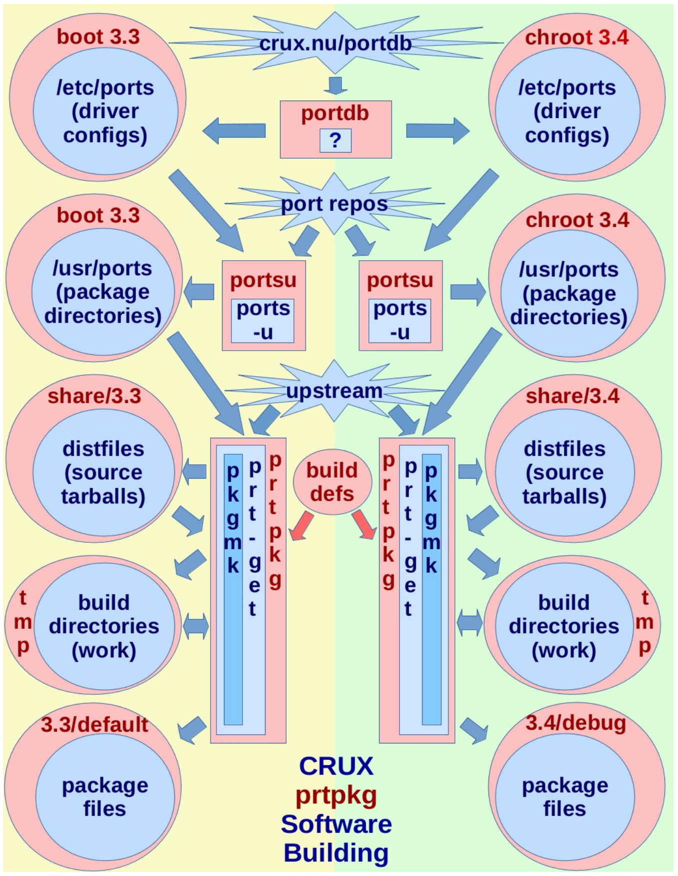
It may be `prtpkg` will never become canonical. That is fine. It may even be `prtpkg` will never be utilized by anyone besides its originator. That, however, is not fine, as previously discussed. But either way, folks should be aware it is available, what it does, and how. If you're a CRUX developer, please avoid breaking it beyond repair if at all possible. If there is sufficient interest, I am willing to pursue establishing a `git` repo before putting up my first port of the `prtpkg` package (I still need to finish the first releasable source tarball!), which would most likely compel me to port this still incomplete document to some other format.

3.3. What The Enhancement Does Not Add to CRUX Software Maintenance

Note `prtpkg` does *not* require any changes to any canonical packages, with one tiny post-***pkgadd*** exception: the software maintenance command executables are renamed to hide them from anything that would directly invoke them. In their places front-ending scripts are installed that kill the process if it hasn't been invoked under `prtpkg` management. It's the simplest way to prevent corruption induced by disregarding `prtpkg` resource locking. If the invocation is approved the front-end uses the shell internal command ***exec*** to invoke the renamed original command. The commands to be protected are ***ports***, ***prt-get***, ***prt-cache*****,** ***pkgmk***, ***pkgadd***, ***pkgrm***, and ***rejmerge***. It is likely ***mpup*** will be added to this list as well. Of course, making it harder for you to shoot yourself in the gut does not make that outcome impossible; e.g., you might do something ill-advised, like editing lock queues without first acquiring the `PRTPKG update` lock (see Section **5.1.3.1** if you don't yet see the need and now won't be able to sleep until you do).

4. Design Concepts

4.1. Graphical Overview



This graphic diagrams how canonical CRUX software building works (blue objects) and what prtpkg adds (red objects). The rectangles are processes (some nested), the ellipses are data areas in the system, and the stars are Internet resources (imagine you're looking down at them hanging over the boxes and ellipses, downloading data into the boxes). The ***portdb*** command at the top updates /etc/ports trees for all affected releases (the canonical approach requires the sysadmin to do this manually if at all). Similarly, the ***portsu*** commands, invoking ***ports -u*** commands, manage the release-specific /usr/ports trees. The ***prtpkg*** commands ride herd on ***prt-get*** commands that ***cd*** into the /usr/ports/*collection*/*port* directories and launch ***pkgmk*** commands that download the upstream distfiles, set up the work directories, expand the source tarballs, then build the packages and "install" them into CRUX package files. What the canonical tools don't know is the ***prtpkg*** commands set things up according to the enterprise's desired build definitions and that the package files are being stored in locations that show exactly how and from what they were built.

If you consider the diagram split vertically down the middle, you see two different build *cells*[4], but there can be more cells in the commonwealth, including boot and chroot cells on different hardware platforms both real and virtual. The associations shown could be 3.4 boot and 3.3 chroot instead, as this is only an illustration. Notice from the canonical perspective, the two cells are close to identical and are totally unaware of each other's existence—the only differences are the versioned collections and ports they process and the likely different *cell* components, especially versioned and builddefed build tools within the *cells*' PATH definitions. Without prtpkg involvement, this diagram requires two bootable CRUX systems or a lot of programming and system administration effort.

4.2. Introducing CRUX Commonwealth Instances (CCI)

A major part of software maintenance and system/network administration for a group of platforms entails defining the ways things must be done on all the platforms in order for the maintenance to be managable by an IT team. While differences between platforms can be tolerated, they must be documented within the `SOP`, and program code as well as human procedures must be expanded to accommodate the differences. It is clearly simpler to maintain many platforms if they are configured identically (or *at least* according to the same set of rules), just as it is clearly maddening to maintain many maverick machines whose vagarities are not clearly documented. Also note `SOP`s are very important to securing `ISO 9001` compliance.

The military preference for the `S` in `SOP` meaning *standing* is closer to the Commonwealth concept, which fully expects and encourages forking the original release to customize it as is useful for any commonwealth deployment. The Extension defines implementations of the Extension that can be fully customized to all requirements identified for one or more Extension deployments. These definitions are called **CRUX** Commonwealth Instances (`CCI`) and are what development of the Extension releases. Initially a single default `CCI` (called `CCIA`) will be released that will encompass all capabilities of the Expansion and serve as the preferred `CCI` to fork. It may turn out maintaining a single `CCI` suitable for all forks becomes impractical, at which point `CCIB` etc. will be released as well as `CCIA`. Any organization can fork any `CCI` as long as the name of the new `CCI` does not begin with `CCI` nor match any other `CCI`'s name.

Each `CCI` encompasses a set of standards and policies to which all participating platforms must conform. A set of `README` files (referred to as the `CCS`), starting with `README.1st` in the `PP-stick`'s `base/_/_` directory, explain and define the `CCI` standards and policies that apply to the release and supplement the `CCI`'s version of the *CRUX 3.4 Commonwealth Manual*. The `CCS README` files for this manual instance (`CCIA`_0.0) are included in Section **7.5** for easy reference. Knowledge of the contents of all `CCI README` files as well as its manual is essential for designing, deploying, and managing a commonwealth based upon that `CCI`. There may be a good reason for multiple `CCI`s to have the same `CCS` definition so that is not precluded.

When forking a `CCI`:

- Document the modifications in the forked manual and fork all files (even if the only change is the name and version of the `CCI`), as well as adding or deleting `README` files as needed (also add or update the `forked from` line of `CHANGES.CCI`),
- Develop/debug the source code needed to comply, and

- Push the forked/extended/new files upline for merging as selectable options into the Extension, thus sharing the alternative CCI so others can benefit while centralizing the maintenance of the alternative CCI within the Extension.

The manual is a LibreOffice 6.0 document intended to be maintained using that software. These tutorials, especially the Writer tutorial, can help make you into a LibreOffice *sensei*—become one before attempting to modify these documents. The official Writer Guide is a valuable and comprehensive reference. All LibreOffice tools use Open Document Format files that can be unarchived, patched, and put back together again, but the XML lines often contain thousands of characters—not at all human or ***patch*** friendly without splitting them somehow in a manner that can be algorithmicly done and undone. This document helps one figure out how to approach ODF revision control. As soon as prtpkg development requires such control, it will be documented elsewhere within this manual.

4.3. Introducing Commonwealth: Global and Local Concepts

To permit a single prtpkg platform to be a prtpkg *porter*[5], *builder*[6], or *deployer*[7] for multiple *releases*, and in consideration of the ***ports -u*** command's requirement that the /etc/ports and /usr/ports trees of the system running it contain the data it is to process, prtpkg is designed to use a ***chroot*** approach for maintaining distribution releases other than that deployed in the prtpkg platform's boot cell. While initially prtpkg will not explicitly support prtpkg platforms that are virtual machines or containers, it may come to light prtpkg really cannot distinguish such from real prtpkg platforms and so supports them inherently. The take-away point of this paragraph is prtpkg is designed to support cells, not systems; i.e., there's not much difference to prtpkg between a system and a cell unless the system is not also a cell (then it just isn't of interest). When a cell deploys prtget-built packages within itself, that should not interfere with any system persona the cell might have, just as that is true for canonical pkgutils activities within a non-prtpkg CRUX system (/etc/pkgadd.conf and ***rejmerge*** are your friends). However, understand such system administration concerns are outside the scope of this documentation.

4.3.1. Boot and chroot Cells For /etc, /usr, and /var

Whether /etc, /usr, and /var reside in the boot cell's root trunk or in a chroot jail elsewhere in a boot cell's root tree, prtpkg considers these construction sites egalitarianly and calls them both cells, jail or not (prtpkg chroot jails contain only one cell).

4.3.1.1. Concepts For chroot Cells

The initial `prtpkg` `chroot` design concepts are:

- `prtpkg` `chroot` cells do not have a system persona—they do `prtpkg` work exclusively (non-`prtpkg` CRUX systems booted and chrooted are of course free to perform canonical software maintenance including simply running ***pkgadd*** using `prtpkg`-built packages without the benefit of `prtpkg` tracking and serialization safeguards);
- `prtpkg` `chroot` cells are treated the same as real machine `boot` cells, virtual machine and container `boot` cells, and any virtual systems' `chroot` cells; and
- `prtpkg` ***chroot*** commands use only local `chroot` cells (this restriction may be relaxed if testing reveals no logical or performance impediments when invoking ***chroot*** into cells accessed via network filesystems).
- If possible, never ***chroot*** into a `chroot` cell from a `boot` cell that does not use the same CRUX version to avoid subtle but nasty bugs due to toolchain incompatibilities. Always perform a CRUX upgrade from an `ISO` or from an `ME` platform that was previously upgraded from an `ISO`.

4.3.2. Senior and Junior Cells For portdb

Cells are also distinguished by their proximity to the ***portdb*** process. Those that are designated as `senior` cells by the enterprise have their `/etc/ports` tree directly updated by ***portdb***, which are designed to closely mirror `crux.nu/portdb` by updating every 24 hours or so, perhaps under control of a `crontab` entry. On the other hand, the `/etc/ports` trees of `junior` cells are updated according to the cells' update policies as defined and implemented by the sysadmin; e.g., when the phase of the moon is just right, examine the state of the associated `senior` cell's `/etc/ports` tree and update the `junior` cell's `/etc/ports` contents with their `senior` counterparts by creating and running a well-commented ***prtpkgbatch*** file of one or more ***cp -p*** and/or ***rm*** commands, thereby leaving excellent tracks. Consider `senior` cells as subscribed to ***portdb*** updates and `junior` cells as not. While the `prtpkg` design expects `senior` cells to fully mirror all `/etc/ports` content of which `crux.nu/portdb` is cognizant, `prtpkg` will support the retention in `senior` and `junior` `/etc/ports` trees of driver config files that are no longer available from `crux.nu/portdb`. There must be at least one `senior` cell in a commonwealth for every distribution *`release`* therein.

4.3.3. Prime, usrport, and symport Cells For portsu

Lastly, cells are also distinguished by their proximity to the ***portsu*** process.

First, some explanation: `prime` and `usrport` cells use the canonical organization of the `/usr/ports` tree, while `symport` cells use a `/usr/ports` tree that contains symbolic links into a canonical `/usr/ports` tree instead of containing the actual package directories (see Section **4.4.1** for more detail about `symports`).

Cells that are designated as `prime` cells by the enterprise have their `/usr/ports` tree directly updated by ***portsu***, which is designed to closely mirror the repositories identified in the active driver config files of the cell's `/etc/ports` tree by updating every 24 hours, perhaps under control of a `crontab` entry. On the other hand, the `/usr/ports` trees of `usrport` and `symport` cells are never so updated, but only according to the cells' update policies as defined and implemented by the sysadmin. Consider `prime` cells as subscribed to ***portsu*** updates and `usrport` and `symport` cells as not. While `usrport` cells can be updated using ***portsu*** or even ***ports -u*** directly, that is contraindicated by any need to locally maintain ports that have been removed from their master repository. There need not be `prtdir` statements for every active collection in the `prime` cell's `/etc/prt-get.conf` file but every commonwealth probably ought to have at least one `prime` cell for each included *release* that does. To prevent ***portsu*** from unknowingly updating the same port multiple times, `symport` layouts must not be used in `prime` cells; i.e., `symport` layouts are only supported for `symport` cells. While it is anticipated `prime` cells will be `senior` cells as well, that is not required. There must be at least one `prime` cell in a commonwealth for every distribution *release* therein. Since `symport` cells link into a particular `prime` or `usrport` cell, an *entanglement* of these cells is recognized (see Section **4.4.2** for details about entanglement).

4.3.4. Filesystems

In a multi-`prtpkg`-platform commonwealth, some of the data managed can and must be shared between cells via a network filesystem and some, like the `/etc/*.conf` files and the `/var` tree, cannot. Well, they could if they could be… well, a lot of things, but it's really easier to just keep them local than attempting to make them sharable. It's a similar problem to supporting multiple releases on one system which is solved using `chroot` cells. In fact `chroot` cells should be shareable across systems with the proper serializations, but in many cases network overhead and propagation delays can render the performance inadequate for tasks like rebuilding `firefox` or `qt5`.

There are several network filesystem packages that are well-supported on GNU/Linux distributions. None are trivial to plan, establish, and maintain. NFS is **CRUX**-canonical (see `opt/nfs-utils`) and is installable via the `ISO` distribution. Samba is the other canonical network filesystem (see `opt/samba`), but it's only for masochists or those who work for a sadist (it should be workable for `prtpkg` if you can control mangling and security adequately). If you're willing and able to port the software, AFS is great if you can swallow its Kerberos dependency, Ceph is up and coming (though for now its CephFS component is still possibly risky), and the Plan 9 paradigm called `9P` developed by Ritchie and Pike might be a good fit for your enterprise's requirements. The point here is if `prtpkg` is your enterprise's first foray into shared filesystems, you really need to look at the big picture for filesystem sharing, not just what `prtpkg` requires, to ensure the effort expended provides the greatest benefit. It's better to get it right the first time (yes, there's "Rule Number Three: Don't Expect It To Work" but that is only in regard to always having a dependable backout plan at the ready when making any system change).

All that `prtpkg` wants from a network filesystem are the fundamentals—enable the cells in the commonwealth to cooperatively share a common file namespace. It does not even require that infrastructure provide intrinsic locking support (see Section **5.1** for details about the `prtpkg` resource serialization facility). Of course, `prtpkg` locks should work just fine on networking filesystems that have their own locking mechanisms, so `prtpkg` can probably just ignore any such filesystem locking.

The location of `senior` and `prime` cells needs to be carefully considered. Because they are directly updated by ***portdb*** and ***portsu*** respectively, they need to be exported into shared namespace if those processes are not running in those cells. Cells' ***prtpkg*** processes do need to be local to the cell, though, for performance considerations. Since it is usually better to avoid exporting `boot` cells' root filesystems, `senior` and `prime` cells should probably be limited to `chroot` cells if their ***portdb*** and ***portsu*** processes will run remotely from them. It's probably better for `senior` and `prime` cells to not have a system persona, as well. Considerations may indicate one or more `boot` cells' root filesystems need to be exported into the network filesystem tree, despite best practices. Security risks may be acceptable. Network performance may not be a bottleneck. You decide.

For now, `prtpkg` is not testing `core/acl` and `selinux` configurations. Let us know how that goes if you venture forth so `prtpkg` can better support everyone.

An unscheduled outage of the shared commonwealth components, especially the lock directories (see Section **5.1** for locking details), is a critical problem if any `update` locks other than the `PRTPKG` lock are in granted state at the time of the outage. These resources should be deployed on the highest-availability system(s) possible to make such difficult recovery tasks as unlikely as possible. Each cell in the commonwealth needs to maintain a `PRTPKG` state indicator in a local filesystem indicating if the shared `prtpkg` infrastructure is functioning or not using a 60-second watchdog function, possibly launched via `crontab`. The craziness that can ensue from losing the shared filesystem is another reason not to ***chroot*** into remote cells.

4.3.5. Userids and Groupids

In addition to the `pkgmk` userid and `nobody` groupid common to canonical **CRUX** software maintenance (that `prtpkg` requires), a `prtpkg` userid and groupid are recommended to facilitate limiting superuser authority available during software maintenance within a commonwealth. The userid should have limited ***sudo*** authorization to minimize risk (but `prtpkg` does not require `sudo`), and the groupid should act the same as `wheel` commonly does for `root` (or just use `wheel`).

Of course, in a commonwealth, all userids and groupids need to have consistent `uid`s and `gid`s across cells. While `opt/openldap` is canonical, it might be simpler to develop scripts to perform the needed maintenance in each cell using the request queue facility described in Section **5.2**.

4.3.6. Environment Variables

4.3.6.1. BOOTOS

A `boot` cell's `/etc/ports`, `/usr/ports`, `/var/log`, and `/var/lib/pkg` trees and `/etc/ports/pkgmk.conf`, `/etc/ports/pkgadd.conf`, and `/etc/ports/prt-get.conf` files are associated with the **CRUX** *release* the cell last booted. Those trees in a `chroot` cell are likewise associated with the **CRUX** *release* installed in the cell. That release is defined in the `BOOTOS` environment variable added to `/etc/rc.conf` file. For a `boot` cell it is automatically exported to every new session by an addition to `/etc/rc.conf`, but for a ***chroot*** session it may need to be reexported as part of the session's initialization.

4.3.6.2. ROOTFS

Also new to `/etc/rc.conf` is the `ROOTFS` environment variable, the filesystem identifier of the `boot` cell's root filesystem (including the enclosing brackets). Refer to Section **7.5.1.3** for details of the naming filesystems in conformance to the `CCS`.

For `chroot` cells not using their own mounted filesystem, it is required the `chroot`'s trunk contain a *ROOTFS* file that shows the *label* is not a filesystem label; e.g., `[SSD1_xyzzy]` where the `SSD1` is the label of the filesystem housing the `chroot` tree. This string is defined as `ROOTFS` in the `chroot's /etc/rc.conf`, and `ROOTFS` must be exported to the environment as part of the ***chroot*** session's initialization since a ***chroot*** session does not boot up.

4.3.6.3. `PRTPKG_CELL`

The `PRTPKG_CELL` variable uniquely identifies the local cell for `prtpkg` purposes. It is simply the concatenation of the cell's `/etc/rc.conf` `HOSTNAME` value (returned by the ***`hostname`*** command) and its `ROOTFS` value (see Section **4.3.6.2**); e.g., `server1[HDD2]`, `server4[SSD3_CRUX-3.4_prime]`. `PRTPKG_CELL` is very nice in a `PS1` prompt.

4.4. Introducing Layers: releases, symports, mixes, builds, and deploys

[Include info from ***`pkgaddconf`*** prologue (Section **8.2.5**) and update there and here]

4.4.1. Symport Collection Sets

Customary local collection trees in `/usr/ports` (used by `prime` and `usrport` cells in a `prtpkg` commonwealth) are arranged as a directory of collection subdirectories with each collection subdirectory containing one port (package directory) for every package in the collection. This is the data organization ***`ports -u`*** works with. The `prt-get prtdir` statements were designed to permit the sysadmin to order the collections such that different port versions of the same package would be selected according to site's preferences with the most-preferred collection earliest in the file. To handle anomalies for particular packages in the ordering, a `prtdir` statement identifying one or more specific ports in a certain collection can be placed before those of the collections more preferred than that certain collection that also contain ports for those packages. Thus, the `prt-get` package selection algorithm finds those ports first and selects them (the `prt-get` Quick start and User Manual documents are highly recommended reading for **CRUX** sysadmins).

A `symport` organization avoids the need for package-specific `prtdir` statements by returning to an organization of collections such that there is only one port for any package within the entire set of collections. This also eliminates the need to get the order of the `prtdir` statements right—the `prtdir` statements effectively define merely the set of collections `prt-get` will work with, even though ***`prt-get`*** processes the `symport`'s set of collections no differently from a `prime` or `usrport` cell's set.

As the name suggests, `symport` layouts use symlinks into a `prime` or `usrport` cell's `/usr/ports` tree to achieve this magic. In establishing a `symport` tree, the upper level directory of collections is identical to that of the `prime` or `usrport` cell; however, the next level usually contains only symlinks to package directories that were or will soon be actually built using the `symport` tree. If the sysadmin wants to change the port of `packageX` from that of collection `xyzzy` to that of collection `plugh`, he/she runs these commands:

```
rm  /usr/symports/xyzzy/packageX
ln  -s  ../../../ports/plugh/packageX  /usr/symports/plugh
```

resulting in /usr/symports/plugh/packageX.

Using symport layouts saves some filesystem space and some ***ports -u*** processing but complicates dependency resolution somewhat as the ***prt-cache --test*** commands using subcommands ***sysup***, ***depinst***, or ***grpinst*** for the symport cell must be run against the entangled prime or usrport cell's /usr/ports tree but using the symport cell's port selection preferences. It would be a good idea to maintain that translation of the prtdir statements as part of the symport cell's maintenance policy (remember ***prt-get*** can be told which prt-get.conf file to use as a command-line argument and prtdir statements can specify absolute paths for the collection directories).

4.4.2. Mixed Cells

Since symport cells link into a particular prime or usrport cell, an entanglement between these cells is recognized. Cells having such entanglements are said to be *mixed* and sets of entangled cells are called *mixes*. A *mix* is defined by a prime or usrport cell followed by the entangled symport cells sorted by collating sequence; e.g.,

```
server3[YY] server1[ZZ] server2[XX_3.3] server4[WW_3.3]
```

where server3[YY] is a prime cell and the rest are all symport cells. The /usr/prtpkg/mixes trunk holds these single-line mix definition files and PORTSU lock requests by a mixed cell are automatically converted to a lock request for the mix. It should be apparent a symport cell can be mixed with only one prime or usrport cell. Like groups, names of mixes may not have the same format as names of cells.

4.4.3. Build Package Trees

Since package trees are shared between the entangled cells of a mix, building by any particular cell must not interfere with building by any other cell in the mix, and updating the shared package trees; e.g., ***portsu*** processing of the prime or usrports tree package tree, must not disturb any build-versioned inodes within. For this reason, prtpkg defines a convention for maintaining such build-versioned files and directories called *build package trees*.

Recall ***pkgmk*** knows nothing related to any particular package it is building outside its working directory; i.e., the package tree. As described in the pkgmk.conf(5) ***man*** page, the PKGMK_PACKAGE_DIR variable defines the directory into which the final CRUX package file will be stored, with a default of $PWD; i.e., the package directory in the prime, usrports, or symports tree which ***pkgmk*** is being processing.

Thus, ***ports -u*** knows not to remove any files in the local package directory that are not in the Internet repository's package tree.

So `prtpkg` is designed to define `PKGMK_PACKAGE_DIR` as `$PWD/pkgs`*`builddef`* where *`builddef`* is the name of a `builddef` in the `/usr/prtpkg/builddefs` tree; e.g., `PKGMK_PACKAGE_DIR=$PWD/pkgsdebug` and ***prtpkg*** ensures the cell's package trees contain the symlinks and the linked directories to receive the package files; e.g.,

```
/usr/ports/core/bc/pkgsdatom →
/usr/prtpkg/CRUX-3.4/packages/core/datom
```

4.5. Introducing Relationships: porters, builders, and deployers

[Develop explanation and relate maintenance roles: which suppliers serve which consumers.]

4.6. Introducing Batches: prtpkgbatch and its *.prtpkg files

[Explain ***prtpkgbatch*** processing and ***prtpkg*** subcommand micromanaging]

4.7. Introducing New Configuration Files: builddef.conf and deploy.conf

[Explain what they do and their formats.]

4.8. Introducing /usr/prtpkg and Where To Find Everything

To accommodate these new variables and processing realignments, it is necessary to transparently extend the canonical organization of the data manipulated by **CRUX** software maintenance. This begins with the addition of a tree to hold the commonwealth-wide extra information. The `/usr` tree is an appropriate location for this trunk because this data must be shareable across multiple systems, but not in `/usr/share` because much of the data will be too volatile for the traditional role of that tree. So it was decided to appropriate `/usr/prtpkg` for this tree.

The root tier holds the following trunks or anchors. Note that `prtpkg` strives to support symlinking as much as possible; indeed, `/usr/prtpkg` itself can be an anchor, if, for example, you want it in a different filesystem from the filesystem containing `/usr` for some reason like not wanting to expose all of `/usr` to other systems. Just remember to ensure dereferences function across multiple platforms as needed (and understand the canonical **CRUX** software maintenance packages are in general quite a bit less tolerant of symlinks even within a single cell).

The branches subordinate to the `/usr/prtpkg` trunk follow the trunk `inodes`.

4.8.1. /usr/prtpkg/release

These inodes anchor the trees for particular versions of any software distribution having ports collections and associated support data compatible with the CRUX maintenance methodology. They are named in the format *distname-version.release*; e.g., `CRUX-3.3`, `CRUX-3.4`.

4.8.2. /usr/prtpkg/broadcasts

This inode is (or anchors) a flat file to which requests and notices to be processed by every cell's commonwealth coordination dæmon are appended. The dæmons receive the requests by piping a ***tail -f*** command for the queue into their ***while read*** loops. If a cell is not active when a request is added then the request is effectively ignored by that cell. See Section **5.2.2.1.1** for more information.

4.8.3. /usr/prtpkg/builddefs

This inode anchors build-definition trees that can be selected for building ports according to a specific combination of options external to the port definition. These can include hardware models/versions, performance objectives, debugging features, linkage attributes—whatever features are desired to be used by one or more packages for one or more systems that can be imposed on package builds outside of the packages proper (yes, systems, not just cells—package files built under `prtpkg` can be installed into non-`prtpkg` CRUX systems). Build definitions do not preclude any package from overriding pieces of the definition. They express the desires of the sysadmin for the package to support that set of features. However, this is in no way a means to describe common package configuration specifications such as the Gentoo/Funtoo distributions' `USE` facility provides; rather, it is intended to target run-time options of the system's build tools; e.g., compilers and linkers. Like the dependency facility, it is not designed to be comprehensive and infallible, but only to facilitate organization and methodologies that enable building, for example, production and debugging versions of a port, hopefully without needing to modify the port itself. Two are provided by the `prtpkg` package: `default` and `debug`, but enterprises can add as many as are useful to them, and perhaps such can be shared as ports have been, to the benefit of all.

4.8.4. /usr/prtpkg/cells

This tree anchors a tree for every cell participating in the commonwealth. Cells are identified via their `PRTPKG_CELL` strings described in Section **4.3.6.3** and each cell provides its own as environment variable `PRTPKG_CELL`. The cell trees anchored in `/usr/prtpkg/cells` are described in Section **4.8.15**.

4.8.5. /usr/prtpkg/groups

Similar to `/usr/prtpkg/cells`, `/usr/prtpkg/groups` anchors trees that define groups of cells and/or groups (a group may include cells and groups) as is useful to the enterprise. Note that a group name cannot conform to the cell name format to preclude being misidentified as a cell. Group definitions that contain no trees are supported. It is expected the inodes in the `/usr/prtpkg/groups` directory are directories containing relative symlinks to trees in `cells`' or `groups`' trunks as is appropriate; e.g.,

```
cellname   → ../../cells/cellname
groupname  → ../groupname
```

Obviously loops of groupnames are unhelpful; e.g.,

```
groups/group1/group2 → ../group2
groups/group2/group3 → ../group3
groups/group3/group1 → ../group1
```

The `prtpkg` package currently provides no group definitions.

4.8.6. /usr/prtpkg/mixes

Somewhat similar to `/usr/prtpkg/groups`, `/usr/prtpkg/mixes` anchors a tree that defines entanglements of one or more `symport` cells with a `prime` or `usrport` cell (see Section **4.4.2** for details). Note that the name of a mix cannot conform to the cell name format defined in Section **4.3.6.3**.

4.8.7. /usr/prtpkg/PRTPKG

The `PRTPKG` tree is a queue of advisory lock bids and grants for `update` (exclusive) and `shared` (unmodifiable) access to commonwealth-scope resources not serialized by other locks. Refer to Section **5.1.3.1** for further details.

4.8.8. /usr/prtpkg/PORTDB

The `PORTDB` tree is a queue of advisory lock bids and grants for `update` (exclusive) and `shared` (unmodifiable) access to all `senior` cells' `/etc/ports` trees in the commonwealth (see Section **5.1.3.2** for details).

4.8.9. /usr/prtpkg/release/builders

4.8.10. /usr/prtpkg/release/deployers

4.8.11. /usr/prtpkg/release/distfiles

4.8.12. /usr/prtpkg/release/packages

4.8.13. /usr/prtpkg/release/work

4.8.14. /usr/prtpkg/release/work

A `PORTSU` tree is a queue of advisory lock bids and grants for `update` (exclusive) and `shared` (unmodifiable) access to all `prime` cells' `/usr/ports` trees subscribed to the release (see Section **5.1.3.3** for details).

4.8.15. /usr/prtpkg/cells/cell

These inodes anchor the trees for the cells associated with this commonwealth. They are named using the *`HOST`*`[`*`ROOTFS`*`]` format described in Section **4.3.6.3**.

4.8.16. /usr/prtpkg/cells/cell/prtpkg.txt

This flat file is the chronological activity log for all `prtpkg` maintenance performed by this cell.

4.8.17. /usr/prtpkg/cells/cell/notices

This inode is (or anchors) a flat file to which notices to be processed by this cell's processes are appended. See Section **5.2.2.1.3** for more information.

4.8.18. /usr/prtpkg/cells/cell/requests

This inode is (or anchors) a flat file to which requests to be processed by this cell's request handler dæmon are appended. The dæmon receives the requests by piping a ***`tail -f`*** command for the queue into its ***`while read`*** loop. If the cell is not active when a request is added and is a `chroot` cell, its associated `boot` cell may be requested to launch the `chroot` cell to begin processing its request queue. If the inactive cell is a `boot` cell, then the sysadmin may need to get involved to reboot the `prtpkg` platform with the *`ROOTFS`* of the needed `boot` cell. See Section **5.2.2.1.2** for more information.

4.8.19. /usr/prtpkg/cells/cell/types

These inodes anchor a flat file containing a single line of three tokens identifying the cell's type attributes in the order shown: `boot` or `chroot`, `senior` or `junior`, and `prime` or `usrport` or `symport`; e.g., `chroot senior usrport`.

4.9. Mapping Old Commands Into New Commands

[relocate from ***`prtpkg h syntax`*** (Section **8.1.2**) and explain port creation process support]

5. Processing Organization

We start detailing processing organization by looking at how to keep multiple releases and/or parallel processes on the same or different `prtpkg` platforms from negatively interfering with each other.

5.1. Resource Serialization (Locks)

To prevent maintenance processes from stepping on each other's toes, there is a need for those processes to communicate their serialization requirements and for `prtpkg`'s logic to coordinate those needs, acting like a traffic light (but it is up to those processes to obey the traffic signals to prevent collisions!). This approach is termed *advisory locking*. Processes requesting a lock will be blocked until the lock becomes available (for some tasks, grants can be blocked for hours). Processes may examine the lock request queues prior to making lock requests to decide, possibly with user input, whether to do something else rather than wait, or to go ahead with what looks to be a short enough wait. The ability to cancel a pending lock request and relinquish ownership of a held lock when a maintenance process is terminating prematurely is also required.

A multi-`prtpkg`-platform commonwealth is best served by a *distributed lock management* solution. As a single-`prtpkg`-platform commonwealth can grow into such a requirement, a single DLM-capable mechanism should provide all `prtpkg` advisory locking even when no networked filesystems are involved. The `prtpkg` resource serialization facility uses its own `NFS2`-style; i.e, stateless, *lock file* mechanism built upon shared directories that are queues containing flat files that are lock requests, both pending and granted (KISS). This locking facility does not even need to itself be explicitly serialized as distributed filesystem and time-of-day syscalls are sufficiently granular and atomic for its purposes. The filenames begin with subsecond-precision timestamps, so `prtpkg` platforms do need to have synchronized clocks (NTP works very well for `prtpkg` purposes).

5.1.1. Serialization Classes

Three serialization classes are used to allow maximum parallelism while ensuring data integrity or cohesiveness to those processes that need it.

The `shared` class provides read-only access to the resource while requiring any modification of the resource be prevented while a `shared` lock is granted. Multiple processes may have `shared` grants for the same resource concurrently.

The `update` class provides exclusive access to the resource for read-write purposes. Only one process may be granted the `update` lock for a resource at any time and no `shared` grants to a particular resource can be in effect when an `update` is granted for that resource. Further, no `shared` rights for a resource will be granted as long as the `update` lock remains granted. All lock requests for a given resource are processed on a FIFO basis, so a pending `update` request blocks subsequent `shared` requests for that lock.

The `floating` class is more accurately a state of an `update` lock in which the lock is currently disassociated from any process ("given up" by the owning process as opposed to "freed up" or "unlocked") but remains held by the cell to which it was granted. Disassociation occurs when the updating process is compelled to terminate without completing the update for which it acquired the lock. Ownership of a floating lock can be subsequently "assumed" by another process in the same cell that will attempt to complete the interrupted update. This might even be an interactive shell should the sysadmin need to resolve the situation without the benefit of suitable automation.

5.1.2. Serialization Operations

When a process needs to change its relationship to a serialized resource, it invokes one of these operations. Processes must ***`trap`*** all signals that by default result in process termination as well as the `EXIT` signal in order to disentangle themselves from any lock queues in which they have active requests or grants. Disentangling requires invocation of ***`lock_cancel`***, ***`lock_unlock`***, or ***`lock_giveup`*** as is appropriate.

5.1.2.1. ***`lock_obtain`***

Locks with which a process has no association are requested via this operation that specifies the resource, class, and any parameters the resource type requires. The operation blocks the requesting process as needed until the lock has been granted.

5.1.2.2. *lock_assume*

When an `update` lock has been `floated` because of a failure to update, only this operation can reacquire the lock so the update can be completed. The requester is expected to be assuming the lock to finish that job and so is distinguished from normal `update` requests that might already be enqueued. As there can be no other grants against a `floating` lock, the request is immediately granted. However, the request must be made from the same cell in which the lock was given up. An error is recognized if there is no `floating` lock to assume.

5.1.2.3. *lock_freeup*

When a process holds an `update` lock for a serialized resource and has completed the update but needs to continue processing with a `shared` grant for the resource, it invokes ***lock_freeup***. This operation, only available when an `update` lock is being processed, is used to convert the lock's class to `shared` and to continue processing with the `shared` authority. Any pending `shared` requests for the lock that can now be granted will be granted before this operation returns.

5.1.2.4. *lock_cancel*

The process' request for a lock that has not yet been granted is removed from the lock queue by this operation, which is normally invoked from a signal handling routine of the process that requested the lock (otherwise it would still be blocked waiting for the lock). If the request has been granted, this operation is treated as a ***lock_unlock*** operation.

5.1.2.5. *lock_unlock*

This operation is used to drop use of the locked resource and continue processing. Any pending requests for the lock that can now be granted will be granted before this operation returns.

5.1.2.6. *lock_giveup*

This operation, only available when an `update` lock is being processed, is used to convert the lock's class to `floating` for premature process termination (or perhaps there is a need to reboot the cell to complete the update). The operation does not terminate the process, it returns so the process may continue managing its termination.

5.1.3. Serialization Types

The specific lock types are listed in the order they must be acquired to preclude process deadlocking. Likewise, lock types must be freed up, given up, or unlocked in the reverse order they were granted.

5.1.3.1. Global Serialization: PRTPKG

There exist `prtpkg` updates that can touch just about everything `prtpkg`, so even though such modifications should be infrequent, non-updating users of those resources need assurance the world will not shift beneath them. This is the *raison d'être* of the `PRTPKG` lock. Experience may reveal the need to add more granularity at this high impact locking level. On the other hand, if only one lock can be implemented, this is the one. The resources encompassed by the `PRTPKG` lock are all resources in the commonwealth not covered by the other lock types—it is the catch-all lock of `prtpkg` serialization.

Since all other `prtpkg` processing depends on the stability of these foundational resources, all processes must hold at least a `shared PRTPKG` lock while performing `prtpkg` processing. If such processes are complex or long-running, they should monitor the `PRTPKG` lock queue and unlock/reacquire their `shared PRTPKG` lock at the first opportunity to yield to any waiting `update` bidder as needed. The more cells that are sharing the `PRTPKG` lock, the more important such cooperation becomes.

When a `PRTPKG update` lock request is made, every other `prtpkg` process in the commonwealth receives a `USR1` signal it interprets as a quiesce directive. Each process must in response gracefully abort what it is doing if it cannot quickly complete what it is doing. Then the process must unlock its `shared PRTPKG` grant and immediately either terminate or issue a new `shared PRTPKG` lock request. Examples of `PRTPKG` updates include a scheduled outage of the network file system for maintenance; manipulating the makeup of the commonwealth such as the sets of defined cells (all *`HOST`*`[`*`ROOTFS`*`]` entities), groups, or releases (the set of *`releases`* in the `/usr/prtpkg` trunk). Also, changing the release of any cell must be performed while holding the `PRTPKG update` lock (perhaps *giving up* the lock to convert it into a `floating` lock to persist across any actual reboots involved, and *assuming* it again once rebooted—see Section **5.1.2**).

5.1.3.2. Driver Config Serialization: PORTDB

The ***`portdb`*** processes can update `/etc/ports` contents of multiple `senior` cells and only knows what will need to be done for each collection while processing each collection. Consequently the `PORTDB` lock is designed to serialize the group of resources ***`portdb`*** maintenance can affect; i.e., the contents of all `senior` cells' `/etc/ports` trees. Such update serialization also ensures only one ***`portdb`*** command can be processing at a time within a commonwealth (but why should there be a need to run different ***`portdb`*** commands at the same time, anyway?). Note this lock does not serialize the `senior` cells, only their `/etc/ports` trees, but `shared PORTDB` locks acquired by ***`portsu`*** processes sufficiently protect them. Also,

`prtpkg` processes do not access `/etc/ports` trees, so they can ignore `PORTDB` locking; however, ***`prtpkgbatch`*** scripts; i.e., *.`prtpkg` files, can, and care must be taken in creating such to ensure `prtpkg` locking is properly utilized.

5.1.3.3. Collection Serialization: PORTSU{*|collection}

`PORTSU` locks protect collection trees from or during (1) ***`ports -u`*** processing of `prime` and `usrport` `/usr/ports` trees, (2) non-***`ports -u`*** reconfiguration of `symport` `/usr/ports` trees, and (3) package building. The lock is requested by a specific cell, but if that cell is entangled with any other cells, the lock request is converted from a cell request into a mix request (see Section **4.4.2** for details).

Note: in the rest of this section, understand the use of *mix* to mean *mix or cell* inclusively.

Two scopes of collection serialization at the same level protect collection trees and they are mutually exclusive for any given process:

5.1.3.3.1. All collections: `PORTSU_*`

This scope protects every collection in the mix of the lock request to be serialized. It is treated as and behaves as a set of `PORTSU` locks for every collection in the mix—all processed at the same time during a lock operation.

5.1.3.3.2. One collection: `PORTSU_`*`collection`*

This scope locks a single collection in the mix of the lock request.

What is serialized specifically is the content of the requesting cell's `/usr/ports` directory or, if entangled, the entangled `prime` or `usrport` cell's `/usr/ports` directory) for sharing or updates. The granularity of `PORTSU` locks is for the mix of the requesting cells. The lock queues are placed in the shared cell trunks of the commonwealth; i.e., every cell has a `PORTSU` lock queue as a `/usr/prtpkg/cells/`*`cell`*`/PORTSU` trunk or anchor. Entangled cells will anchor a lock queue at `/usr/prtpkg/mixes/`*`mix`*`/PORTSU` that is shared by all the entangled cells (instead of a private cell lock queue).

Requests for `PORTSU update` locks require a `shared PORTDB` lock be already granted to the requesting process. Similarly, processes requesting a `PORTDIR update` lock are required to already be granted the appropriate `shared PORTSU` lock.

5.1.3.4. Port Directory Serialization: PKGDIR_collection_package

The integrity and cohesion of package directories is maintained through a `PORTDIR` lock and only one may be requested or held by a single process and its children at a time. Changing the collection of a `symport`'s package requires a `PORTDIR update` lock for the port, as does changing the content of a usrport or symport package directory.

The granularity of PORTDIR locks is for whatever mix or unmixed cell the requesting process runs within. The PORTDIR lock queues are placed in the shared mix or unmixed cell trunks of the commonwealth; i.e., in every cell, a /usr/prtpkg/cells/*cell*/PORTDIR trunk or anchor provides its PORTDIR lock queue.

5.1.3.5. Build Serialization: WORK_portname_version_buildname

These locks protect the package building work directory for active build and post-build work archive processes.

5.1.3.6. Deploy Serialization: PKGFILE_package_version_buildname

These locks protect the package file for active prtpkg and deploy processes. Note that all pkgutils commands maintain their own serialization via ***flock***(2) processing on the /var/lib/pkg/db file and therefore within the cell being modified by their processing.

5.2. Inter-process Communication

Processes communicate with each other both synchronously and asynchronously in a commonwealth via shoulder-tapping provided by:

5.2.1. Signal Processing

Prtpkg processes can raise a signal(7) for another process in the same cell to handle. This is accomplished via a ***kill*** command, system call, or some library call. Prtpkg will not depend upon cross-cell signal support within prtpkg platforms (see Section **5.2.2.1.2** for how prtpkg meets this need).

5.2.2. Shared Files

The creation, modification, or removal of an inode in common namespace by one process for notice and/or processing by another process is a methodology for inter-process communication as old as UNIX (actually, older—IBM S/360 mainframe).

5.2.2.1. Broadcast, Request, and Notices Queues

In the interest of KISS, these queue polling facilities are used to eliminate a prtpkg requirement for ssh with its passwords and/or PKI infrastructure (the networked inode permissions of the broadcast, request, and notices queue paths are sufficiently secure for prtpkg's purposes).

5.2.2.1.1. Commonwealth Broadcast Queue

This shared file needs to be polled by every cell's commonwealth coordination dæmon to act upon inter-cell notices and requests to groups of cells, possibly to all cells in the commonwealth. Each commonwealth coordination dæmon appends requests germane to its cell to the cell's `request` queue for handling by the cell's request handling dæmon and appends germane notices to its `notices` file.

5.2.2.1.2. Cell Request Queues

Polling of a cell's `request` queue by the cell's request handling dæmon enables cells to respond to processing tasks assigned by other cells within the commonwealth. This mechanism provides the means to indirectly signal processes in other `prtpkg` platforms as well as other cells within the same `prtpkg` platform.

5.2.2.1.3. Cell Notices Queues

This shared file acts as a broadcast file at the cell level, receiving appended lines representing notices of significance only within that cell. It is expected that all processes in the cell poll this file as needed. It is not intended to serve as a logging facility.

6. Installs and Upgrades For Cells

While this manual replaces the canonical *CRUX 3.4 Handbook* as the documentation for installing and upgrading CRUX within a commonwealth, it is expected the reader already understands all the information the handbook contains. This section may include links into the handbook as are helpful.

The canonical approach to installation or upgrade of a CRUX distribution release uses (for upgrade) a scheduled outage of the target system (for install that's unnecessary) during which the release's ISO is booted and the system is installed or upgraded as needed. Within a `prtpkg` commonwealth, this canonical processing is expanded, particularly via a merge and enhancement of the canonical ***setup*** scripts into the ***prtpkg_setup dash*** script, plus the new ***mk_cell*** and ***up_cell*** scripts (for `chroot` cells only that need not the ISO environment).

The chicken-and-egg conundrum of `prtpkg` not being part of the ISO nor formally packaged for CRUX is side-stepped by employing an additional installation medium called the `PP-stick` that is an `ext4`-formatted partition labeled `PP` residing in a GPT-formatted USB-stick (refer to Section **7.6.9** for more detail).

Other differences between canonical and `prtpkg` installation/updating should be noted. Parameters such as the target system's *`hostname`* are solicited prior to invoking ***`setup`*** and many are determined without human interaction, in particular, the target system mountpoint. The Extension includes a script for partitioning the primary SSD/HDD, ***`prtpkg_pdrive`***, and a script for formatting `ext4` partitions, ***`prtpkg_mkfs`***. There are many other similar components that automate much of the install and upgrade processes.

As cells in a commonwealth require synchronized clocks, it becomes necessary to install an `NTP` dæmon, which in turn requires a package; e.g., `chrony`, `ntpd`, `ntpclient`, not distributed in the CRUX `ISO`. This complication is compounded by the `ISO` design of running in a system using a `RAM` disk for its root filesystem which of course takes away from memory space for processing programs. This can be a show-stopper for machines with less than 1 GB of `RAM` (the handbook states a minimum of 192 MB is needed to install CRUX). Therefore, `prtpkg` provides multiple ways to provision the maintenance system for a particular machine and automata for implementing something that gets the job done. The sysadmin must be more involved in the planning of the maintenance methodology for each machine in a commonwealth, as a consequence, but that is hopefully only necessary when the machine is initially added to the enterprise's infrastructure. Indeed, the first is always the most laborious, but as experience grows and new platforms can be more and more cloned from existing machines, the labor diminishes and machines can be added more and more quickly.

6.1. Canonical CRUX Kernel Maintenance

The `ISO` boots a recent stable Linux kernel built using the `ISO`'s `.config` file which the `ISO` contains along with the vanilla kernel source tarball that kernel was built from plus any applicable vanilla patches. The `ISO`'s ***`setup`*** script looks in the target's `/usr/src` tree for the `ISO` kernel's version. If it's not there, (1) the kernel tarball is expanded into the target's `/usr/src` tree, (2) any kernel patches the `ISO` contains are applied, (3) the `ISO`'s default `.config` file is copied into the new tree, and (4) if the target's `/lib/modules` tree for the release does not exist, its branch is created and the target system's module dependencies are pre-populated via a ***`depmod -b $ROOT -a $KERNEL_VERSION`*** command. Other than this, CRUX completely ignores kernel maintenance. The release-dependent handbook expects the sysadmin installing or updating the system will ***`chroot`*** into the target root filesystem from the `ISO` system and configure, build, and install a kernel therefrom (not necessarily the one the ***`setup`*** script may have expanded). CRUX does not provide, let alone maintain, a kernel package that can be processed by `pkgutils`. As most CRUXers cannot wait for a new CRUX release every 12-18 months to update their kernels, they do what the CRUX developers expect: they retrieve kernel releases and/or patches from kernel.org, expand/apply them, customize the .config files as needed by the enterprise and the platforms, and build and install the new kernels, all according to the kernel's README document, just as they did for the install or upgrade. CRUX development neither adds to kernel distributions nor engineers CRUX-specific kernel patches.

6.2. CRUX Commonwealth Kernel Maintenance

Everything in the preceding section is also true for prtpkg software maintenance systems; however, an enterprise is completely at liberty to add its own kernel software maintenance policy and automata to its own commonwealth(s). Contribution of same to the community is certainly encouraged—there will hopefully be many organizations with similar goals in this regard.

6.3. Obtain the CRUX ISO and Provision the Bootable Medium

This is exactly as described in the beginning of Section 3.2 of the handbook (of course, you should only need to download the ISO image, verify its MD5 checksum, and install it on an appropriate medium once per CRUX ISO release. It would be very good to document how your enterprise obtains and maintains the CRUX ISOs.

6.4. Obtain the prtpkg Components and Provision the PP-stick

Refer to Section **9.2** for the details of obtaining, customizing, and provisioning the PP-stick to use. It would be very good to document how your enterprise obtains and maintains its PP-stick(s), too.

6.5. Installing Into Boot Cells

6.5.1. Boot the ISO

Installs into boot cells begin in the canonical way by loading the ISO media via an appropriate peripheral device or hardware port of the platform, then booting the the not-yet-a-cell system from that media. This process is no different from what is described in Section 3.2 of the handbook. Immediately after the root shell becomes available for command input via the keyboard, the first difference between the canonical and commonwealth installation procedures is encountered.

6.5.2. Mount the PP-Stick

Insert the PP-stick into a USB port and mount it; e.g.,

```
mkdir -p /_/l/PP && mount -L PP /_/l/PP -o rw,noatime
```

6.5.3. Invoke the prtpkg_source Script

The `prtpkg_source` file is invoked via the following command (`.` or ***source***):

```
. /_/l/PP/bin/prtpkg_source
```

Refer to Section **7.6.16** for full details of this script including its source code.

6.5.3.1. Pre-existing Commonwealth

For installs involving a pre-existing commonwealth, some sysadmin legerdemain installs any needed networking filesystem software into the `RAM`-resident root tree, configures it, and mounts all needed network filesystems into the target root tree, usually mounted as `/mnt` in the `RAM`-resident root tree, before running the ***prtpkg_setup*** script instead of the canonical ***setup*** script to convert the freshly formatted target root tree into a new `boot` cell root tree. This includes obtaining the `PRTPKG update` lock to safely add the cell to the commonwealth (refer to Section **5.1.3.1** for details). Experience with installing new platforms into existing commonwealths may lead to revisions in this approach that will ease the complexity of the legerdemain phase, possibly with goodly amounts of `prtpkg` package and/or homegrown automation.

6.5.3.2. New Single-platform Commonwealth

If a single-`prtpkg`-platform commonwealth is being co-installed with the `boot` cell, a `/usr/prtpkg` tree is simply established in the target cell's root tree.

6.5.3.3. New Multi-platform Commonwealth

However, if a multi-`prtpkg`-platform commonwealth is being co-installed (only `NFS` is installable from the `ISO` and must be installed for this situation), ***prtpkg_setup*** configures the `boot` cell's network filesystem server to export its `/usr/prtpkg` tree per a dialog with the sysadmin (that may prove to involve homegrown scripts).

6.6. Installing Into chroot Cells

For new `chroot` cells, installation is accomplished by running the ***mk_cell*** script in the associated `boot` cell as a normal command without needing to reboot anything (TODO really soon). Such installs are only supported from `boot` cells already within a commonwealth.

6.7. Release Upgrades—Overview

Upgrades are only performed for cells already within a commonwealth. There are two phases of the process regardless of the cell types: (1) the upgrade of the core toolchain packages and the `BOOTOS` in the `/etc/rc.conf` file, and (2) the upgrade of versioned collections to the new version; e.g., in file `/etc/ports/core.rsync`, the version `3.3` in the line `collection=ports/crux-3.3/core/` becomes `3.4`.

6.7.1. Phase 1 for boot Cells

For `boot` cells, the first phase is effected by booting the versioned **CRUX** `ISO` in the cell being upgraded and using the ***`prtpkg_setup`*** script in place of the `ISO`'s ***`setup`*** script—not radically different from a canonical upgrade.

6.7.2. Phase 1 for chroot Cells

For `chroot` cells, the first phase is effected by running the ***`up_cell`*** script in the `chroot` cell as a normal command without needing to reboot anything (TODO really soon).

6.7.3. Phase 2 for All Cells (portdb update)

For all cells (but see Section **4.3.2** for some important nuances), the second phase is handled by the ***`portdb`*** script and its `update` lock. Usually this change occurs within a few weeks of the announcement of the new release. If a `senior` cell (see the same section) for the *`release`* does not exist in the commonwealth when that occurs, the versioned build config files are inactively held in an available `senior` cell until an appropriate `senior` cell for them has been established.

7. Commonwealth Extension Components

Note that the authoritative licensing and warranty information for canonical **CRUX** components is found in the Licensing section of the CRUX 3.4 Handbook.

7.1. Version and Status

The version of the `CCIA` **CRUX** Commonwealth Instance (`CCI`) is 0.0, which includes the version of the handbook you are reading, all its source code, and its `CCS`. The status is early alpha. A `CCI` version has two parts:

- the first is the ordinal number of the current baseline of the release (if zero, the release has yet to be baselined, the process in which change history is archived and cleared).

- The second is the ordinal number of the the most recent change to the release affecting the component-in-context. If the context is the entire release, it is the ordinal of the most recent change to the release. Zero indicates no changes so far since the baseline was performed.

Thus, files of the CCI can have different version numbers within the same baseline of the CCI, and no file can have a version higher that that of the entire CCI. The change log for a CCI baseline is maintained in its CHANGES.CCI file in its barbar directory, and modified files maintain a list of CCI change ordinals affecting them in their prolog sections.

The distinctions between the Extension, a CCI, and a CCS can be difficult to conceptualize. Technically speaking, there is no global CEX release—only CCIs are released. It is easiest to view CEX changes as global in scope and affecting multiple if not all CCI versions. In terms of applying global changes, they are simply applied (or not) to each CCI as CCI changes and are assigned CCI change ordinals; thus, a CEX change can have different CCI change ordinals associated with it. If a particular CCI has no impact from a CEX change, it can just be ignored excepting a comment to that effect in the CCI change log (refer to Section **7.2.2**) without assigning it a CCI change ordinal.

7.2. Summary of Changes to the Current Baseline Release

7.2.1. CHANGES.CEX—Log of Global Modifications

All CCI releases will include an importation of this file. Note there has been no previous release—thus far David L. Craig is developing the Extension independently and has not yet begun tracking changes.

```
# CCIA_0.0    /_/_/CHANGES.CEX ...:....4....:....5....:....6....:....7....:....8....:....9....:....0
# ****************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ****************************************************************************************************
# List of Global CRUX Commonwealth Extension Changes Affecting Multiple CCIs

Ordinal YYYYMMDD                   Summarization and List of Affected Components
------- -------- ------------------------------------------------------------------------------------
------- -------- ------------------------------------------------------------------------------------
```

7.2.2. CHANGES.CCI—Log of Instance Modifications

When a CCI is forked, the CHANGES.CCI file of the new CCI must add or update a forked from line having the format:

```
# This CCI was originally forked from CCIName_CCIVersion on YYYYMMDD.
```

immediately following the `# List of changes` line. The version of a new `CCI` must always be 0.0, and the log must be empty. If a `CCI` was not forked, this line should be absent.

```
# CCIA_0.0    /_/_/CHANGES.CCI ...:....4....:....5....:....6....:....7....:....8....:....9....:....0
# ****************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ****************************************************************************************************
# List of Changes to this CCI (CEX changes with CEX ordinals are cited and given CCI ordinals).

Ordinal YYYYMMDD                   Summarization and List of Affected Components
------- -------- ------------------------------------------------------------------------------------
```

7.3. Licensing

To over-simplify the licensing policy of both canonical CRUX and the Commonwealth Extension, "*Freely you have received, freely give*" is a useful quotation. See the `DLC_Copyright_License_and_Warranty_Information` file for licensing details and possible clarifications of the following subsections.

7.3.1. Source Code Files

The source code components of this release of the Commonwealth Extension consist of ***bash***, ***dash***, and ***gawk*** script files. All files that are entirely source code for executable computer programs within the release (including comments that are processable by each file's target interpreter or compiler) are Copyright © 2015-2018 by David L. Craig <dlc.usa@gmail.com> and are licensed through the standard GNU General Public License Version 2 which is also contained in the `COPYING` file.

7.3.2. All Other Files

All Commonwealth Enhancement files that are not entirely source code for executable computer programs are Copyright © 2015-2018 by David L. Craig <dlc.usa@gmail.com> and are licensed through the standard *Creative Commons Attribution-ShareAlike 4.0 International License* (aka CC-BY-AT-4.0).

All executable source code examples within this set of files including example shell commands suitable to be copy/pasted into a ***bash*** or ***dash*** shell by the reader are still licensed under the CC-BY-AT-4.0. However, any entire source code files as discussed in Section **7.3.1** that are embedded within a file covered by this section shall not be construed to be governed by the CC-BY-AT-4.0 (the comments within such embedded files will include the copyright and licensing particulars of the embedded file). This set of files includes the document you are reading, written and maintained using the WYSIWYG *LibreOffice 6.0 Writer* tool, and the graphic in Section **4.1**, created and maintained using the WYSIWYG *LibreOffice 6.0 Draw* tool.

7.4. Warranty

Like canonical CRUX itself, the Commonwealth Extension is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY; without even the implied warranty of MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Use this software at YOUR OWN RISK.

7.5. CRUX Commonwealth Standards (CCS) README Files

7.5.1. CCS README Files in the barbar Directory

7.5.1.1. First, Some Orientation—README.1st

```
# CCIA_0.0    /_/_/README.1st ....:....4....:....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#---------------------------------------------------------------------------------------------------
These README text files (within a particular root filesystem when it is being used as such) document
the CRUX Commonwealth Standards (CCS) used by any commonwealth using this specific CCI as identified
in the first line of this file.  The CCS defines standing design and maintenance aspects of the
computers participating in a commonwealth using this CCI.  Not including README.thisdir* files, the
CCS primary README files are (in collating sequence):

/_/_/README.1st             CCIA introduction [this is the file you are reading]
/_/_/README.barbar          describes the "barbar" directory tree branch
/_/_/README.basebar         describes the "basebar" directory tree branch
/_/_/README.bootbar         describes the "bootbar" directory tree branch
/_/_/README.cci-variations  describes how to fork a new CCI from CCIA
/_/_/README.change-ctl      describes how CCIA impacts CRUX change control
/_/_/README.dir-types       describes the dir-type/entity organization scheme
/_/_/README.filesystems     describes filesystem labeling and relationships
/_/_/README.rootbar         describes the "rootbar" directory tree branch

This entire set of CCS-documenting files is deployed by the Commonwealth installation process by
copying the rootbar template tree branch in the PP-stick being used; e.g.,
  /bin/cp  -a  /_/l/PP/base/_  /_/l/XX
assuming the target root filesystem is labeled XX and is properly mounted.  As that template tree
branch is the base/_ directory of the PP-stick, it is referred to as the basebar directory and only
exists within PP-sticks.  When copied into a root filesystem as its _ directory, the new tree branch
is referred to as the rootbar directory of that root filesystem; e.g., /_ when the root filesystem
has either been booted or chrooted into.

README.thisdir files are intended to answer the question, "For what purpose(s) does this directory
exist?"  When tree branches are copied elsewhere, those purposes often change, and it becomes
necessary to change one or more README.thisdir files within the new tree branch.  It is also often
helpful to document the origin of the branch, so the file can be renamed README.thisdir.origin while
some copied new version named with an appropriate new-purpose suffix; e.g., README.thisdir.rootbar,
gets renamed to README.thisdir.
#---------------------------------------------------------------------------------------------------
```

The `README.thisdir*` files referenced in the previous paragraph that are components of the CCS are embedded within Section **7.5.2**.

7.5.1.2. What Are dir-types—README.dir-types

```
# CCIA_0.0    /_/_/README.dir-types ...4....:....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ****************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
Many tree branches in the CCI contain sub-branches of various component types having recurring
natures.  A dir-type convention is defined to standardize these breakouts where they are needed
and/or useful.  The primary structure is to deploy a set of peer subdirectories defining the types
of each subdirectory with each named using a single letter.  The subdirectories' entries' names are
instances of the appropriate type; e.g., 'h' is the dir-type for a tree branch of hosts and the
hostnames included in the tree are the names of the subdirectories of the 'h' directory; e.g.,
h/host1, h/host2.

The dir-type standard allows the dir-type/instance organization to map into an inode name format
where the separating / becomes a dot or an underbar.  The enables flattening the two directory
levels into a single directory when that organization is preferable; e.g., h_host1, h_host2.  Use
only one of the separators within any such directory unless there is a good reason for both (which
must be documented, possibly only in the README.thisdir file of the directory, but still requiring a
fork of the CCI in and of itself unless the CCI has already been forked to discuss this modification
of the CCI).

A '_' directory is not technically a dir-type since it is not normally qualified by multiple
instances; however, bar directories are documented here as _ is an important part of the CCI
organization:

  /_ (aka rootbar)
       Resides of course within the root trunk directory of the booted system and serves as the
       anchor for the CCS thereof.  See /_/_/README.rootbar for more information.
  /boot/_ (aka bootbar)
       Contains only files common to all root filesystems of the platform that need to be available
       when networking is not; e.g., boot management, single-user mode.  See /_/_/README.bootbar for
       more information.
  /_/_ (aka rootbarbar or barbar)
       The anchor of the universal rsynced tree that every platform maintains.  See
       /_/_/README.barbar for more information.
  base/_ (aka basebar--intended to exist only in a PP-stick)
       Provides the template tree branch for all rootbar trees installed using its particular
       PP-stick; e.g.,
         cp -a /_/l/PP/base/_ /_/l/NewRootFS
       See /_/_/README.basebar for more information.
  base/_/_ (aka basebarbar--intended to exist only in a PP-stick)
       Provides the template for all barbar trees installed using that particular PP-stick, normally
       installed automatically as a subtree of the target rootbar using the previously shown command
       example.  See /_/_/README.basebarbar for more information.
  All other "_" directories are explained in their README.thisdir files.

The following dir-types contain mountpoints named using the labels of the filesystems mounted
therein.  Note these dir-types may be used in any dir-type hierarchy as is useful--they may only
used as mountpoints in the rootbar trunk; e.g., /_/l/PP is the mountpoint for the PP-stick.

  l/label - local filesystems used privately by the real host and possibly any hosts it virtualizes,
            or for local filesystems fully or partially exported to other hosts via a network file
            system
  n/label - read-only network-mounted filesystems
  w/label - read-write network-mounted filesystems
  r/label - base directories for various rsynced filesystems or tree branches

In addition to mountpoint-related dir-types, the following dir-types are defined:

  h/host   - versioned for a real (non-virtual) host platform
  v/host   - versioned for a virtual host platform
  o/os     - versioned for an OS; e.g., CRUX, FreeDOS; each should have release subdirectories as
             needed for release-versioned files and subdirectories; e.g., o/CRUX/3.4
  p/arch   - versioned for a processor type; e.g., Atom, Core2, ARM
  u/login  - items versioned for the user (usually symlinked from the user's home directory and not
             owned by the users, and the tree branches should not be traversable by the world)
  g/group  - non-home directory files/directories that are not part of a supported package, not
             associated with any particular user, and are associated with the particular group (the
            tree branches should not be traversable by the world)
#-----------------------------------------------------------------------------------------------
```

7.5.1.3. Filesystem Standards—README.filesystems

```
# CCIA_0.0    /_/_/README.filesystems .4....:....5....:....6....:....7....:....8....:....9....:....0
# ***********************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***********************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#-----------------------------------------------------------------------------------------------
All CCIA_0.0 filesystems should have partition names agreeing with their filesystem names (denoted
as label herein), and the filesystem's root directory must contain, if possible, a "[label]" file to
enable straight-forward, consistent identification within the filesystem itself; e.g.,

  ls -aC /
  .       _    dev  lib   lost+found proc sbin usr
  ..      bin  etc  lib32 mnt        root sys  var
  '[XD]' boot home lib64 opt        run  tmp

for the filesystem named 'XD'.

Usually label files contain the host name (or possibly the virtual host name) of the platform on
which they are hosted, but may be empty.

In CCIA, each UNIX-platform must have a separate non-FAT/non-NTFS boot partition.  UEFI platforms
must have a FAT16 or FAT32 ESP partition in addition to the boot filesystem and that must be
bind-mounted as /boot/efi.

All filesystems that can be specified as a root filesystem when booting a CCIA system's kernel must
contain a rootbar directory (see /_/_/README.rootbar for details).

All filesystems that can be specified by BIOS or UEFI setup as a Linux boot partition must contain a
bootbar directory (see /_/_/README.bootbar for details).  CCIA requires Linux kernels to be in
non-FAT partitions, and UEFI ESP partitions must be mounted as /boot/efi when processed within a
CCIA system.

Alas, some trees, like /lib/modules' release build branches and /lib/firmware, must reside in the
root filesystem, but tarballs thereof are maintained in the bootbar tree for rescue facilitation.
In this way, then, all kernel objects are deployed into the boot partition for sharing by all of a
platform's root filesystems.  Kernel objects should be distributable across the enterprise, of
course, but only those in actual use by a platform need reside in the platform's boot filesystem.
Note no kernel components are tracked in the CRUX database (/var/lib/pkg/db).

Only local filesystems' directories should be used as mountpoints including bind-mountpoints.  All
primary mountpoints should reside in the rootbar directory.  The platform's root directory should be
the only exception, and it must be bind-mounted to its /_/l/label mountpoint.  Normal local mounts
shall be to the primary mountpoints in /_ and any logical mounts shall be bind-mounted to the
primary mountpoints; e.g.,

  In /etc/fstab:

    LABEL=XD   /        ext4   defaults,noatime     1 1
    /          /_/l/XD  none   bind
    LABEL=XB   /_/l/XB  ext4   defaults,noatime     0 0
    /_/l/XB    /boot    none   bind

  which df -alHT displays as:

    Filesystem     Type      Size  Used Avail Use% Mounted on
    /dev/root      ext4      155G   18G  138G  12% /
    /dev/root      ext4      155G   18G  138G  12% /_/l/XD
    /dev/sda6      ext4      520M   56M  449M  11% /_/l/XB
    /dev/sda6      ext4      520M   56M  449M  11% /boot
#-----------------------------------------------------------------------------------------------
```

7.5.1.4. What Is the /_ Directory—README.rootbar

```
# CCIA_0.0    /_/_/README.rootbar :....4....:....5....:....6....:....7....:....8....:....9....:....0
# **************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# **************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#---------------------------------------------------------------------------------------------------
Existing in every root filesystem's trunk directory conforming to this CCS, the /_ (rootbar)
directory is the repository for every operating system localization of that boot cell (with the
exception of the /boot/_ aka bootbar tree) that must be available when only local filesystems are
available.  It includes all dir-type directories as described in the /_/_/README.dir-type file;
i.e., it includes the CCS mountpoint directories for the cell.
#---------------------------------------------------------------------------------------------------
```

7.5.1.5. What Is the /boot/_ Directory—README.bootbar

```
# CCIA_0.0    /_/_/README.bootbar :....4....:....5....:....6....:....7....:....8....:....9....:....0
# **************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# **************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#---------------------------------------------------------------------------------------------------
Existing in every boot filesystem's root directory conforming to this CCS, the /boot/_ (bootbar)
directory is the repository for all /boot tree branches' localizations of all local root filesystems
of the platform that must be available before the kernel and the selected root filesystem are
available during boot processing.
#---------------------------------------------------------------------------------------------------
```

7.5.1.6. What Is the /_/_ Directory—README.barbar

```
# CCIA_0.0    /_/_/README.barbar .:....4....:....5....:....6....:....7....:....8....:....9....:....0
# **************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# **************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#---------------------------------------------------------------------------------------------------
Existing in every root filesystem's root directory conforming to this CCS, the '/_/_' (barbar or
rootbarbar) directory is the universal rsynced tree that every platform maintains.  It contains only
files that most (if not all) platforms need to have in their root filesystems that are never
versioned other than as a global update to be propagated to all platforms immediately.  No r trees
are required to be universal and r trees should never be symlinked from barbar.  The rationale of
the barbar tree is for all systems to have up-to-date copies of files therein accessible when remote
systems are not; e.g., an NFS host is down.

The barbar directory has h, v, o, p, u, and g dir-type subdirectories (see /_/_/README.dir-types)
for inodes used by most if not all platforms using the CCS, vis-a-vis those in the rootbar directory
that are used only by the system using that filesystem as its root filesystem.
#---------------------------------------------------------------------------------------------------
```

7.5.1.7. What Is the PP-stick base/_/_ Directory—README.basebarbar

```
# CCIA_0.0    /_/_/README.basebarbar ..4....:....5....:....6....:....7....:....8....:....9....:....0
# **************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# **************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
```

```
#---------------------------------------------------------------------------------------------------
A  base/_/_ directory, aka basebarbar, resides only in PP-sticks.  It is the barbar template tree
branch for installation into new boot cell root filesystems before the network filesystems of the
new cell have been established (at which point the barbar directory can begin being updated by SOP
rsync operations to keep it in sync with the rest of the commonwealth).  Basebarbar trees are
usually installed when installing their parent basebar trees into the target root filesystems
installed using the parent PP-stick, via a single cp command; e.g.,
  /bin/cp  -a  /_/l/PP/base/_  /_/l/NewRootFS
followed by adjusting all the README.thisdir files therein to document their promotions; especially
the
  basebar    -> rootbar and
  basebarbar -> barbar
transitions.
#---------------------------------------------------------------------------------------------------
```

7.5.1.8. How Is Change Control Effected—README.change-ctl

```
# CCIA_0.0    /_/_/README.change-ctl ..4....:....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#---------------------------------------------------------------------------------------------------
In order to minimize the complexity of platform maintenance and migration, strive for all
modifications to platform-canonical files and directories to reside somewhere within the rootbar
branch of the platform's logical filesystem.  Here are some examples of making this so:

  Instead of changing /etc/fstab during installation, rename /etc/fstab to /etc/fstab.orig and
  create a symlink for /etc/fstab pointing to the configured version herein, which of course must
  reside in the root filesystem; i.e., /_/h/xyzzy/l/XD/etc/fstab.  Note the insertion of the label
  layer in the path to permit distinguishing between different root filesystems available to the
  same host.  If another root filesystem for the same host exists; e.g., [XE], that can be mounted
  into the [XD] root filesystem as /_/l/XE.  Then the [XD] filesystem's /_/h/xyzzy/l/XE inode can be
  a symlink to ../l/XE to access the [XE] version of /etc/fstab, enabling /_/h/xyzzy/l/XE/etc/fstab
  to be a dependable path to the file inode regardless of which root filesystem has been booted.

  If /usr is a separate local filesystem; e.g., [XU], that should be mounted as /_/l/XU.  Then /usr
  in all root filesystems for the host must be symlinks to /_/l/XU or ../_/l/XU unless a particular
  root filesystem maintains its own /usr tree.  These duplicate paths can be a source of path
  dereferencing loops so be careful to avoid creating such.  Note that the find command notes and
  skips over such duplications.

If some file tracked in the CRUX package database needs to be modified, renamed, or deleted in some
manner, if at all possible, that should be accomplished via some mechanism that tracks such
maintenance.  The package database will be ignorant of the true maintenance condition of such files.
Many prt-get pre-install and post-install scripts are guilty of such rogue activity, not to mention
the sysadmin's actions completely outside the prtpkg scope of system maintenance in tuning or
localizing a package's files.  Perhaps some tool needs to be developed that enables CRUX to
definitely and completely maintain the history of all files within the purview of the package
database, although that will not appear to be simple.  Until that time, at minimum, scrupulously
maintain a chronological text file noting:

1. the timestamp,
2. the file(s) modified, deleted, moved, replaced with a symlink to ???; etc.,
3. who made the change,
4. how/what made the change, and
5. why the change was made.

Including the location of an automated tool that can back out the change(s) may someday save
somebody, maybe even you, a lot of work, possibly more than once.
#---------------------------------------------------------------------------------------------------
```

7.5.1.9. How To Change the CCI—README.cci-variations

```
# CCIA_0.0    /_/_/README.cci-variations ...:....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#-------------------------------------------------------------------------------------------------
Variations of a distributed CCI must be implemented by forking the basis CCI as described in Section
4.2 of the CRUX 3.4 Commonwealth Manual for the basis CCI.  Each CCI needs its own unique name, and
those starting with CCI are reserved for the default CCIs as they are scratched out according to
demand.  This information helps everyone involved with the enterprise's system administration as
well as non-privileged users to understand and comply with the standards.
#-------------------------------------------------------------------------------------------------
```

7.5.2. CCS README.thisdir* Files

7.5.2.1. README.thisdir File in PP-stick Trunk Directory

```
# CCIA_0.0    /_/l/PP/README.thisdir ..4....:....5....:....6....:....7....:....8....:....9....:....0
# ************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#-------------------------------------------------------------------------------------------------
This directory is the root directory of a PP-stick which is used for installing CRUX with the
version of the Commonwealth Enhancement identified at the beginning of the first line.  The
Enhancement is primarily documented in the LibreOffice Writer
CRUX_3.4_Commonwealth_Manual_CCIA_0.0.odt file, which may be exported into a PDF as well if you
trust the LibreOffice PDF-writing function.

The "base" subdirectory of this directory contains various branches of files intended to be
customized as useful for particular hosts and even particular root filesystems of hosts in which
CRUX with the Extensions will be installed.  The base/_ (aka basebar) is intended as the template
for installation of the rootbar directory into the root filesystems created using this particular
PP-stick.  Thus, the template README files to be initially installed into a root filesystem's barbar
tree branch using this PP-stick (before SOP rsyncs can be started) are found in this PP-stick's
basebarbar subdirectory; i.e., /_/l/PP/base/_/_.

The "BUILDDOC" subdirectory contains tools, documents, and logs pertaining to the creation of the
USB-stick containing this PP filesystem, the PA partition (template ESP), etc.
#-------------------------------------------------------------------------------------------------
```

7.5.2.2. README.thisdir File in PP-stick base Directory

```
# CCIA_0.0    /_/l/PP/base/README.thisdir ..:....5....:....6....:....7....:....8....:....9....:....
# ***********************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***********************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization.............
#------------------------------------------------------------------------------------------------
This directory is the base directory of a PP-stick which is used for installing CRUX with the CCI
instance of the Commonwealth Enhancement identified at the beginning of the first line of this
file.  It contains files and various branches of files likely to be customized as useful for
particular hosts (and even their particular root or boot filesystems) in which CRUX with this
version of the Extension will be installed.

This directory is similar in purpose to its sibling directory "bin", but its files are expected to
likely need, at the platform and root filesystem levels, tailoring that does not impact the
overriding CCI.  Nonetheless, it is needful when modifying these components to be alert to falling
out of conformance to the CCI, which necessitates forking the CCI and documenting such changes in
the new CCI.  It is better to have performed such analysis and prepared any new CCI and
corresponding PP-stick before booting the ISO to start an installation.

This directory's _ subdirectory (aka basebar) is designed as the template for installation of the
rootbar tree branch into the root filesystems created using this particular PP-stick.  Thus, the
```

```
template CRUX Commonwealth Standards (CCS) README files to be installed into a root filesystem's
rootbarbar tree branch using this PP-stick are found in this PP-stick's base/_/_ directory which
will be called the basebarbar directory of the PP-stick.
#-----------------------------------------------------------------------------------------------
```

7.5.2.3. README.thisdir File in PP-stick base/_ Directory

```
# CCIA_0.0    /_/l/PP/base/_/README.thisdir :....5....:....6....:....7....:....8....:....9....:....0
# ************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#-------------------------------------------------------------------------------------------------
This directory, base/_, aka basebar, resides only in PP-sticks.  It is the template rootbar tree
branch for installation into new boot cell root filesystems as the rootbar tree branch.  The file
/_/_/README.rootbar contains more details about what a rootbar tree branch entails.
#-------------------------------------------------------------------------------------------------
```

7.5.2.4. README.thisdir.rootbar File in PP-stick base/_ Directory

```
# CCIA_0.0    /_/README.thisdir ..:....4....:....5....:....6....:....7....:....8....:....9....:....0
# ************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#-------------------------------------------------------------------------------------------------
This directory, /_, aka root_ and rootbar, resides in the root filesystem of a platform using the
CRUX Commonwealth Standards (CCS), specifically the CCI version identified at the beginning of this
file's first line.  The files /_/_/README.* and other README.thisdir files contain details about
what that entails, with /_/_/README.rootbar specifically discussing the rootbar directory.
#-------------------------------------------------------------------------------------------------
```

7.5.2.5. README.thisdir File in PP-stick base/_/_ Directory

```
# CCIA_0.0    /_/l/PP/base/_/_/README.thisdir ...5....:....6....:....7....:....8....:....9....:....0
# ************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#-------------------------------------------------------------------------------------------------
This directory, base/_/_, aka basebarbar, resides only in PP-sticks.  It is the template barbar tree
branch for installation into new boot cell root filesystems before the network filesystems of the
new cell have been established (at which point the barbar directory can begin being updated by SOP
rsync operations to keep it in sync with the rest of the commonwealth.  The file /_/_/README.barbar
contains more details about what a barbar tree branch entails.
#-------------------------------------------------------------------------------------------------
```

7.5.2.6. README.thisdir.barbar File in PP-stick base/_/_ Directory

```
# CCIA_0.0    /_/_/README.thisdir :....4....:....5....:....6....:....7....:....8....:....9....:....0
# ************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#-------------------------------------------------------------------------------------------------
This directory, /_/_, aka barbar and rootbarbar, resides in the root filesystem of a platform using
the version of the CCS identified in the first line.  The file /_/_/README.barbar contains details
about what that entails.
```

```
#-----------------------------------------------------------------------------------------------
```

7.5.2.7. README.thisdir File in PP-stick bin Directory

```
# CCIA_0.0    /_/1/PP/bin/README.thisdir ...:....5....:....6....:....7....:....8....:....9....:....0
# ***********************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***********************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#-----------------------------------------------------------------------------------------------
This directory is the bin directory of a PP-stick which is used for installing CRUX with the version
of the Commonwealth enhancement identified at the beginning of the first line.  It is similar in
purpose to its sibling directory "base", differing by containing only executable files needed by the
ISO/CME systems considered either not customizable (particulary prtpkg_source) or unlikely to
require any modification.  Note modifications made to files in this directory are likely to require
forking the CCS and documenting such changes in the new CCS.
#-----------------------------------------------------------------------------------------------
```

7.6. PP-stick bin Directory Files

7.6.1. batlog

```
#!/bin/bash
# CCIA_0.0    PP-stick bin/batlog :....4....:....5....:....6....:....7....:....8....:....9....:....0
# ***********************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***********************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#-----------------------------------------------------------------------------------------------
# Append one or more messages per key argument(s) to the battery status log file
# (/root/batstat.log).  Note for lastu to get the crashed uptime, batlog lastu must be invoked
# (within batstat mon) before batuptime is started (within batstat mon).
#-----------------------------------------------------------------------------------------------
set +xv
# if   test ! -f /sys/class/power_supply/BAT0/status
# then echo 'This platform does not have a battery installed.'
#      exit 16
# fi
if   test    -x /root/batlog  \
          -a -x /root/batstat \
          -a -x /root/batuptime
then :
elif test    -f /_/1/PP/\[PP\] \
          -a -x /_/1/PP/bin//batlog  \
          -a -x /_/1/PP/bin//batstat \
          -a -x /_/1/PP/bin//batuptime
then cp -p /_/1/PP/bin/bat* /root
else echo 'Cound not find the bat scripts--aborting!'
     exit 32
fi
while test $# -gt 0
do case ".$1" in
     '.lastu' ) msg="Last uptime before boot"    ;;
     '.start' ) msg="/root/batstat mon started"  ;;
     '.stop'  ) msg="/root/batstat mon stopped"  ;;
     '.cin'   ) msg="Charger connected"          ;;
     '.cout'  ) msg="Charger disconnected"       ;;
     '.rboot' ) msg="Shutdown -reboot"           ;;
     '.rseat' ) msg="Reseat battery"             ;;
     '.halt'  ) msg="Shutdown -halt (power off)" ;;
     '.poff'  ) msg="Power off, no shutdown"     ;;
     '.pcut'  ) msg="Power cut off, no shutdown" ;;
     '.pon'   ) msg="Power on"                   ;;
```

```
      '.edit'  ) msg="Clean up /root/batstat.log" ;;
      *) echo 'Use lastu start stop cin cout rboot rseat halt poff pcut pon edit'
         exit 32
         ;;
    esac
    lastuptm=`/bin/cat /root/batstat.uptime`
    /usr/bin/gawk -v msg="$msg" -v lastuptm="$lastuptm" '\
      END {\
        printf("%-41.41s %s %s\n",\
               msg,\
               strftime("%Y/%m/%d_%H:%M:%S_UT_%a_%b", systime(), "UT"),\
               lastuptm\
              ) ;\
      }' \
      /dev/null \
    | /usr/bin/tee -a /root/batstat.log
    shift
done
/bin/sync
```

7.6.2. batstat

```
#!/bin/bash
# CCIA_0.0    PP-stick bin/batstat ....4....:....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# This script reports on the platform's battery status, either as a one-and-done invocation (no
# arguments) or ($1 is 'mon') as a status reporter that logs all changes while reporting them,
# usually as a session background job.
#----------------------------------------------------------------------------------------------------
set +xv
do_batstat() {
  test_msg=''
  test -f /root/batpresent \
    && BAT_OLD=`/bin/cat /root/batpresent` \
    || BAT_OLD=''
  test -d /sys/class/power_supply/BAT0 \
    && BAT_NEW='Y' \
    || BAT_NEW='N'
  echo $BAT_NEW >/root/batpresent
  if   test ".$BAT_OLD" = '.'
  then test $BAT_NEW = 'Y' \
         && test_msg="Battery found `/bin/cat /sys/class/power_supply/BAT0/status`"  \
         || test_msg='No battery found'
  elif test $BAT_NEW != $BAT_OLD
  then test $BAT_NEW = 'Y' \
         && test_msg="Battery inserted `/bin/cat /sys/class/power_supply/BAT0/status`" \
         || test_msg="Battery removed"
  fi
  if   test ".$test_msg" != '.'
  then /usr/bin/gawk -v msg="$test_msg" -v mon=$mon -v id=$$ '\
         END {\
           ol = sprintf("%-41.41s", msg) ;\
           if ( mon == "T" ) print ol >  "/tmp/batstat." id ;\
           else              print ol >> "/dev/stdout"      ;\
         }' /dev/null
  fi
  test -s /tmp/batstat.$$ \
    && print_stat
  if   test ".`/bin/cat /root/batpresent`" = '.Y'
  then echo `/bin/cat /sys/class/power_supply/BAT0/{status,charge_{now,full}}` \
       | /usr/bin/gawk -v mon=$mon -v id=$$ '\
           {\
             pc = $2 / $3 * 100.000 ;\
             alarm = pc < 15.0 ? "!!!" : "   " ;\
             if      ( pc == 100.0 ) level = "100.00000%" ;\
             else if ( pc < 10.0   ) level = sprintf("  %7.5f%%", pc) ;\
                  else               level = sprintf(" %8.5f%%", pc) ;\
             ol = sprintf("Battery Status %-11.11s %s %s", $1, level, alarm) ;\
             if ( mon == "T" ) print ol >  "/tmp/batstat." id ;\
             else              print ol >> "/dev/stdout"      ;\
           }'
       test -s /tmp/batstat.$$ \
         && print_stat
  fi
}
print_stat() {
  lastuptm=`/bin/cat /root/batstat.uptime`
  /usr/bin/gawk -v ls="$laststat" -v lu="$lastuptm" '\
    { if ( $0 != ls ) {\
        printf("%s %s %s\n",\
               $0,\
               strftime("%Y/%m/%d_%H:%M:%S_UT_%a_%b",\
                        systime(),\
                        "UT"\
                       ),\
               lu\
              ) ;\
        system("/bin/touch /tmp/batsync");\
      }\
    }' \
    /tmp/batstat.$$ 2>&1 \
  | /usr/bin/tee -a /root/batstat.log
  /bin/sync
  laststat=`/bin/cat /tmp/batstat.$$`
  : > /tmp/batstat.$$
}
handle_trap() {
  trap - 0 1 2 3 15
  /root/batlog stop
  rm -f /root/bat{stat.mon,present}
  exit 0
}
# Mainline logic:
if   test    -x /root/batlog  \
          -a -x /root/batstat \
          -a -x /root/batuptime
then :
elif test    -f /_/l/PP/\[PP\] \
          -a -x /_/l/PP/bin//batlog  \
          -a -x /_/l/PP/bin//batstat \
          -a -x /_/l/PP/bin//batuptime
then cp -p /_/l/PP/bin/bat* /root
elif test    -f /_/PP/\[PP\] \
          -a -x /_/PP/bin//batlog  \
          -a -x /_/PP/bin//batstat \
          -a -x /_/PP/bin//batuptime
then cp -p /_/PP/bin/bat* /root
else echo 'Cound not find the bat scripts--aborting!'
     exit 32
fi

if   test ".$1" != '.mon'
then mon='F'
     do_batstat
     test -f /root/batstat.mon \
       || /bin/rm -f /root/batpresent
else mon='T'
     if   test -f /root/batstat.mon
     then echo 'batstat mon is already running'
          exit 2
     else : > /root/batstat.mon
     fi
     test -f /tmp/batuploop || /root/batuptime
     test -f /tmp/batuploop || echo 'Warning! batuploop is not running'
     /root/batlog start
     trap 'handle_trap' 0 1 2 3 15
     laststat=''
     while :
     do do_batstat
        /bin/sleep 6s
     done
fi
```

7.6.3. batuploop

```
#!/bin/bash
# CCIA_0.0    PP-stick bin/batuploop ..4....:....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# This script is an orphaned batch job that updates /root/batstat.uptime with the output of
# uptime -p every 6 seconds.  The idea is to provide the current uptime to the /root/batstat script
# while it is running and to provide the uptime of a system crash following a reboot.  This script
# is intended to be launched only once by the first invocation of /root/batstat mon following a
# reboot.
#----------------------------------------------------------------------------------------------------
set +xv
if   test -f /tmp/batuploop
then echo "batuploop invoked but /tmp/batuploop already exists--aborting!"
     exit 32
fi
echo "Running batuploop ($0) in PID<$$>"
echo $$ > /tmp/batuploop
trap 'trap - 0 1 2 3 15 && echo "batuploop ending" && rm -f /tmp/batuptime && exit' 0 1 2 3 15
while :
do /usr/bin/uptime -p >/root/batstat.uptime 2>&1
   sleep 6s
done >> /tmp/batuploop.log 2>&1
```

7.6.4. batuptime

```
#!/bin/bash
# CCIA_0.0    PP-stick bin/batuptime ..4....:....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# This script spawns an orphaned batch job that updates /root/batstat.uptime with the output of
# uptime -p every 6 seconds.  The idea is to provide the current uptime to the /root/batstat script
# while it is running and to provide the uptime of a system crash following a reboot.  This script
# is intended to be launched only once by the first invocation of /root/batstat mon following a
# reboot.
#----------------------------------------------------------------------------------------------------
set +xv
echo "Running batuptime ($0) in PID<$$>"
if   test    -x /root/batlog  \
          -a -x /root/batstat \
          -a -x /root/batuptime
then :
elif test    -f /_/l/PP/\[PP\] \
          -a -x /_/l/PP/bin/batlog  \
          -a -x /_/l/PP/bin/batstat \
          -a -x /_/l/PP/bin/batuptime
then cp -p /_/l/PP/bin/bat* /root
elif test    -f /_/PP/\[PP\] \
          -a -x /_/PP/bin/batlog  \
          -a -x /_/PP/bin/batstat \
          -a -x /_/PP/bin/batuptime
then cp -p /_/PP/bin/bat* /root
else echo 'Cound not find the bat scripts--aborting!'
     exit 32
fi
getpsl() {
  psl=`/bin/ps -A -F -H -w\
       | gawk '/[0-9]   \/bin\/bash \/root\/batuptime$/ && $3 == 1 {printf("%u\n", $2)}'`
  # echo "getpsl set psl to <$psl>" 1>&2
}
echo "On entering batuptime, /root/batstat.uptime<`cat /root/batstat.uptime`>" 1>&2
getpsl
if   test    ! -f  /tmp/batuploop \
          -a ".$psl" = '.'
then /root/batlog lastu
     echo "Launching batuploop logging to /tmp/batuptime.$$.log"
     /root/batuploop >/tmp/batuptime.$$.log 2>&1 &
     sleep 2s
     if   test -f     /tmp/batuploop
     then /bin/ls -lh /tmp/batuploop
          echo "/tmp/batuploop contains <`cat /tmp/batuploop`>"
     fi
     getpsl
     echo "PIDs running /bin/bash /root/batuptime as orphan(s): $psl"
     exit 1
fi
```

7.6.5. chrony_post-install

```
#!/bin/sh
# CCIA_0.0    PP-stick bin/chrony_post-install ..5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2000-2018 by Per Lidén and the CRUX development team, and licensed through the GNU
# * General Public License available at http://www.gnu.org/copyleft/gpl.html.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# This is the CRUX 3.4 post-install script for port opt/chrony.
#----------------------------------------------------------------------------------------------------

chown -R chrony:chrony /var/lib/chrony /var/log/chrony
```

7.6.6. chrony_pre-install

```
#!/bin/sh
# CCIA_0.0    PP-stick bin/chrony_pre-install ...5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2000-2018 by Per Lidén and the CRUX development team, and licensed through the GNU
# * General Public License available at http://www.gnu.org/copyleft/gpl.html.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# This is the CRUX 3.4 pre-install script for port opt/chrony.
#----------------------------------------------------------------------------------------------------

getent group chrony || /usr/sbin/groupadd -g 55 chrony
getent passwd chrony || /usr/sbin/useradd -g chrony -u 55 -d /var/lib/chrony -s /bin/false chrony
/usr/bin/passwd -l chrony
```

7.6.7. editfile

```
#!/bin/dash
# CCIA_0.0    PP-stick bin/editfile ...4....:....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#---------------------------------------------------------------------------------------------------
# For the specified source file, create the associated target file ($tf) from
#   /_/l/PP/h/$PPHOST/l/$PPROTL/$1 if it exists; otherwise, from
#   /_/l/PP/h/$PPHOST/$1           if it exists; otherwise, from
#   /_/l/PP/base/$1.
# Next, let the user modify the target file using $EDITOR.  Then, put the diffs between
# /_/l/PP/base/$1 and the associated target file into the /_/l/PP/h/$PPHOST/l/$PPROTL/$1.diffs file
# if $PPROTL is known; otherwise, save it as /_/l/PP/h/$PPHOST/$1.diffs, and, if there aren't any
# diffs, rm the diffs file, else copy the target file into the same directory as the diffs file but
# named $1.
#---------------------------------------------------------------------------------------------------
set +xv
test    ".$EDITOR" != '.vi' \
     -a ".$EDITOR" != '.nano' \
  && { echo "The EDITOR environment variable is missing or unsupported--aborting!"
       exit 32
     }
test    $# -ne 1 \
     -o ".$1" = '.' \
  && { echo "Specify the source filename to upgrade as the only argument."
       exit 1
     }
case "$1" in # in collating sequence
  '.vimrc'                       ) tf='/root/.vimrc'                     ;;
  '.toprc'                       ) tf='/root/.toprc'                     ;;
  'PPvars'                       ) tf='/root/PPvars'                     ;;
  'etc.chrony.conf'              ) tf='/etc/chrony.conf'                 ;;
  'etc.fstab'                    ) tf='/etc/fstab'                       ;;
  'etc.hosts'                    ) tf='/etc/hosts'                       ;;
  'etc.ntpd.conf'                ) tf='/etc/ntpd.conf'                   ;;
  'etc.pkgadd.conf'              ) tf='/etc/pkgadd.conf'                 ;;
  'etc.pkgmk.conf'               ) tf='/etc/pkgmk.conf'                  ;;
  'etc.prt-get.conf'             ) tf='/etc/prt-get.conf'                ;;
  'etc.rc.conf'                  ) tf='/etc/rc.conf'                     ;;
  'etc.rc.d.net'                 ) tf='/etc/rc.d/net'                    ;;
  'etc.resolv.conf'              ) tf='/etc/resolv.conf'                 ;;
  'extlinF1.msg'                 ) tf='/boot/extlinF1.msg'               ;;
  'extlinF2.msg'                 ) tf='/boot/extlinF2.msg'               ;;
  'extlinux.conf'                ) tf='/boot/extlinux.conf'              ;;
  'prtpkg_altinstall'            ) tf='/root/prtpkg_altinstall'          ;;
  'prtpkg_altinstall.txt'        ) tf='/root/prtpkg_altinstall.txt'      ;;
  'prtpkg_chroot'                ) tf='/root/prtpkg_chroot'              ;;
  'prtpkg_getdrives.awk'         ) tf='/root/prtpkg_getdrives.awk'       ;;
  'prtpkg_locking.dash.source'   ) tf='/root/prtpkg_locking.dash.source' ;;
  'prtpkg_mkfs'                  ) tf='/root/prtpkg_mkfs'                ;;
  'prtpkg_mount'                 ) tf='/root/prtpkg_mount'               ;;
  'prtpkg_pdrive'                ) tf='/root/prtpkg_pdrive'              ;;
  'prtpkg_presetup'              ) tf='/root/prtpkg_presetup'            ;;
  'prtpkg_root.profile'          ) tf='/root/prtpkg_root.profile'        ;;
  'prtpkg_setup'                 ) tf='/root/prtpkg_setup'               ;;
  'prtpkg_shortcuts.bash.source' ) tf='/root/prtpkg_shortcuts.bash'      ;;
  'var.lib.pkg.prt-get.aliases'  ) tf='/var/lib/pkg/prt-get.aliases'     ;;
  *) echo "Specified source filename is unsupported--aborting!"
     exit 32
     ;;
esac
test -f "/_/l/PP/base/$1" \
  || { echo "Specified template file does not exist in /_/l/PP/base--aborting!"
       exit 32
     }
echo "Upon entering editfile, PPROTL<$PPROTL>"
if   test    ".$PPROTL" != '.'       \
          -a ".$PPROTL" != '.ISORAM'
then sd="/_/l/PP/h/$PPHOST/l/$PPROTL"
     if   test -d $sd
     then /bin/mkdir -p $sd
          echo "editfile created directory $sd"
     fi
     if   test -f "$sd/$1"
     then /bin/cp -p "$sd/$1" "$tf"
```

```
          echo "Editfile of $1 based upon $sd/$1"
     elif test -f "/_/l/PP/h/$PPHOST/$1"
     then /bin/cp -p "/_/l/PP/h/$PPHOST/$1" "$tf"
          echo "Editfile of $1 based upon /_/l/PP/h/$PPHOST/$1"
     else /bin/cp -p "/_/l/PP/base/$1"      "$tf"
          echo "Editfile of $1 based upon /_/l/PP/base/$1"
     fi
else sd="/_/l/PP/h/$PPHOST"
     if test -f "$sd/$1"
     then /bin/cp -p "/$sd/$1" "$tf"
          echo "Editfile of $1 based upon $sd/$1"
     else /bin/cp -p "/_/l/PP/base/$1"      "$tf"
          echo "Editfile of $1 based upon /_/l/PP/base/$1"
     fi
fi
$EDITOR "$tf"
/usr/bin/diff -C 1 "/_/l/PP/base/$1" "$tf" \
              > "$sd/$1.diffs" 2>&1
if   test $? -eq 0
then /bin/rm -f "$sd/$1.diffs"
     echo "editfile removed $sd/$1.diffs"
else /bin/cp -p "$tf" "$sd/$1"
     echo "editfile stowed $sd/$1"
fi
```

7.6.8. lsdisks (and partitioning information)

This ***bash*** script generates just about everything you might need to know about a platform's disk devices (except `RAID` and `lvm` support still need to be developed). With a `$1` argument, it writes the information using a script-friendly format (similar to output of the ***parted***'s `-m` parameter) into the `filename` provided). With no arguments, it produces a human-readable report something like the following:

```
NAME     TYPE   SIZE STATE   TRAN   RO RM  RA ROTA HOTPLUG PHY-SEC LOG-SEC MIN-IO OPT-IO ALIGNMENT
/dev/sda disk  59.6G running sata    0  0 128    0       0     512     512    512      0         0
/dev/sdb disk 223.6G running sata    0  0 128    0       0     512     512    512      0         0
/dev/sdc disk        running usb     0  1 128    1       1     512     512    512      0         0
/dev/sdd disk        running usb     0  1 128    1       1     512     512    512      0         0
/dev/sde disk        running usb     0  1 128    1       1     512     512    512      0         0
/dev/sdf disk        running usb     0  1 128    1       1     512     512    512      0         0
/dev/sdg disk   7.2G running usb     0  1 128    1       1     512     512    512      0         0
NAME        LABEL PARTLABEL PARTFLAGS FSTYPE MOUNTPOINT
/dev/sda
├─/dev/sda1       B0
├─/dev/sda2  BB   BB        0x4       ext4   /boot
├─/dev/sda3  BM   BM                  ext4
└─/dev/sda4  BD   BD                  ext4   /
/dev/sdb
├─/dev/sdb1       Y0
├─/dev/sdb2       Y1                  swap   [SWAP]
├─/dev/sdb3       Y2                  swap   [SWAP]
├─/dev/sdb4       Y3                  swap   [SWAP]
├─/dev/sdb5       Y4                  swap   [SWAP]
├─/dev/sdb6       Y5                  swap   [SWAP]
├─/dev/sdb7       Y6                  swap   [SWAP]
├─/dev/sdb8       Y7                  swap   [SWAP]
├─/dev/sdb9       Y8                  swap   [SWAP]
├─/dev/sdb10      Y9                  swap   [SWAP]
├─/dev/sdb11 YA   YA                  vfat
├─/dev/sdb12 YB   YB                  ext4
├─/dev/sdb13 YP   YP                  ext4
├─/dev/sdb14 YD   YD                  ext4
├─/dev/sdb15 YE   YE                  ext4
└─/dev/sdb16 YM   YM                  ext4
/dev/sdg
├─/dev/sdg1  P0   P0                  ext2
├─/dev/sdg2  PA   PA                  vfat
├─/dev/sdg3  PP   PP                  ext2   /_/PP
```

```
└─/dev/sdg4  PZ    PZ                vfat
-Type|Label Device-path --First-sector ---Last-sector ------Size(KiB) ---Size(MiB) Size(GiB) Name
        gpt /dev/sda                 0      125045423     62522712.00     61057.34     59.63
         B0 /dev/sda1               34           2047         1007.00         0.98      0.00 B0
         BB /dev/sda2             2048         524287       261120.00       255.00      0.25 BB
         BM /dev/sda3           524288        6815743      3145728.00      3072.00      3.00 BM
         BD /dev/sda4          6815744      125045390     59114823.50     57729.32     56.38 BD
        gpt /dev/sdb                 0      468862127    234431064.00    228936.59    223.57
         Y0 /dev/sdb1               34           2047         1007.00         0.98      0.00 Y0
         YA /dev/sdb11            2048         524287       261120.00       255.00      0.25 YA
         YB /dev/sdb12          524288        1048575       262144.00       256.00      0.25 YB
         Y1 /dev/sdb2          1048576        2097151       524288.00       512.00      0.50 Y1
         Y2 /dev/sdb3          2097152        3145727       524288.00       512.00      0.50 Y2
         Y3 /dev/sdb4          3145728        4194303       524288.00       512.00      0.50 Y3
         Y4 /dev/sdb5          4194304        5242879       524288.00       512.00      0.50 Y4
         Y5 /dev/sdb6          5242880        6291455       524288.00       512.00      0.50 Y5
         Y6 /dev/sdb7          6291456        7340031       524288.00       512.00      0.50 Y6
         Y7 /dev/sdb8          7340032        8388607       524288.00       512.00      0.50 Y7
         Y8 /dev/sdb9          8388608        9437183       524288.00       512.00      0.50 Y8
         Y9 /dev/sdb10         9437184       10485759       524288.00       512.00      0.50 Y9
         YP /dev/sdb13        10485760      117440511     53477376.00     52224.00     51.00 YP
         YM /dev/sdb16       117440512      134217727      8388608.00      8192.00      8.00 YM
         YD /dev/sdb14       134217728      301989887     83886080.00     81920.00     80.00 YD
         YE /dev/sdb15       301989888      468862094     83436103.50     81480.57     79.57 YE
        gpt /dev/sdg                 0       15138815      7569408.00      7392.00      7.22
         P0 /dev/sdg1               34           2047         1007.00         0.98      0.00 P0
         PA /dev/sdg2             2048         526335       262144.00       256.00      0.25 PA
         PP /dev/sdg3           526336       14612479      7043072.00      6878.00      6.72 PP
         PZ /dev/sdg4         14612480       15136767       262144.00       256.00      0.25 PZ
            /dev/sdg--        15136768       15138782         1007.50         0.98      0.00
```

The RM boolean defines disk removability, the RO boolean indicates R/O versus R/W capability, RA means Read Ahead, and ROTA (short for rotating) more accurately means fast (0) or slow (1). Any entry in the final section whose Device-path ends with two hyphens is free space, such as the last line in the example. Extents not shown are unavailable areas (see the enhanced ***parted*** listings below for details of these unreported sectors for GPT drives).

Before making changes to the platform's drives and/or adding/removing entire drives, save a report via shell redirection beforehand; e.g.,

```
/_/l/PP/bin/lsdisks > /tmp/before
```

make the changes, then rerun ***lsdisks*** redirecting into a different file and run ***diff*** to show the actual changes that occurred; e.g.,

```
/_/l/PP/bin/lsdisks > /tmp/after
/usr/bin/diff -C 1 /tmp/{before,after}
```

The following expanded (the Notes column) ***parted*** listings show how the disk sectors unmentioned by ***parted*** (and ***lsdisks***) are allocated (all Start, End, and Size values are in 512-byte sectors) and also shows different partition flags for the ESP and /boot partitions:

```
Model: ATA SSD2SC120G1CS175 (scsi)
Disk /dev/sda: 234441648s
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:
PN Start---- End------ Size----- File_system--- Name Flags-------- Notes-------------------------
   0         0         1         PMBR                              Protective MBR
   1         1         1         PGH                              Primary GPT Header
   2         33        32        PPEA                             Primary Partition Entry Array
 1 34        2047      2014                     Z0   bios_grub     Reserved free space for GRUB
11 2048      524287    522240    fat32          ZA   msftdata, esp Active EFI System Partition
12 524288    1048575   524288    ext4           ZB                 Inactive BIOS boot partition
 2 1048576   2097151   1048576   linux-swap(v1) Z1
 3 2097152   3145727   1048576   linux-swap(v1) Z2
 4 3145728   4194303   1048576   linux-swap(v1) Z3
 5 4194304   5242879   1048576   linux-swap(v1) Z4
 6 5242880   6291455   1048576   linux-swap(v1) Z5
 7 6291456   7340031   1048576   linux-swap(v1) Z6
 8 7340032   8388607   1048576   linux-swap(v1) Z7
 9 8388608   9437183   1048576   linux-swap(v1) Z8
10 9437184   10485759  1048576   linux-swap(v1) Z9
13 10485760  234440703 223954944 ext4           ZD
   234440704 234441614 911       Free Space
   234441615 234441646 32        SPEA                             Secondary Partition Entry Array
   234441647 234441647 1         SGH                              Secondary GPT Header

Model: ATA Corsair Force GT (scsi)
Disk /dev/sdb: 468862128s
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:
PN Start---- End------ Size----- File_system--- Name Flags-------- Notes-------------------------
   0         0         1         PMBR                             Protective MBR
   1         1         1         PGH                              Primary GPT Header
   2         33        32        PPEA                             Primary Partition Entry Array
 1 34        2047      2014                     Y0   bios_grub     Reserved free space for GRUB
11 2048      524287    522240    fat32          YA   msftdata      Inactive ESP System Partition
12 524288    1048575   524288    ext4           YB   legacy_boot   Active BIOS boot partition
 2 1048576   2097151   1048576   linux-swap(v1) Y1
 3 2097152   3145727   1048576   linux-swap(v1) Y2
 4 3145728   4194303   1048576   linux-swap(v1) Y3
 5 4194304   5242879   1048576   linux-swap(v1) Y4
 6 5242880   6291455   1048576   linux-swap(v1) Y5
 7 6291456   7340031   1048576   linux-swap(v1) Y6
 8 7340032   8388607   1048576   linux-swap(v1) Y7
 9 8388608   9437183   1048576   linux-swap(v1) Y8
10 9437184   10485759  1048576   linux-swap(v1) Y9
13 10485760  117440511 106954752 ext4           YP
16 117440512 134217727 16777216  ext4           YM
14 134217728 301989887 167772160 ext4           YD
15 301989888 468862094 166872207 ext4           YE
   468862095 468862126 32        SPEA                             Secondary Partition Entry Array
   468862127 468862127 1         SGH                              Secondary GPT Header
```

See https://en.wikipedia.org/wiki/GUID_Partition_Table for complete details of GPT drives and https://en.wikipedia.org/wiki/Master_Boot_Record for the formats of all MBR types (sector 0).

Getting partitions correctly aligned (important for I/O performance) is discussed at https://www.rodsbooks.com/gdisk/advice.html#alignment but for detailed "dark art" discussion see this article with its embedded link to this explanation. Often using a partition starting sector address that is evenly divisible by eight is sufficient but it is better to ensure the I/O is properly optimized; e.g., a ***lsblk -dt*** command will display the pertinent parameters. See the `--machine` (aka `-m`) flag of the ***parted*** command to use parsable print output format as documented within this notice; e.g. (note the *free space* entry at the end):

```
# parted /dev/sda -m unit s print free
BYT;
/dev/sda:234441648s:scsi:512:512:gpt:ATA SSD2SC120G1CS175:;
```

```
1:34s:2047s:2014s::Z0:bios_grub;
11:2048s:524287s:522240s:fat32:ZA:boot, esp, msftdata;
12:524288s:1048575s:524288s:ext4:ZB:;
2:1048576s:2097151s:1048576s:linux-swap(v1):Z1:;
3:2097152s:3145727s:1048576s:linux-swap(v1):Z2:;
4:3145728s:4194303s:1048576s:linux-swap(v1):Z3:;
5:4194304s:5242879s:1048576s:linux-swap(v1):Z4:;
6:5242880s:6291455s:1048576s:linux-swap(v1):Z5:;
7:6291456s:7340031s:1048576s:linux-swap(v1):Z6:;
8:7340032s:8388607s:1048576s:linux-swap(v1):Z7:;
9:8388608s:9437183s:1048576s:linux-swap(v1):Z8:;
10:9437184s:10485759s:1048576s:linux-swap(v1):Z9:;
13:10485760s:234440703s:223954944s:ext4:ZD:;
1:234440704s:234441614s:911s:free;
```

Also see the excellent online documentation of Rod Smith's `GPT Fdisk` tool and the link to its source code at http://www.rodsbooks.com/gdisk/—this is a much more capable tool than ***parted*** if needed, especially for repairs.

```
#!/bin/bash
# CCIA_0.0    PP-stick bin/lsdisks ....4....:....5....:....6....:....7....:....8....:....9....:....0
# ***********************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***********************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#-----------------------------------------------------------------------------------------------
# Show importnat particulars for all disks and partitions on the platform
#-----------------------------------------------------------------------------------------------
set +xv
/bin/rm -f /tmp/lsdisks.*
/bin/lsblk -dapo \
           NAME,TYPE,SIZE,STATE,TRAN,RO,RM,RA,ROTA,HOTPLUG,PHY-SEC,LOG-SEC,MIN-IO,OPT-IO,ALIGNMENT
/bin/lsblk -lapo NAME,LABEL,PARTLABEL,PARTFLAGS,FSTYPE,MOUNTPOINT
/bin/lsblk -lnpo NAME,TYPE \
| /usr/bin/gawk '\
$2 == "disk" { print $1 >> "/tmp/lsdisks.disks" } \
$2 == "part" { print $1 >> "/tmp/lsdisks.parts" } \
'
echo -n "-Type|Label Device-path --First-sector ---Last-sector "
echo    "------Size(KiB) ---Size(MiB) Size(GiB) Name"
#
# dsk[0]<>
# dsk[1]</dev/sdg>
# dsk[2]<15138816s>
# dsk[3]<scsi>
# dsk[4]<512>
# dsk[5]<512>
# dsk[6]<gpt>
# dsk[7]<Verbatim STORE N GO>
# dsk[8]<>
#
# prt[3][0]<>
# prt[3][1]<4>
# prt[3][2]<14612480s>
# prt[3][3]<15136767s>
# prt[3][4]<524288s>
# prt[3][5]<fat32>
# prt[3][6]<PZ>
# prt[3][7]<msftdata>
#
for ii in `cat /tmp/lsdisks.disks`
do  /usr/sbin/parted $ii -m unit s print free \
    | /usr/bin/gawk -v dev=$ii '\
      BEGIN { px = 0;\
              ef = "/dev/stderr";\
            }\
      NR == 1 { if ( $0 != "BYT;" ) {\
                     print "Encountered unsupported addressing line " $0 " -- aborting!" >> ef;\
                     exit 2;\
                  }\
              }\
      NR == 2 { ds = split($0, dsk, /[:;]/);\
                  # for (kk = 0 ; kk < ds ; kk++ ) print "dsk[" kk "]<" dsk[kk] ">";\
                  if ( dsk[6] != "gpt" && dsk[6] != "msdos" ) {\
                     print "Encountered unsupported disk organization " dsk[6] " -- aborting!" >> ef;\
                     exit 2;\
                  }\
                  if ( dsk[4] != 512 || dsk[5] != 512 ) {\
                     print "Encountered unsupported sector sizes " dsk[4] "/" dsk[5] " -- aborting!"\
                           >> ef;\
                     exit 2;\
                  }\
                  lss = 512.0;\
              }\
      NR > 2  { prt[px][0] = 'x' ; delete prt[px][0];\
                  ps[px] = split($0, prt[px], /[:;]/);\
                  # for (ll = 0 ; ll < ps[px] ; ll++ ) print "prt[" px "][" ll "]<" prt[px][ll] ">";\
                  if ( dsk[6] == "msdos" && prt[px][6] == "" && prt[px][5] != "free" ) {\
                    "lsblk -no LABEL " dsk[1] prt[px][1] | getline cmdl;\
                    if ( cmdl != "" ) prt[px][6] = cmdl;\
                  }\
                  if ( prt[px][5] == "free" ) prt[px][1] = "--";\
                  sub(/s/, "", prt[px][2]);\
                  sub(/s/, "", prt[px][3]);\
                  sizs = prt[px][4];\
                  sub(/s/, "", sizs);\
                  kibs[px] = sizs * lss / 1024.;\
                  mibs[px] = kibs[px]   / 1024.;\
                  gibs[px] = mibs[px]   / 1024.;\
                  # print "$0<" $0 "> px<" px ">";\
                  # printf("kibs<%15.2f> mibs<%12.2f> gibs<%9.2f>\n", kibs[px], mibs[px], gibs[px]);\
                  px++;\
              }\
      END { # print "px<" px ">";\
            sizd = dsk[2];\
            sub(/s/, "", sizd);\
            kibd = sizd * lss / 1024.;\
            mibd = kibd       / 1024.;\
            gibd = mibd       / 1024.;\
            printf("%11.11s %s    %14.14s %14.14s %15.2f %12.2f %9.2f\n",\
                        dsk[6], dsk[1], 0, sizd - 1, kibd, mibd, gibd);\
            for (jj = 0 ; jj < px ; jj++)\
                printf("%11.11s %s%-3.3s %14.14s %14.14s %15.2f %12.2f %9.2f %s\n",\
                        prt[jj][6], dsk[1], prt[jj][1], prt[jj][2], prt[jj][3],\
                        kibs[jj], mibs[jj], gibs[jj], prt[jj][6]);\
        }'
done
```

7.6.9. mk_PP-stick

```
#!/bin/dash
# CCIA_0.0    PP-stick bin/mk_PP-stick 4....:....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#---------------------------------------------------------------------------------------------------
# Prepare the replacement Verbatim black and gray USB stick as a GPT unit with:
#   1. an ext2 bios_grub partition,
#   2. a fat32 ESP partition,
#   3. an ext4 partition for CRUX platform support files, and
#   4. a 255 MB partition for clipping ESP partition images with platform-specific labels
#   to replace the original green USB-stick that catastrophicly failed Nov 1, 2017.
#   The dd files to restore from are in /_/l/PP/arcs: G0.dd, GA.dd, GE.dd,and GZ.dd.
#
# The PP USB disk should look very much like this when finished:
#
#   By sectors:
#      Model: Verbatim STORE N GO (scsi)
#      Disk /dev/sdc: 15138816s
#      Sector size (logical/physical): 512B/512B
#      Partition Table: gpt
#      Disk Flags:
#
#      Number  Start      End        Size       File system  Name  Flags
#       1      34s        2047s      2014s      ext2         P0    bios_grub
#       2      2048s      526335s    524288s    fat32        PA    boot, esp
#       3      526336s    14612479s  14086144s  ext4         PP
#       4      14612480s  15136767s  524288s    fat32        PZ
#              15136768s  15138782s  2015s      Free Space
#
#   By MiBs:
#      Model: Verbatim STORE N GO (scsi)
#      Disk /dev/sdc: 7392MiB
#      Sector size (logical/physical): 512B/512B
#      Partition Table: gpt
#      Disk Flags:
#
#      Number  Start    End      Size     File system  Name  Flags
#       1      0.02MiB  1.00MiB  0.98MiB  ext2         P0    bios_grub
#       2      1.00MiB  257MiB   256MiB   fat32        PA    boot, esp
#       3      257MiB   7135MiB  6878MiB  ext4         PP
#       4      7135MiB  7391MiB  256MiB   fat32        PZ
#              7391MiB  7392MiB  0.98MiB  Free Space
#-------------------------------------------------------------------------------------------------
umnt() { # $1 full path to unmount if mounted
test `/bin/df -alHT \
      | /usr/bin/grep "$1" \
      | /usr/bin/wc -l \
     ` -eq 1 \
  && /bin/umount "$1"
}
#-------------------------------------------------------------------------------------
test `/usr/bin/whoami` != root \
  && { echo 'Only root can use the script.'
       exit 2
     }
set -xv
test ! -f /_/ZH/PP_dd/GA.dd \
  && { echo 'Cannot find the ESP image (/_/ZH/PP_dd/GA.dd)--aborting!'
       exit 32
     }
test ! -f /_/ZH/PP_dd/GE.tar.bz2 \
  && { echo 'Cannot find the GE tarball (/_/ZH/PP_dd/GE.tar.bz2)--aborting!'
       exit 32
     }
did=`/usr/bin/lsusb -v -d 18a5:0243 2>/dev/null \
     | /usr/bin/gawk '/ iSerial /{print "usb-Verbatim_STORE_N_GO_"$3"-0:0"}' \
    `
D=`/bin/ls -lh /dev/disk/by-id/$did \
   | /bin/sed -e '/'$did'/s/^.*\///'
  `
test ! -b /dev/$D \
  && { echo 'Did not find Verbatim USB drive (did<$did>)--aborting!'
       exit 32
     }
echo "`date` -- Running $0 in PWD<$PWD>"
/bin/ls -lh $0
# Ensure everything is unmounted in case this is a restart
test ! -d  /_    \
  && /bin/mkdir /_
```

```
test ! -d  /_/P0 \
  && /bin/mkdir /_/P0
test ! -d  /_/PA \
  && /bin/mkdir /_/PA
test ! -d  /_/PP \
  && /bin/mkdir /_/PP
test ! -d  /_/PZ \
  && /bin/mkdir /_/PZ
for mp in /_/P0 \
          /_/PA \
          /_/PP \
          /_/PZ
do  umnt $mp
done
set -e
/bin/df -alHT
/bin/mount -l
#-------------------------------------------------------------------------------------
# Build the GPT partition table and its partition allocations:
#
/usr/sbin/parted -s /dev/$D mklabel gpt unit s \
mkpart P0   ext2              34       2047                   toggle  1 bios_grub \
mkpart PA   fat32           2048     526335 align-check opt  2 toggle  2 boot      \
mkpart PP   ext4          526336   14612479 align-check opt  3                     \
mkpart PZ   fat32       14612480   15136767 align-check opt  4                     \
print free unit GiB print free unit MiB print free quit
#-------------------------------------------------------------------------------------
# Build the bios_grub area (partition P0)
#
L='P0'
P='1'
T="/dev/${D}$P"
test `/bin/df \
      | /usr/bin/grep "/_/$L" \
      | /usr/bin/wc -l \
     ` -gt 0 \
  && /bin/umount /_/$L
test ! -d  /_/$L \
  && /bin/mkdir /_/$L
test `/bin/df \
      | /usr/bin/grep "$T" \
      | /usr/bin/wc -l \
     ` -gt 0 \
  && { echo "Encountered unexpected mount of $T"
       exit 8
     }
set +e
/sbin/mke2fs -v -t ext2 -r 1 -L $L -T default \
-O \
filetype,\
large_file,\
^resize_inode,\
sparse_super \
-c -b 4096 -I 256 -i 16384 -m 1 $T \
&& \
/sbin/tune2fs -c 256 -e remount-ro -i 6m -r 1024 -u 0 \
-o \
^acl,\
^debug,\
^bsdgroups,\
uid16,\
user_xattr \
-O \
dir_index,\
filetype,\
large_file,\
sparse_super \
$T
set -e
mount $T /_/$L
if   test `/bin/df -alHT \
          | /usr/bin/grep " /_/$L" \
          | /usr/bin/wc -l \
         ` -eq 1
then test ! -f "/_/$L/[$L]" \
       && /bin/touch -r /_/$L/lost+found /_/$L/\[$L\]
     /bin/df    -alHT \
     | /usr/bin/grep $T
     /bin/mount -lnf \
     | /usr/bin/grep $T
else echo "Partition $L failed to mount--aborting!"
     exit 32
fi
echo "Root directory of partition $L:"
/bin/ls -lah /_/$L
/bin/umount  /_/$L
/sbin/tune2fs -l $T
#--------------------------------------------------------------------------------
# Build the PA ESP (aka EFI) partition:
#
L='PA'
P='2'
T="/dev/${D}$P"
# LANG=C /sbin/mkfs.vfat -F 32 -n $L $T
/bin/dd if=/_/ZH/PP_dd/GA.dd of=$T bs=8192 count=32768
/sbin/fatlabel $T PA
/bin/mount $T /_/PA
test -f /_/PA/\[GA\] \
  && /bin/mv /_/PA/[GA\] /_/PA/\[PA\] \
  || { echo PA ID error
       exit 16
     }
echo "Root directory of partition $L:"
/bin/df -alHT
/bin/ls -lah /_/$L
/bin/umount  /_/$L
#--------------------------------------------------------------------------------
# Build the PP partition:
#
L='PP'
P='3'
T="/dev/${D}$P"
# /usr/sbin/parted -s /dev/${D} unit MiB print free
test `/bin/df \
      | /usr/bin/grep "/_/$L" \
      | /usr/bin/wc -l \
     ` -gt 0 \
  && /bin/umount /_/$L
test ! -d  /_/$L \
  && /bin/mkdir /_/$L
test `/bin/df \
      | /usr/bin/grep "$T" \
      | /usr/bin/wc -l \
     ` -gt 0 \
  && { echo "Encountered unexpected mount of $T"
       exit 8
     }
/sbin/mke2fs -v -t ext2 -r 1 -L $L -T default \
-O \
filetype,\
large_file,\
^resize_inode,\
sparse_super \
-b 4096 -I 256 -i 16384 -m 1 $T
set +e
/sbin/tune2fs -c 256 -e remount-ro -i 6m -r 1024 -u 0 \
-o \
^acl,\
^debug,\
^bsdgroups,\
uid16,\
user_xattr \
-O \
dir_index,\
filetype,\
large_file,\
sparse_super \
$T
set -e
/bin/mount $T /_/$L
if   test `/bin/df -alHT \
           | /usr/bin/grep " /_/$L" \
           | /usr/bin/wc -l \
          ` -eq 1
then test ! -f "/_/$L/[$L]" \
       && /bin/touch -r /_/$L/lost+found /_/$L/\[$L\]
     /bin/df    -alHT \
     | /usr/bin/grep $T
     /bin/mount -lnf \
     | /usr/bin/grep $T
else echo "Partition $L failed to mount--aborting!"
     exit 32
fi
/bin/umount  /_/$L
/sbin/tune2fs -l $T
/bin/mount $T /_/$L
test ! -f /_/$L/[$L] \
  && { echo "$L ID error"
       exit 16
     }
( cd   /_/PP \
    && /bin/tar xjf /_/ZH/PP_dd/GE.tar.bz2
)
test -f /_/PP/\[GE\] \
  && /bin/rm /_/PP/[GE\] \
  || { echo PP ID error
       exit 16
     }
echo "Root directory of partition $L:"
/bin/ls -lah /_/$L
#-------------------------------------------------------------------------------------
# Build the GZ ESP label clipping partition:
# The normal procedure starts with the template image, GZ, which is identical to the
# GA partition save it has GZ for the filesystem lable and [GZ] for the lable ID
# file.  Give the desire to create an ESP image to be instaled into a platform's
# partition, for example, WA, that is accomplished by changing the GZ partition's
# filesystem label to WA, then mounting the partition and changing the [GZ] label ID
# file to [WA].  Now that the partition is exactly what is wanted to install into
# the partition on the target platform, the image is copied into the GE partition as
# file WA.dd.  Finally the original condition of the GZ partition is restored.
#
# To establish the template partition, this script copies the GA partition to the GZ
# partition, sets the GZ filesystem label to GZ, mounts GZ, and changes its label ID
# file from [GZ] to [GZ].  As a precaution, a backup copy of the template is placed
# in the GE partition as file GZ.dd in case there is a need to recover it without
# the GA partition.
#
P='4'
T="/dev/${D}$P"
/bin/dd if=/dev/${D}2 of=/dev/${D}4 bs=8192 count=32768
/sbin/fatlabel $T PZ
/bin/mount $T /_/PZ
/bin/mv /_/PZ/\[PA\] /_/PZ/\[PZ\]
/bin/dd if=/dev/${D}4 of=/_/PP/PZ.dd bs=8192 count=32768
echo "Root directory of partition PZ:"
/bin/ls -lah /_/PZ
/bin/umount $T
/bin/df -alHT
```

7.6.10. mntdrvs

```
#!/bin/bash
# CCIA_0.0    PP-stick bin/mntdrvs ....4....:....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#---------------------------------------------------------------------------------------------------
# For each mounted /dev/hd* and /dev/sd*, report device, label, mount point, file type, and options.
#---------------------------------------------------------------------------------------------------
set +xv
/bin/mount -l \
| /usr/bin/gawk '\
$1 ~ /^\/dev\/[hs]d/ {\
  printf("%-10.10s %-20.20s %-8.8s %-12.12s %s\n", $1, $7, $5, $3, $6) ;\
}'
```

7.6.11. ntp_pre-install

```
#!/bin/sh
# CCIA_0.0    PP-stick bin/ntp_pre-install .:....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2000-2018 by Per Lidén and the CRUX development team, and licensed through the GNU
# * General Public License available at http://www.gnu.org/copyleft/gpl.html.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#---------------------------------------------------------------------------------------------------
# This is the CRUX 3.4 pre-install script for port contrib/ntp.
#---------------------------------------------------------------------------------------------------

getent group ntp || /usr/sbin/groupadd -g 77 ntp
getent passwd ntp || /usr/sbin/useradd -g ntp -u 77 -d /var/lib/ntp -s /bin/false ntp
/usr/bin/passwd -l ntp
```

7.6.12. prep-for-pkgadd

```
#!/bin/bash
# CCIA_0.0    PP-stick bin/prep-for-pkgadd .:....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#---------------------------------------------------------------------------------------------------
# WIP to replace code in bin/prtpkg_source
#---------------------------------------------------------------------------------------------------
set +xv -e
export IPV4=10.220.186.188
export CIDR=25
export NETDEV=enp1s0
export IPV4GW=10.220.186.129
export DOMAINDOT='home.'
export DNSHOST=192.168.1.1
export ISOPKGS='bind-utils'
#---------------------------------------------------------------------------------------------------
# If running the canonical ISO, the network is not configured nor started, but it is both if using
# a permanent root filesystem.  If the former, uncomment and EDIT AS NEEDED to configure and start
# the network (note that /etc/resolv.conf is overlaid by pkgadd activity so it is backed up in
# /etc/resolv.conf.wait for recovery after the pkgadd activity has been completed):
if   test "$PPTYPE" = 'CME'
then /bin/cp -p /etc/resolv.conf /etc/resolv.conf.wait
else # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     # For ethernet IPv4:
     /sbin/ip a add $IPV4/$CIDR dev $NETDEV
     /sbin/ip l set             dev $NETDEV up
     /sbin/ip r add default     via $IPV4GW
     # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     # For wireless IPv4:
```

```
     # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     # For ethernet IPv6:
     # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     # For wireless IPv6:
     # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     # Provision /etc/resolv.conf as /etc/resolv.conf.wait:
     /bin/cat <<EOD >/etc/resolv.conf.wait
domain $DOMAINDOT
search $DOMAINDOT
nameserver $DNSHOST
EOD
     /bin/cp -p /etc/resolv.conf.wait /etc/resolv.conf
fi
#----------------------------------------------------------------------------------------------------
# Ensure the network is properly configured and operational:
/sbin/ip a
/sbin/ip r
/bin/ping -c 3 $DNSHOST
/bin/ls -lh /etc/resolv.conf
/bin/cat    /etc/resolv.conf
#----------------------------------------------------------------------------------------------------
# Ensure the booted ISO's RAM filesystem is prepared for remaining pkgutils activity:
#
# Provision /etc/pkgmk.conf
#
test -f /etc/pkgmk.conf \
  || /bin/cat <<'EOD' >/etc/pkgmk.conf
#
# /etc/pkgmk.conf: pkgmk(8) configuration
#

export CFLAGS="-O2 -march=x86-64 -pipe"
export CXXFLAGS="${CFLAGS}"

# export MAKEFLAGS="-j2"

case ${PKGMK_ARCH} in
             "64"|"")
                          ;;
             "32")
                          export CFLAGS="${CFLAGS} -m32"
                          export CXXFLAGS="${CXXFLAGS} -m32"
                          export LDFLAGS="${LDFLAGS} -m32"
                          export PKG_CONFIG_LIBDIR="/usr/lib32/pkgconfig"
                          ;;
             *)
                          echo "Unknown architecture selected! Exiting."
                          exit 1
                          ;;
esac

# PKGMK_SOURCE_MIRRORS=()
# PKGMK_SOURCE_DIR="$PWD"
# PKGMK_PACKAGE_DIR="$PWD"
# PKGMK_WORK_DIR="$PWD/work"
# PKGMK_DOWNLOAD="no"
# PKGMK_IGNORE_FOOTPRINT="no"
# PKGMK_IGNORE_NEW="no"
# PKGMK_NO_STRIP="no"
# PKGMK_DOWNLOAD_PROG="wget"
# PKGMK_WGET_OPTS=""
# PKGMK_CURL_OPTS=""
# PKGMK_COMPRESSION_MODE="gz"

# End of file
EOD
#----------------------------------------------------------------------------------------------------
# Install into the booted ISO's RAM filesystem the needed ISO-3-3 package files not pre-installed
# within the ISO's $MEDIA/rootfs.tar.xz root filesystem file:
#
for i in $ISOPKGS
```

```
do  /usr/bin/grep -s "^$i$" /var/lib/pkg/db 2>/dev/null \
      && echo "Package $i was previously installed" \
      || { j=`echo $MEDIA/crux/*/${i}#*`
           echo "`/bin/date -u` -- /usr/bin/pkgadd -f $j started..."
           /usr/bin/pkgadd -f $j 1>/root/pkgadd.$i.log 2>&1 ; rc=$?
           echo "`/bin/date -u` -- /usr/bin/pkgadd -f $j ended with status $rc"
           get_stats 'pkga'
         }
done
# Restore the customized DNS resolver file:
/bin/mv /etc/resolv.conf.wait /etc/resolv.conf
```

7.6.13. prtpkg_mk_vars

This ***bash*** script is invoked from the ***prtpkg_prepare*** script for a platform that has booted either:

- the CRUX ISO using its canonical RAM disk root filesystem or
- a persistent CRUX Maintenance Environment (CME) that boots from the platform's primary SSD/HDD and uses a small root filesystem based upon the canonical CRUX ISO system's RAM disk root filesystem

to, when a /_/l/PP/h/$PREHOST/l/$PREROOT/PPvars file does not yet exist, assemble the parameters needed to set up the ISO/CME environment to perform the tasks for which the user booted the ISO/CME environment. Refer to Section **6** for more information about the processes using this script.

```
1. Determine the state of the environment:
   -- booted type (ISO | CME), version, label             PREBTYP, PREBVER, PREROOT
   -- BIOS | UEFI                                         PREBORU
   -- SSD/HDDs:                                           PREPDRV, PREDRVA..Z=[hs]da
      -- msdos | gpt                                      PREDRDA..Z={M|G}
      -- sector_phys_size                                 PREDRPA..Z=physical_size
      -- sector_logical_size                              PREDRSA..Z=logical_size
      -- bios_grub | no bios_grub                         PREDRGA..Z={part_num|0}
      -- esp | no esp (aka boot in parted)                PREDREA..Z={part_num|0}
      -- legacy_boot | no legacy_boot                     PREDRLA..Z={part_num|0}
      +> free_extents (sizes)                             PREDRXA..Z="[:xtent]*:"
      |  where xtent is
      |    first_sec,last_sec,num_secs,part_num|'',part_label|'',fstype|''
      |  in first_sec sequence
      |  pseudo-fstypes are PMBR, HMBR, PGH, SGH, PPEA, SPEA)
      +> partitions
         -- part_labels
         -- part_sizes
         -- File systems
            -- swaps
            -- CM(version, host_of_label)s
            -- CP(version, host_of_label)s
            -- non-CRUX-roots
   -- NIC(s)
      -- ifname
      -- addr(s), domain(s)
      +> ether
      +> wireless
         --  SSID
2. Determine reason(s) for preparing:
```

```
+> (Re)install                                           PPCMOP=I
|  -- BIOS | UEFI                                        PPTBOU
|  -- Repartition/format disk drives | leave as are     PPDRMA..Z
|     +> CRUX (nothing to do here)                       PPCRUX
|     +> Cell
|        -- Maintenance (CME) | non-maintenance (CWC)    PPTTYP
|           +> New commonwealth
|           |  +> Single-platform                        PPNCWT
|           |  +> Multi-platform
|           +> Existing commonwealth
|              +> Local /usr/prtpkg                      PPECWN
|              +> Remote /usr/prtpkg
+> Upgrade                                               PPCMOP=U
|  +> CRUX (nothing to do here)                          PPCRUX
|  +> Cell
|     -- Maintenance (CME) | non-maintenance (CWC)       PPTTYP
|        +> Local /usr/prtpkg                            PPECWN
|        +> Remote /usr/prtpkg
+> Rescue (?)                                            PPCMOP=R
```

```bash
#!/bin/bash
# CCIA_0.0    PP-stick bin/prtpkg_mk_vars ..:....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. .............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# See the CRUX 3.4 Commonwealth Manual for the CCI identified in the second line for the main
# documentation of this script.
#----------------------------------------------------------------------------------------------------
set +xv
# Validate environment variables expected from caller prtpkg_prepare:
if   test          ".$PREHOST"  = '.' \
          -o       ".$PREBOOT"  = '.' \
          -o       ".$PREROOT"  = '.' \
          -o \(    ".$PREBTYP" != '.ISO' \
                -a ".$PREBTYP" != '.CME' \
             \) \
          -o       ".$PREILAB"  = '.' \
          -o       ".$PREBVER"  = '.' \
          -o \(    ".$PREBORU" != '.BIOS' \
                -a ".$PREBORU" != '.UEFI' \
             \) \
          -o ! -s $MEDIA/crux-media
then echo "$0: PREBOOT<$PREBOOT> PREROOT<$PREROOT> PREBTYP<$PREBTYP> PREHOST<$PREHOST>"
     echo "$0: PREILAB<$PREILAB> PREBVER<$PREBVER> MEDIA<$MEDIA> PREBORU<$PREBORU>"
     echo "$0 was invoked without proper preparation--aborting!"
     exit 32
fi
if   test ! -f "/[${PREROOT}]"
then echo "$0 invoked with non-existent PREROOT label file (/[${PREROOT}])--aborting!"
     exit 32
fi
if   test ! -d "$PPHDIR"
then echo "$0 invoked with non-existent PPHDIR ($PPHDIR)--aborting!"
     exit 32
fi
if   test ! -d "$PPCDIR"
then echo "$0 invoked with non-existent PPCDIR ($PPCDIR)--aborting!"
     exit 32
fi
echo "In $0:" && env|sort
```

```
if   test ! -s /media/crux-media
then echo "$0 invoked with non-existent /media/crux-media file--aborting!"
     exit 32
fi
# Ask the user if BIOS or UEFI is desired and set PPTBOU accordingly:
if   test $PREBORU = 'UEFI'
then echo    'This platform booted using UEFI firmware.  If you know it supports legacy'
     echo    'BIOS in the BIOS setup and you want to normally use that configuration,'
     echo    'reply with a "B"; otherwise, enter 'U' to use the UEFI mode that requires'
     echo    'a functional EFI System Partition (ESP) exists or will be created on the'
     echo    'primary SSD/HDD.'
else echo    'This platform booted using BIOS firmware.  If you know it supports UEFI'
     echo    'in the BIOS setup and you want to normally use that configuration, reply'
     echo    'with a "U"; otherwise, enter "B" to use the BIOS mode that needs not a'
     echo    'functional EFI System Partition (ESP) on the primary SSD/HDD.'
fi
while test    ".$PPTBOU" != '.U' \
           -a ".$PPTBOU" != '.B'
do    read -p 'Enter "U" or "B":  ' PPTBOU
done
test $PPTBOU = 'B' \
  && export PPTBOU='BIOS' \
  || export PPTBOU='UEFI'
echo "PPTBOU<$PPTBOU>"
exit 1 #+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
# 3. Analyze environment and requests, assess sanity
#============================================================================================
export EDITOR=''                 # $EDITOR, only 'vi' and 'nano' are currently supported
export ISOLABEL=''               # ISO partition LABEL                                 dynamic
export MEDIA=''                  # path to ISO MEDIA                                   dynamic
export PPBORU=''                 # Bios OR Uefi; i.e., 'BIOS' or 'UEFI'
export PPPDRV=''                 # Primary DRiVe; e.g., 'hda', 'sdb'
export PPESPL=''                 # Efi System Partition Label; e.g., 'XA'             optional
export PPBOTL=''                 # BOoT fs Label; e.g., 'XB'                          optional
export PPMERL=''                 # ME Root fs Label; e.g., 'ME'                       optional
export PPMEMP=''                 # ME root fs Mount Point; e.g., "/_/l/$PPMERL"       optional
export PPTCRL=''                 # Target Crux Root fs Label; e.g., 'XD
export PPTCMP=''                 # Target Crux root fs Mount Point; e.g., "/_/l/$PPTCRL"
export PPTARD=''                 # TARget crux Drive; e.g., sda7                       dynamic
export MUSTBUILD='true'          # Set to '' if you want to install pre-built non-ISO
export NEEDNFS='true'            # If single-platform commonwealth, set to '' (false)
export NETDEV='enp?s0'           # Interface ID to network through
export IPV4='ddd.ddd.ddd.ddd'    # Interface's IPv4 address
export CIDR='dd'                 # Number of network bits in the IPV4 address
export IPV4GW='ddd.ddd.ddd.ddd'  # Default IPv4 gateway address
export DOMAINDOT='domainname.'   # /etc/resolv.conf domain/search
export DNSHOST='ddd.ddd.ddd.ddd' # /etc/resolv.conf nameserver
export NTPD='chrony'             # Choose chrony, ntpd, or ntpclient for NTP software
export NTPHOST='ddd.ddd.ddd.ddd' # IPv4 of the nearby NTP server for NTP protocol software
#xport NTPHOST='92.242.140.21'   # IPv4 of the nearby NTP server (time-b.nist.gov)
#xport NTPHOST='104.245.32.240'  # IPv4 of the nearby NTP server (cns1.atlantic.net)
#xport ADJFREQ=''                #   ntpclient -f value (null for unknown)
export RAMFSBOOT=315476          # Minimum root fs space needed to boot the RAM root fs ISO system;
export PRMFSBOOT=1289124         # Minimum root fs space needed to boot a permanent root filesystem
export RFSTOBUILD=288788         # Minimum free root filesystem space needed to install all
export RFSNONISO=100000          # Minimum free root filesystem space needed after installing build
export MEMMINRAM=368580          # Used memory after booting the canonical environment
export MEMMINPRM=172588          # Used memory after booting the permanent environment
export MEMTOBUILD=429544         # Minimum available memory needed to install all pre-built packages
export MEMNONISO=100000          # Minimum available memory after installing build tools needed to
exit 1 #+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
test    ".$EDITOR" != '.vi' \
     -a ".$EDITOR" != '.nano' \
  && { echo "The EDITOR environment variable is missing or unsupported--aborting!"
       exit 32
     }
test    $# -ne 1 \
     -o ".$1" = '.' \
  && { echo "Specify the source filename to upgrade as the only argument."
       exit 1
     }
```

```
case "$1" in # in collating sequence
  '.vimrc'                       ) tf='/root/.vimrc'                     ;;
  '.toprc'                       ) tf='/root/.toprc'                     ;;
  'PPvars'                       ) tf='/root/PPvars'                     ;;
  'etc.chrony.conf'              ) tf='/etc/chrony.conf'                 ;;
  'etc.fstab'                    ) tf='/etc/fstab'                       ;;
  'etc.hosts'                    ) tf='/etc/hosts'                       ;;
  'etc.ntpd.conf'                ) tf='/etc/ntpd.conf'                   ;;
  'etc.pkgadd.conf'              ) tf='/etc/pkgadd.conf'                 ;;
  'etc.pkgmk.conf'               ) tf='/etc/pkgmk.conf'                  ;;
  'etc.prt-get.conf'             ) tf='/etc/prt-get.conf'                ;;
  'etc.rc.conf'                  ) tf='/etc/rc.conf'                     ;;
  'etc.rc.d.net'                 ) tf='/etc/rc.d/net'                    ;;
  'etc.resolv.conf'              ) tf='/etc/resolv.conf'                 ;;
  'extlinF1.msg'                 ) tf='/boot/extlinF1.msg'               ;;
  'extlinF2.msg'                 ) tf='/boot/extlinF2.msg'               ;;
  'extlinux.conf'                ) tf='/boot/extlinux.conf'              ;;
  'prtpkg_altinstall'            ) tf='/root/prtpkg_altinstall'          ;;
  'prtpkg_altinstall.txt'        ) tf='/root/prtpkg_altinstall.txt'      ;;
  'prtpkg_chroot'                ) tf='/root/prtpkg_chroot'              ;;
  'prtpkg_getdrives.awk'         ) tf='/root/prtpkg_getdrives.awk'       ;;
  'prtpkg_locking.dash.source'   ) tf='/root/prtpkg_locking.dash.source' ;;
  'prtpkg_mkfs'                  ) tf='/root/prtpkg_mkfs'                ;;
  'prtpkg_mount'                 ) tf='/root/prtpkg_mount'               ;;
  'prtpkg_pdrive'                ) tf='/root/prtpkg_pdrive'              ;;
  'prtpkg_presetup'              ) tf='/root/prtpkg_presetup'            ;;
  'prtpkg_root.profile'          ) tf='/root/prtpkg_root.profile'        ;;
  'prtpkg_setup'                 ) tf='/root/prtpkg_setup'               ;;
  'prtpkg_shortcuts.bash.source' ) tf='/root/prtpkg_shortcuts.bash'      ;;
  'var.lib.pkg.prt-get.aliases'  ) tf='/var/lib/pkg/prt-get.aliases'     ;;
  *) echo "Specified source filename is unsupported--aborting!"
     exit 32
     ;;
esac
test -f "/_/l/PP/base/$1" \
  || { echo "Specified template file does not exist in /_/l/PP/base--aborting!"
       exit 32
     }
echo "Upon entering editfile, PPROTL<$PPROTL>"
if   test    ".$PPROTL" != '.'       \
          -a ".$PPROTL" != '.ISORAM'
then sd="/_/l/PP/h/$PPHOST/l/$PPROTL"
     if   test -d $sd
     then /bin/mkdir -p $sd
          echo "editfile created directory $sd"
     fi
     if   test -f "$sd/$1"
     then /bin/cp -p "$sd/$1" "$tf"
          echo "Editfile of $1 based upon $sd/$1"
     elif test -f "/_/l/PP/h/$PPHOST/$1"
     then /bin/cp -p "/_/l/PP/h/$PPHOST/$1" "$tf"
          echo "Editfile of $1 based upon /_/l/PP/h/$PPHOST/$1"
     else /bin/cp -p "/_/l/PP/base/$1"      "$tf"
          echo "Editfile of $1 based upon /_/l/PP/base/$1"
     fi
else sd="/_/l/PP/h/$PPHOST"
     if test -f "$sd/$1"
     then /bin/cp -p "/$sd/$1" "$tf"
          echo "Editfile of $1 based upon $sd/$1"
     else /bin/cp -p "/_/l/PP/base/$1"      "$tf"
          echo "Editfile of $1 based upon /_/l/PP/base/$1"
     fi
fi
$EDITOR "$tf"
/usr/bin/diff -C 1 "/_/l/PP/base/$1" "$tf" \
              > "$sd/$1.diffs" 2>&1
if   test $? -eq 0
then /bin/rm -f "$sd/$1.diffs"
     echo "editfile removed $sd/$1.diffs"
else /bin/cp -p "$tf" "$sd/$1"
     echo "editfile stowed $sd/$1"
```

```
fi
```

7.6.14. prtpkg_mkfsstatus

```
#!/bin/dash
# CCIA_0.0    PP-stick bin/prtpkg_mkfsstatus ....5....:....6....:....7....:....8....:....9....:....0
# ****************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ****************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# List /_/l/PP/h/$PPHOST/prtpkg_mkfs.$1.log after changing all backspaces into newlines and deleting
# all null lines.
#----------------------------------------------------------------------------------------------------
set +xv
if   test $# -ne 1
then echo "Syntax: $0 partition_label"
     exit 1
fi
/bin/cat /_/l/PP/h/$PPHOST/prtpkg_mkfs.$1.log \
| /usr/bin/tr '\010' '\012' \
| /bin/sed -e '/^$/d'
```

7.6.15. prtpkg_prepare

This *bash* script is invoked for a platform that has booted either:

- the CRUX ISO using its canonical RAM disk root filesystem or
- a persistent CRUX Maintenance Environment (CME) that boots from the platform's primary SSD/HDD and uses a small root filesystem based upon the canonical CRUX ISO system’s RAM disk root filesystem

to incorporate the Commonwealth Extension’s ISO/CME components into the root filesystem prior to installing or upgrading a CRUX CME or boot cell on the platform (or for modifying one or more SSD/HDD units of the platform prior to performing an install or upgrade of a canonical CRUX platform). Refer to Section **6** for more information about the processes using this script.

```
#!/bin/bash
# CCIA_0.0    PP-stick bin/prtpkg_prepare ..:....5....:....6....:....7....:....8....:....9....:....0
# ****************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ****************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# See the CRUX 3.4 Commonwealth Manual for the CCI identified in the second line for the main
# documentation of this script.
# Set flags may be used to aid debugging this script's execution:
# set -e: If a command's status is not zero, kill the shell
# set +e: Ignore status codes of all commands (normal bash default)
# set -v: Output shell source code to stderr
# set +v: Unset -v (normal shell default)
# set -x: Output substituted shell commands/assignments with leading '+ ' to stderr
# set +x: Unset -x (normal shell default)
#----------------------------------------------------------------------------------------------------
# This really should never be true, but if necessary, mount the PP-stick...
```

```
if   test ! -d /_/l/PP
then /bin/mkdir -p /_/l/PP
fi
set -e
if   test ! -f /_/l/PP/\[PP\]
then /bin/mount -L PP /_/l/PP -o rw,noatime
fi
# Set environment variables to be passed to prtpkg_mk_vars (if necessary) and prtpkg_chk_vars:
# PREBTYP boot system type -- either 'ISO' when the CRUX ISO has been booted or 'CME' when a CRUX
#         Maintenance Environment has been booted ('CWC' for CommonWealth Cell is not a valid
#         prtpkg_prepare boot system).
# PREROOT partition label of the root file system (if ISO, 'ISORAM' is used).
# PREBOOT hostname of booted system, either 'crux' for an ISO boot or not 'crux' for a CME boot.
# PREHOST hostname to use for the platform as extracted from the boot environment or the user.
export PREBOOT=`/bin/hostname` # Canonical host is 'crux'--CME hosts must never be
# Obtain the root filesys label file if any:
if   test `echo /\[*\] | /usr/bin/wc -w` -gt 1
then echo "Found multiple label files in the root file system -- aborting!"
     exit 32
else export PREROOT=`echo /\[*\] | /bin/sed -e 's/^\/\[//;s/\]$//' | /usr/bin/tr -d '\012'`
fi
echo "PREBOOT<$PREBOOT> PREROOT<$PREROOT>"
if   test ".$PREROOT" != '.*'
then # There is a label file--read the hostname and any PREBTYP it contains:
     # If there is a [ISORAM] label file for host 'crux', it contains not 'crux'
     # but the previously solicited platform hostname
     if   test ! -s "/[$PREROOT]"
     then echo "The root file system label file is empty -- aborting!"
          exit 32
     fi
     /bin/cat /\[$PREROOT\] | read PREHOST PREBTYP
     if   test ".$PREHOST" = '.'
     then echo "The [ISORAM] file contains no hostname -- aborting!"
          exit 32
     elif test ".$PREHOST" = '.crux'
     then echo "The [ISORAM] file contains invalid host name 'crux' -- aborting!"
          exit 32
     fi
     export PREHOST
     if   test    ".$PREBTYP" != '.' \
               -a ".$PREBTYP" != '.ISO' \
               -a ".$PREBTYP" != '.CME' \
               -a ".$PREBTYP" != '.CWC'
     then echo "The [ISORAM] file contains invalid PREBTYP ($PREBTYP) -- aborting!"
          exit 32
     else echo "PREBTYP set to <$PREBTYP> using /[ISORAM] file contents"
     fi
     export PREBTYP
     if   test ".$PREBOOT" = '.crux'
     then echo "PREHOST set to <$PREHOST> using /[ISORAM] file contents"
     elif test ".$PREBOOT" != ".$PREHOST"
     then echo "The hostname command returned <$PREBOOT> but the $PREROOT file"
          echo "contains an inappropriate hostname <$PREHOST> -- aborting!"
          exit 32
     else echo "PREHOST set to <$PREHOST> using hostname (which matches $PREROOT label file)."
     fi
     if   test    "$PREHOST" = 'crux' \
               -a "$PREROOT" != 'ISORAM'
     then echo "The $PREROOT label file contains inappropriate hostname $PREHOST -- aborting!"
          exit 32
     elif test    \(     "$PREROOT" = 'crux' \
                     -a "$PREBTYP" != 'ISO' \
                  \) \
               -o       "$PREBTYP" != 'CME'
     then echo "The $PREROOT label file contains inappropriate prepare PREBTYP $PREBTYP -- aborting!"
          exit 32
     fi
else # There is no label file--this had better be hostname crux with no l+f file in a
     # tmpfs, in which case, we solicit the host name and create the label file with it
     if   test    ".$PREBOOT" = '.crux' \
               -a ! -f /lost\+found \
               -a '.tmpfs' = .`/usr/bin/gawk '$2 == "/" {print $3}' /proc/mounts`
     then while test     ".$PREHOST" = '.' \
                     -o ".$PREHOST" = '.crux'
          do   read -p 'Enter the hostname this platform is to use:  ' PREHOST
               test     ".$PREHOST" = '.' \
                    -o ".$PREHOST" = '.crux' \
                 && echo 'The response is an unacceptable host name'
          done
          export PREHOST
          export PREBTYP='ISO'
          echo "$PREHOST   $PREBTYP" > /\[ISORAM\]
          export PREROOT='ISORAM'
     else # This is permanent--where did its label file go?
          echo "Non-canonical root file system has no label file -- aborting!"
          exit 32
     fi
fi
echo "PREBOOT<$PREBOOT> PREROOT<$PREROOT> PREBTYP<$PREBTYP> PREHOST<$PREHOST>"
# If running a permanent root filesystem, mount the ISO (if necessary) and set the MEDIA variable
# to /_/l/CRUX-$PPBVER; else set MEDIA to /media
export PREILAB=`cd /dev/disk/by-label && echo CRUX-* | /bin/sed -e 's/CRUX-EFI//;s/ //g'`
export PREBVER=`echo $PREILAB | /bin/sed -e 's/^CRUX-//;s/-updated$//'`
if   test $PREBTYP = 'ISO'
then if   test ! -s /media/crux-media
     then echo "$0 invoked with non-existent /media/crux-media file--aborting!"
          exit 32
     fi
     export MEDIA=/media
else /bin/mkdir -p /_/l/CRUX-$PREBVER
     if   test ! -f /_/l/CRUX-$PREBVER/crux-media
     then /bin/mount -L $ISOLAB /_/l/CRUX-$PREBVER -o ro
     fi
     export MEDIA=/_/l/CRUX-$PREBVER
fi
# Set PREBORU based upon booted mode:
test -f /sys/firmware/efi/efivars \
  && export PREBORU='UEFI' \
  || export PREBORU='BIOS'
echo "PREILAB<$PREILAB> PREBVER<$PREBVER> MEDIA<$MEDIA> PREBORU<$PREBORU>"
echo "The ISO's crux-media file contains <`/bin/cat $MEDIA/crux-media`>"
# If necessary establish this platform's subdirectory in the PP-stick:
export PPHDIR=/_/l/PP/h/$PREHOST # Directory in PP-stick for customized files and logs for this host
if   test ! -d $PPHDIR
then /bin/mkdir -p $PPHDIR # Ensure there is a directory in the PP-stick for this host
fi
# If necessary, establish this cell's subdirectory in the PP-stick using the bare root label:
export PPCDIR=$PPHDIR/l/$PREROOT # Directory in PP-stick for customized files and logs for this cell
if   test ! -d $PPCDIR
then /bin/mkdir -p $PPCDIR
fi
# If there is a $PPCDIR/PPargs file (from previous prtpkg_prepare processing), source it to export
# its variables; otherwise, deduce the variables, asking the user when that is not possible, and
# build the PPargs file in the /root directory (once all variables are defined and determined to
# be sane, /root/PPargs will be moved to $PPCDIR).
if   test \(    -f $PPCDIR/PPargs \
             -o -L $PPCDIR/PPargs \
          \)
then .              $PPCDIR/PPargs
     echo "Sourced $PPCDIR/PPargs"
else env | /usr/bin/sort # show vars so far
     /_/l/PP/bin/prtpkg_mk_vars
fi
env | /usr/bin/sort
/_/l/PP/bin/prtpkg_chk_vars # Ensure all variables are sane before continuing
exit 1 #+++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++++
# If there is a $PPCDIR/prtpkg_shortcuts.bash.source file or symlink, source it; else
# if there is a $PPHDIR/prtpkg_shortcuts.bash.source file or symlink, source it; else
# if there is a /_/l/PP/base/prtpkg_shortcuts.bash.source file or symlink, source it.
if   test \(    -f $PPCDIR/prtpkg_shortcuts.bash.source \
             -o -L $PPCDIR/prtpkg_shortcuts.bash.source \
          \)
```

```
then .             $PPCDIR/prtpkg_shortcuts.bash.source
     echo "Sourced $PPCDIR/prtpkg_shortcuts.bash.source"
elif test \(    -f $PPHDIR/prtpkg_shortcuts.bash.source \
             -o -L $PPHDIR/prtpkg_shortcuts.bash.source \
          \)
then .             $PPHDIR/prtpkg_shortcuts.bash.source
     echo "Sourced $PPHDIR/prtpkg_shortcuts.bash.source"
elif test \(    -f /_/l/PP/base/prtpkg_shortcuts.bash.source \
             -o -L /_/l/PP/base/prtpkg_shortcuts.bash.source \
          \)
then .             /_/l/PP/base/prtpkg_shortcuts.bash.source
     echo "Sourced /_/l/PP/base/prtpkg_shortcuts.bash.source"
fi
# Determine the preferred editor, asking if necessary, and establish the EDITOR
# environment variable
while case $EDITOR in
        'n' | 'nano' ) export EDITOR='nano' ; false ;;
        'v' | 'vi'   ) export EDITOR='vi'   ; false ;;
        *            ) export EDITOR=''     ; true  ;;
      esac
do    read -p "Select your editor ('v' or 'vi' for vi, or 'n' or 'nano' for nano):  " EDITOR
done
set +e
if   test PPBTYP = 'CME'
     # Since this is a permanent CME root filesys, get its label and check/set PPMERL
     if   test ".$PPMERL" != '.'
     then if   test ! -f /\[$PPMERL\]
          then echo "In the CME system, the [$PPMERL] label file was not found--aborting!" \
               | /usr/bin/tee -a /root/prtpkg.fail
               /bin/sleep 3s
               exit 32 # this kills the session
          fi
          if   test ! -d $PPHDIR/l/$PPMERL
          then echo "No $PPMERL subdirectory found in $PPHDIR!" \
               | /usr/bin/tee -a /root/prtpkg.fail
               /bin/sleep 3s
               exit 32 # this kills the session
          fi
     else echo "In the CME system, the PPMERL environment variable is undefined--aborting!" \
          | /usr/bin/tee -a /root/prtpkg.fail
          /bin/sleep 3s
          exit 32 # this kills the session
     fi
fi
# Identify which ISO is afoot and set the ISOLABEL variable; else kill the session:
if   test -L /dev/disk/by-label/CRUX-3.4
then export ISOLABEL='CRUX-3.4'
elif test -L /dev/disk/by-label/CRUX-3.4-updated
then export ISOLABEL='CRUX-3.4-updated'
else echo 'Could not find the label of the CRUX ISO--is the USB-stick plugged in?' \
     | /usr/bin/tee -a /root/prtpkg.fail
     echo 'Aborting the session!' \
     | /usr/bin/tee -a /root/prtpkg.fail
     /bin/sleep 3s
     exit 32 # this kills the session
fi
# If running the permanent root filesystem, mount the ISO and set the MEDIA variable to
# /_/l/CRUX-3.4; else set MEDIA to /media
if   test $PPBTYP = 'CME'
then /bin/mkdir -p /_/l/CRUX-3.4
     if   test ! -f /_/l/CRUX-3.4/crux-media
     then /bin/mount -L $ISOLABEL /_/l/CRUX-3.4 -o ro
     fi
     export MEDIA=/_/l/CRUX-3.4
else export MEDIA=/media
fi
echo 'The ISO root directory contains these inodes:'
/bin/ls -lhd $MEDIA/*
# Prepare to run pkgadd in this basic environment:
if   test ! -d /var/lib/pkg
then /bin/mkdir -p /var/lib/pkg
```

```
fi
if   test ! -f /var/lib/pkg/db
then : >/var/lib/pkg/db
fi
test -d /root/pkgadd \
  || mkdir /root/pkgadd
# If the /root directory has not been populated yet by prtpkg_source, do that now:
if   test ! -f /root/prtpkg_presetup
then set +e
     # If necessary, temporarily provision /etc/pkgadd.conf (it will be deleted,
     # then formally added via the editfile loop).:
     if   test -f /etc/pkgadd.conf
     then rm_pkgadd_conf=N
     else rm_pkgadd_conf=Y
          /bin/cat <<'EOD' >/etc/pkgadd.conf
#
# /etc/pkgadd.conf: pkgadd(8) configuration
#

# Default rule (implicit)
#UPGRADE     ^.*$                                 YES

UPGRADE                 ^etc/.*$                  NO
UPGRADE                 ^var/log/.*$              NO
UPGRADE                 ^var/spool/\w*cron/.*$    NO
UPGRADE                 ^var/run/utmp$                          NO

UPGRADE                 ^etc/ports/drivers/.*$    YES
UPGRADE                 ^etc/X11/.*$              YES

UPGRADE                 ^etc/rc.*$                YES
UPGRADE                 ^etc/rc\.local$                         NO
UPGRADE                 ^etc/rc\.modules$         NO
UPGRADE                 ^etc/rc\.conf$                          NO
UPGRADE                 ^etc/rc\.d/net$                         NO

UPGRADE                 ^etc/udev/rules.d/.*$     YES
UPGRADE                 ^etc/udev/rules.d/1.*$    NO
UPGRADE                 ^etc/udev/hwdb.d/.*$      YES
UPGRADE                 ^etc/udev/hwdb.bin$       YES

UPGRADE                 ^etc/ssl/cert.pem$        YES
# End of file
EOD
     fi
     # Install cmp and diff (very useful for rescue work):
     if   test ! -x /usr/bin/diff
     then cmd="/usr/bin/pkgadd -f $MEDIA/crux/core/diffutils#3.6-1.pkg.tar.xz"
          echo "Command: $cmd" > /root/pkgadd/diffutils.log 2>&1
          $cmd > /root/pkgadd/diffutils.log 2>&1
          echo "pkgadd for diffutils returned status $?"
     fi
     test rm_pkgadd_conf = Y \
       && /bin/rm /etc/pkgadd.conf
     # Set up the prtpkg scripts in /root, letting the user edit them straight-forwardly:
     for ii in .vimrc \
               PPvars \
               prtpkg_root.profile \
               prtpkg_shortcuts.bash.source \
               prtpkg_pdrive \
               prtpkg_mkfs \
               prtpkg_mount \
               prtpkg_altinstall.txt \
               prtpkg_altinstall \
               prtpkg_presetup \
               prtpkg_setup \
               prtpkg_chroot \
               etc.rc.conf \
               etc.fstab \
               etc.hosts \
               etc.chrony.conf \
               etc.ntpd.conf \
               etc.rc.d.net \
               etc.resolv.conf \
               etc.pkgadd.conf \
               etc.pkgmk.conf \
               etc.prt-get.conf \
               var.lib.pkg.prt-get.aliases \
               extlinux.conf \
               extlinF1.msg \
               extlinF2.msg \
               prtpkg_getdrives.awk \
               prtpkg_locking.dash.source \
               .toprc
     do  /_/l/PP/bin/editfile $ii
         if   test $ii = 'PPvars'
         then /_/l/PP/bin/prtpkg_checkvars
              if   test $? -ne 0
              then echo "The PPvars file is unacceptable--aborting!" \
                   | /usr/bin/tee -a /root/prtpkg.fail
                   /bin/sleep 3s
                   exit 32 # this kills the session
              fi
         fi
     done
     . /root/prtpkg_root.profile # Update PS1 to know which environment we're in
                                 # Note ISORAM has no .profile initially, but
                                 # prtpkg_pdrive will build one from this file plus
                                 # prtpkg_shortcuts.bash.source
     /bin/cp -p /_/l/PP/bin/prtpkg_mkfsstatus    /root/mkfsstat
fi
# Invoke prtpkg_pdrive without the build argument to ascertain the drive(s) situation:
/root/prtpkg_pdrive 2>&1 \
| /usr/bin/tee /root/prtpkg_pdrive.info.log
if   test -f /root/PPargs
then . /root/PPargs
else echo 'prtpkg_source found no /root/PPargs file returned by prtpkg_pdrive!'
fi
/bin/cat <<EOD >>/root/PPargs # Extend the /root/PPargs file
export PPBORU='$PPBORU'
export MEDIA='$MEDIA'
export ISOLABEL='$ISOLABEL'
export PPBTYP='$PPBTYP'
export PPHOST='$PPHOST'
export EDITOR='$EDITOR'
EOD
# /usr/bin/sort /root/PPargs | /usr/bin/uniq >/root/PPargs.new
# test $? -eq 0 && /bin/mv /root/PPargs{.new,}
/bin/cp -p /root/PPargs $PPCDIR
/bin/df -alHT | /usr/bin/grep ^/dev/[sh]d
/_/l/PP/bin/mntdrvs
echo "prtpkg_source has finished preparing for the invocation of the /root/prtpkg_presetup"
echo "script.  If the primary SSD/HDD needs to be (re)partitioned and (re)formatted, run the"
echo "'/root/prtpkg_pdrive build' script"
echo "before    (if you decide to create a persistent maintenance environment (CME) root"
echo "           filesystem tree--be sure to follow the procedure documented in the"
echo "           persistent_filesystem.txt file before running /root/prtpkg_presetup!)"
echo "or after  (if you decide otherwise)"
echo "you run the /root/prtpkg_presetup script.  If the primary SSD/HDD is ready for CRUX"
echo "maintenance, just run /root/prtpkg_presetup to prepare the environment to run the"
echo "/root/prtpkg_setup script on the production CRUX system you intend to maintain--you"
echo "do NOT want to invoke prtpkg_pdrive build processing in this situation."
if   test $PPBTYP = 'ISO'
then echo ""
     echo "If you have not yet decided to establish an CME root filesystem as described in the"
     echo "PP-stick's persistent_filesystem.txt file, you can now run /root/prtpkg_presetup."
     echo "It will simply stop if the RAMdrive is too small for complete package building"
     echo "capability to be installed and tell you so.  If it does stop (the platform likely"
     echo "has less than 2 GB of RAM) and you don't need the full build capability, you can"
     echo "then change the MUSTBUILD variable in the prtpkg_presetup script to false (null)"
     echo "before rerunning it, and it will only install needed packages not in the ISO from"
     echo "the prebuilt packages you have put in the PP-stick."
```

```
fi
```

7.6.16. prtpkg_source

This file is to be sourced by a ***bash*** or ***dash*** root shell for a platform that has booted either:

- the CRUX ISO using its canonical RAM disk root filesystem or
- a persistent CRUX Maintenance Environment (CME) that boots from the platform's primary SSD/HDD and uses a small root filesystem based upon the canonical CRUX ISO system's RAM disk root filesystem

to incorporate the Commonwealth Extension ISO/CME components into the root filesystem prior to installing or upgrading a CRUX CME or boot cell on the platform. Refer to Section **6** for more information about the processes using this script.

The file is a ***bash*** script that is run using the shell's ***source*** (alias ***.*** aka dot) command. The primary difference is the commands in a source file are executed in the same shell that accepted the ***source*** command (as though they were typed in), not in a spawned sub-shell. This enables changes to the shell's set of environment variables to persist after the ***source*** command finishes its processing.

These files are incorporated into the maintenance environment by running these commands after the root shell prompt is initially proffered:

```
/bin/mkdir -p /_/l/PP # Defaults to attributes root:root 644
/bin/mount -L PP /_/l/PP -o rw,noatime
. /_/l/PP/bin/prtpkg_source
```

This file's processing sets up some useful command aliases (by sourcing the relevant likely customized prtpkg_shortcuts.bash.source file), evaluates the environment, administers editing of the needed prtpkg components into the /root directory, and validates and/or sets many global variables used by prtpkg ISO/CME automata.

As a consequence of being able to preserve state in the PP-stick from prior invocations of this source file, the processing will change between the first time it is run for a particular platform (as identified by a host name) and any subsequent reruns. The major rationale is to enable any customization of the prtpkg scripts for one or more platforms to be "remembered" and thus performed only once. Then, rebuilding such a platform from scratch becomes a highly automated process. Also, defining a clone of such a platform becomes very straight-forward and often quite trivial. This can be performed beforehand either by, after mounting the PP-stick in the ISO environment, creating the modified and/or cloned files before the prtpkg_source file is invoked, or even before the ISO image is booted by mounting the PP-stick on a different platform and preparing the PP-stick for the new platform, then unmounting it, ejecting it, unloading the media, and taking it to the target platform to begin the installation process.

For each needed `prtpkg` component, the `prtpkg_source` processing runs the `/_/l/PP/bin/editfile` script which finds the best candidate, copies that to its target location in `/root`, invokes the interactive editor of choice upon the target file, and, if the target file is now different from the default version, the target file is copied into the appropriate location (see the next paragraph), along with the output of a ***`diff`*** of the two.

Platform-versioned (as well as platform-root-filesystem-versioned) scripts and other files are maintained in a subdirectory tree of the `PP-stick` named using the platform's host name in accordance with the `CCIA dir-type` conventions (see `README.dir-types` for details, which is embedded in Section **7.5.1.2**). The host name is obtained from the sysadmin's prompted keystrokes very early in the `prtpkg_source` file's invocation as a last resort if it cannot be found in the `PP-stick` or the main drive of the booted system. A host's tree will include subdirectories for each root filesystem of the host, each named using the filesystem's label as discussed in Section **7.5.1.3** which contains the `README.filesystems` file.

Note the `prtpkg_source` file should never be versioned for a particular host, let alone a particular root filesystem. If something like that is needful for groups of platforms, it is necessary to maintain commonwealth- or group- versioned `PP-sticks` for use with the platforms within those groups by forking a new `CCI`. It follows that all platforms using the same `PP-stick` are also using the same `CCI`.

This file's processing sets up some useful command aliases, copies `prtpkg` components into the `/root` directory, evaluates the environment and sets many global variables for use by `prtpkg ISO/CME` automata. As a consequence of being able to preserve state on the `PP-stick` from prior invocations of this source file, the processing will change between the first time it is run for a particular platform (as identified by a host name) and any subsequent reruns. The major rationale is to enable customization of the `prtpkg` scripts for each platform to be "remembered" and thus performed only once. Then, rebuilding that platform from scratch becomes a highly automated process.

Platform-versioned scripts and other files are maintained in a subdirectory tree of the `PP-stick` named using the platform's host name. The host name is obtained from the sysadmin's prompted keystrokes very early in this source file's invocation as a last resort if it cannot be found in one of several places in the `PP-stick` or the main drive of the booted system. A host's tree will include subdirectories for each root filesystem of the host, each named using the filesystem's label.

Note the `prtpkg_source` file should never be versioned for platforms. If something like that is needful, `CCIA` needs to be forked.

When running in the `ISO` environment, there is never a preexisting `/root/PPvars` file (only used briefly while establishing a host previously unencountered when using the participating `PP-stick`).

```
#!/bin/bash
# CCIA_0.0    PP-stick bin/prtpkg_source ...:....5....:....6....:....7....:....8....:....9....:....0
# *************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# *************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#-------------------------------------------------------------------------------------------------
# See the CRUX 3.4 Commonwealth Manual for the CCI identified in the second line for the main
# documentation of this script.
# Set flags may be used to aid debugging this source file's execution:
# set -e: If a command's status is not zero, kill the shell
# set +e: Ignore status codes of all commands (normal bash default)
# set -v: Output shell source code to stderr
# set +v: Unset -v (normal shell default)
# set -x: Output substituted shell commands/assignments with leading '+ ' to stderr
# set +x: Unset -x (normal shell default)
#-------------------------------------------------------------------------------------------------
# This really should never be true, but if necessary, mount the PP-stick...
if   test ! -d /_/l/PP
then /bin/mkdir -p /_/l/PP
fi
set -e
if   test ! -f /_/l/PP/\[PP\]
then /bin/mount -L PP /_/l/PP -o rw,noatime
fi
BOOTHOST=`/bin/hostname` # Canonical host is 'crux'--CME hosts must never be
# Obtain the root filesys label file if any:
if   test `echo /\[*\] | /usr/bin/wc -w` -gt 1
then echo "Found multiple label files in the root file system -- aborting!" \
     | /usr/bin/tee -a /root/prtpkg.fail
     /bin/sleep 3s
     exit 32 # this kills the session
else PPROTL=`echo /\[*\] | /bin/sed -e 's/^\/\[//;s/\]$//' | /usr/bin/tr -d '\012'`
fi
export PPROTL
echo "BOOTHOST<$BOOTHOST> PPROTL<$PPROTL>"
if   test ".$PPROTL" != '.*'
then # There is a label file--read the host name it contains:
     # If there is a [ISORAM] label file for host 'crux', it contains not 'crux'
     # but the previously solicited platform hostname
     if   test ! -s "/[$PPROTL]"
     then echo "The root file system label file is empty -- aborting!" \
          | /usr/bin/tee -a /root/prtpkg.fail
          /bin/sleep 3s
          exit 32 # this kills the session
     fi
     PPHOST=`cat /\[$PPROTL\] | /usr/bin/tr -d '\012'`
     if   test $PPHOST = 'crux'
     then echo "The [ISORAM] file contains invalid host name 'crux' -- aborting!" \
          | /usr/bin/tee -a /root/prtpkg.fail
          /bin/sleep 3s
          exit 32 # this kills the session
     fi
     if   $BOOTHOST = 'crux'
     then echo "PPHOST set to <$PPHOST> using /[ISORAM] file contents"
     elif test ".$HOSTNAME" != ".$PPHOST"
```

```
          then echo "The hostname command returned <$BOOTHOST> but the $PPROTL file" \
               | /usr/bin/tee -a /root/prtpkg.fail
               echo "contains an inappropriate hostname <$PPHOST> -- aborting!" \
               | /usr/bin/tee -a /root/prtpkg.fail
               /bin/sleep 3s
               exit 32 # this kills the session
          else echo "PPHOST set to <$PPHOST> using hostname (which matches $PPROTL label file)."
          fi
          if   test    "$PPHOST" = 'crux' \
                    -a "$PPROTL" != 'ISORAM'
          then echo "The $PPROTL label file contains inappropriate hostname $PPHOST -- aborting!" \
               | /usr/bin/tee -a /root/prtpkg.fail
               /bin/sleep 3s
               exit 32 # this kills the session
          elif test "$PPHOST" = 'crux'
          then export PPTYPE='ISO'
          else export PPTYPE='CME'
          fi
     else # There is no label file--this had better be hostname crux with no l+f file in a
          # tmpfs, in which case, we solicit the host name and create the label file with it
          if   test    ".$HOSTNAME" = '.crux' \
                    -a ! -f /lost\+found \
                    -a '.tmpfs' = .`/usr/bin/gawk '$2 == "/" {print $3}' /proc/mounts`
          then while test    ".$PPHOST" = '.' \
                          -o ".$PPHOST" = '.crux'
               do   read -p 'Enter the hostname this platform is to use:  ' PPHOST
                    test    ".$PPHOST" = '.' \
                         -o ".$PPHOST" = '.crux' \
                      && echo 'The response is an unacceptable host name'
               done
               export PPHOST
               export PPTYPE='ISO'
               echo "$PPHOST" > /\[ISORAM\]
               export PPROTL='ISORAM'
          else # This is permanent--where did its label file go?
               echo "Non-canonical root file system has no label file -- aborting!" \
               | /usr/bin/tee -a /root/prtpkg.fail
               /bin/sleep 3s
               exit 32 # this kills the session
          fi
     fi
     echo "BOOTHOST<$BOOTHOST> PPROTL<$PPROTL> PPTYPE<$PPTYPE> PPHOST<$PPHOST>"
     # If necessary establish this platform's subdirectory in the PP-stick:
     PPHDIR=/_/l/PP/h/$PPHOST # Directory in PP-stick for customized files and logs for this host
     if   test ! -d $PPHDIR
     then /bin/mkdir -p $PPHDIR # Ensure there is a directory in the PP-stick for this host
     fi
     # If necessary, establish this cell's subdirectory in the PP-stick using the bare root label:
     PPCDIR=/$PPHDIR/$PPROTL # Directory in PP-stick for customized files and logs for this cell
     if   test ! -d $PPCDIR
     then /bin/mkdir -p $PPCDIR
     fi
     # If there is a $PPCDIR/PPvars file or symlink, source it; else
     # if there is a $PPHDIR/PPvars file or symlink, source it; else
     # if there is a /_/l/PP/PPvars file or symlink (very temporary), source it.
     if   test \(    -f $PPCDIR/PPvars \
                  -o -L $PPCDIR/PPvars \
               \)
     then .             $PPCDIR/PPvars
          echo "Sourced $PPCDIR/PPvars"
     elif test \(    -f $PPHDIR/PPvars \
                  -o -L $PPHDIR/PPvars \
               \)
     then .             $PPHDIR/PPvars
          echo "Sourced $PPHDIR/PPvars"
     elif test \(    -f /_/l/PP/PPvars \
                  -o -L /_/l/PP/PPvars \
               \)
     then .             /_/l/PP/PPvars
          echo "Sourced /_/l/PP/PPvars"
     fi
```

```
# If there is a $PPCDIR/prtpkg_shortcuts.bash.source file or symlink, source it; else
# if there is a $PPHDIR/prtpkg_shortcuts.bash.source file or symlink, source it; else
# if there is a /_/l/PP/base/prtpkg_shortcuts.bash.source file or symlink, source it.
if   test \(    -f $PPCDIR/prtpkg_shortcuts.bash.source \
             -o -L $PPCDIR/prtpkg_shortcuts.bash.source \
          \)
then .             $PPCDIR/prtpkg_shortcuts.bash.source
     echo "Sourced $PPCDIR/prtpkg_shortcuts.bash.source"
elif test \(    -f $PPHDIR/prtpkg_shortcuts.bash.source \
             -o -L $PPHDIR/prtpkg_shortcuts.bash.source \
          \)
then .             $PPHDIR/prtpkg_shortcuts.bash.source
     echo "Sourced $PPHDIR/prtpkg_shortcuts.bash.source"
elif test \(    -f /_/l/PP/base/prtpkg_shortcuts.bash.source \
             -o -L /_/l/PP/base/prtpkg_shortcuts.bash.source \
          \)
then .             /_/l/PP/base/prtpkg_shortcuts.bash.source
     echo "Sourced /_/l/PP/base/prtpkg_shortcuts.bash.source"
fi
# Determine the preferred editor, asking if necessary, and establish the EDITOR
# environment variable
while test    ".$EDITOR" != '.n' \
           -a ".$EDITOR" != '.v'
do    read -p "Select your editor ('v' for vi or 'n' for nano):  " EDITOR
      test    ".$EDITOR" != '.n' \
           -a ".$EDITOR" != '.v' \
        && echo 'The response must be the character v or n'
done
if   test $EDITOR = 'n'
then export EDITOR='nano'
else export EDITOR='vi'
fi
set +e
# Determine type of environment and define the PPTYPE variable:
if   test    ! -f /lost\+found \
          -a '.tmpfs' = .`/usr/bin/gawk '$2 == "/" {print $3}' /proc/mounts`
then export PPTYPE='ISO'
else export PPTYPE='CME'
     # Since this is a permanent CME root filesys, get its label and check/set PPRDIR
     if   test ".$PPMERL" != '.'
     then if   test ! -f /\[$PPMERL\]
          then echo "In the CME system, the [$PPMERL] label file was not found--aborting!" \
               | /usr/bin/tee -a /root/prtpkg.fail
               /bin/sleep 3s
               exit 32 # this kills the session
          fi
          if   test ! -d $PPHDIR/l/$PPMERL
          then echo "No $PPMERL subdirectory found in $PPHDIR!" \
               | /usr/bin/tee -a /root/prtpkg.fail
               /bin/sleep 3s
               exit 32 # this kills the session
          fi
     else echo "In the CME system, the PPMERL environment variable is undefined--aborting!" \
          | /usr/bin/tee -a /root/prtpkg.fail
          /bin/sleep 3s
          exit 32 # this kills the session
     fi
fi
# Identify which ISO is afoot and set the ISOLABEL variable; else kill the session:
if   test -L /dev/disk/by-label/CRUX-3.4
then export ISOLABEL='CRUX-3.4'
elif test -L /dev/disk/by-label/CRUX-3.4-updated
then export ISOLABEL='CRUX-3.4-updated'
else echo 'Could not find the label of the CRUX ISO--is the USB-stick plugged in?' \
     | /usr/bin/tee -a /root/prtpkg.fail
     echo 'Aborting the session!' \
     | /usr/bin/tee -a /root/prtpkg.fail
     /bin/sleep 3s
     exit 32 # this kills the session
fi
# If running the permanent root filesystem, mount the ISO and set the MEDIA variable to
# /_/l/CRUX-3.4; else set MEDIA to /media
if   test $PPTYPE = 'CME'
then /bin/mkdir -p /_/l/CRUX-3.4
     if   test ! -f /_/l/CRUX-3.4/crux-media
     then /bin/mount -L $ISOLABEL /_/l/CRUX-3.4 -o ro
     fi
     export MEDIA=/_/l/CRUX-3.4
else export MEDIA=/media
fi
echo 'The ISO root directory contains these inodes:'
/bin/ls -lhd $MEDIA/*
# Prepare to run pkgadd in this basic environment:
if   test ! -d /var/lib/pkg
then /bin/mkdir -p /var/lib/pkg
fi
if   test ! -f /var/lib/pkg/db
then : >/var/lib/pkg/db
fi
test -d /root/pkgadd \
  || mkdir /root/pkgadd
# If the /root directory has not been populated yet by prtpkg_source, do that now:
if   test ! -f /root/prtpkg_presetup
then set +e
     # If necessary, temporarily provision /etc/pkgadd.conf (it will be deleted,
     # then formally added via the editfile loop).:
     if   test -f /etc/pkgadd.conf
     then rm_pkgadd_conf=N
     else rm_pkgadd_conf=Y
          /bin/cat <<'EOD' >/etc/pkgadd.conf
#
# /etc/pkgadd.conf: pkgadd(8) configuration
#

# Default rule (implicit)
#UPGRADE      ^.*$                                  YES

UPGRADE                  ^etc/.*$                   NO
UPGRADE                  ^var/log/.*$               NO
UPGRADE                  ^var/spool/\w*cron/.*$     NO
UPGRADE                  ^var/run/utmp$                          NO

UPGRADE                  ^etc/ports/drivers/.*$     YES
UPGRADE                  ^etc/X11/.*$               YES

UPGRADE                  ^etc/rc.*$                 YES
UPGRADE                  ^etc/rc\.local$                         NO
UPGRADE                  ^etc/rc\.modules$          NO
UPGRADE                  ^etc/rc\.conf$                          NO
UPGRADE                  ^etc/rc\.d/net$                         NO

UPGRADE                  ^etc/udev/rules.d/.*$      YES
UPGRADE                  ^etc/udev/rules.d/1.*$     NO
UPGRADE                  ^etc/udev/hwdb.d/.*$       YES
UPGRADE                  ^etc/udev/hwdb.bin$        YES

UPGRADE                  ^etc/ssl/cert.pem$         YES
# End of file
EOD
     fi
     # Install cmp and diff (very useful for rescue work):
     if   test ! -x /usr/bin/diff
     then cmd="/usr/bin/pkgadd -f $MEDIA/crux/core/diffutils#3.6-1.pkg.tar.xz"
          echo "Command: $cmd" > /root/pkgadd/diffutils.log 2>&1
          $cmd > /root/pkgadd/diffutils.log 2>&1
          echo "pkgadd for diffutils returned status $?"
     fi
     test rm_pkgadd_conf = Y \
       && /bin/rm /etc/pkgadd.conf
     # Set up the prtpkg scripts in /root, letting the user edit them straight-forwardly:
     for ii in .vimrc \
               prtpkg_root.profile \
               prtpkg_shortcuts.bash.source \
               PPvars \
               prtpkg_pdrive \
               prtpkg_mkfs \
               prtpkg_mount \
               prtpkg_altinstall.txt \
               prtpkg_altinstall \
               prtpkg_presetup \
               prtpkg_setup \
               prtpkg_chroot \
               etc.rc.conf \
               etc.fstab \
               etc.hosts \
               etc.chrony.conf \
               etc.ntpd.conf \
               etc.rc.d.net \
               etc.resolv.conf \
               etc.pkgadd.conf \
               etc.pkgmk.conf \
               etc.prt-get.conf \
               var.lib.pkg.prt-get.aliases \
               extlinux.conf \
               extlinF1.msg \
               extlinF2.msg \
               prtpkg_getdrives.awk \
               prtpkg_locking.dash.source \
               .toprc
     do  /_/l/PP/bin/editfile $ii
     done
     . /root/prtpkg_root.profile # Update PS1 to know which environment we're in
                                 # Note ISORAM has no .profile initially, but
                                 # prtpkg_pdrive will build one from this file plus
                                 # prtpkg_shortcuts.bash.source
     /bin/cp -p /_/l/PP/bin/prtpkg_mkfsstatus    /root/mkfsstat
fi
# If this is an UEFI-capable platform as far as the CRUX ISO is concerned, ask the user
# if BIOS is prefered and set PPBORU accordingly:
if   test -f /sys/firmware/efi/efivars
then echo 'This platform booted with UEFI firmware.  If you know it supports legacy BIOS in'
     echo 'the BIOS setup and you want to normally use that configuration, reply with a "Y";'
     echo 'otherwise, enter a null string to use the UEFI mode that requires a functional'
     read -p 'EFI System Partition (ESP) exists or is created on the primary SSD/HDD):  ' PPBORU
     if   test ".$PPBORU" = '.'
     then export PPBORU='UEFI'
     elif test ".$PPBORU" = '.Y'
     then export PPBORU='BIOS'
     else echo "Your reply <$PPBORU> was neither 'Y' nor null--PPBORU is set to null."
          export PPBORU=''
     fi
else export PPBORU='BIOS'
fi
# Invoke prtpkg_prive without the build argument to ascertain the driv(s) situation:
/root/prtpkg_pdrive 2>&1 \
| /usr/bin/tee /root/prtpkg_pdrive.info.log
if   test -f /root/PPvars
then . /root/PPvars
else echo 'prtpkg_source found no /root/PPvars file returned by prtpkg_pdrive!'
fi
/bin/cat <<EOD >>/root/PPvars # Extend the /root/PPvars file
export PPBORU='$PPBORU'
export MEDIA='$MEDIA'
export ISOLABEL='$ISOLABEL'
export PPTYPE='$PPTYPE'
export PPHOST='$PPHOST'
export EDITOR='$EDITOR'
EOD
/usr/bin/sort /root/PPvars | /usr/bin/uniq >/root/PPvars.new
test $? -eq 0 && /bin/mv /root/PPvars{.new,}
/bin/cp -p /root/PPvars $PPCDIR
/bin/df -alHT | /usr/bin/grep ^/dev/[sh]d
/_/l/PP/bin/mntdrvs
```

```
echo "prtpkg_source has finished preparing for the invocation of the /root/prtpkg_presetup"
echo "script.  If the primary SSD/HDD needs to be (re)partitioned and (re)formatted, run the"
echo "'/root/prtpkg_pdrive build' script"
echo "before    (if you decide to create a persistent maintenance environment (CME) root"
echo "          filesystem tree--be sure to follow the procedure documented in the"
echo "          persistent_filesystem.txt file before running /root/prtpkg_presetup!)"
echo "or after  (if you decide otherwise)"
echo "you run the /root/prtpkg_presetup script.  If the primary SSD/HDD is ready for CRUX"
echo "maintenance, just run /root/prtpkg_presetup to prepare the environment to run the"
echo "/root/prtpkg_setup script on the production CRUX system you intend to maintain--you"
echo "do NOT want to invoke prtpkg_pdrive build processing in this situation."
if   test $PPTYPE = 'ISO'
then echo ""
     echo "If you have not yet decided to establish an CME root filesystem as described in the"
     echo "PP-stick's persistent_filesystem.txt file, you can now run /root/prtpkg_presetup."
     echo "It will simply stop if the RAMdrive is too small for complete package building"
     echo "capability to be installed and tell you so.  If it does stop (the platform likely"
     echo "has less than 2 GB of RAM) and you don't need the full build capability, you can"
     echo "then change the MUSTBUILD variable in the prtpkg_presetup script to false (null)"
     echo "before rerunning it, and it will only install needed packages not in the ISO from"
     echo "the prebuilt packages you have put in the PP-stick."
fi
```

7.6.17. rate.awk

```
#!/usr/bin/gawk
# CCIA_0.0    PP-stick bin/rate.awk ...4....:....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright ©  1997-2015 Larry Doolittle at http://doolittle.icarus.com/ntpclient/ and licensed
# * through the GNU General Public License available at http://www.gnu.org/copyleft/gpl.html.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#---------------------------------------------------------------------------------------------------
# Internally undocumented script of the Larry Doolittle's ntpclient package
#---------------------------------------------------------------------------------------------------
(FNR==1){ day_off=$1;        t1=$2; o1=$5; f1=$7}
        { t2=($1-day_off)*86400+$2; o2=$5; f2=$7; if (f2!=f1) fwarn=1}

END{
  print "delta-t",t2-t1,"seconds"
  print "delta-o",o2-o1,"useconds"
  if (fwarn) print " *** frequency changed in the middle - don't use ***"
  slope=(o2-o1)/(t2-t1)
  print "slope",slope,"ppm"
  print "old frequency",f1,"(",f1/65536,"ppm)"
  f3=f1+int(slope*65536);
  print "new frequency",f3,"(",f3/65536,"ppm)"
}

# the final value is what you should push into the adjtimex(2) field.
# i.e., if the last line shows
# new frequency -1318109 ( -20.1127 ppm)
# you put -1318109 into the -f switch of adjtimex (e.g., adjtimex -f -1318109)
```

7.7. PP-stick base Directory Template Files

7.7.1. PPvars—ISO/CME Environment Variables

Set optional variables (checked as **Opt**) to [N/A] to decline the option. Values for null variables will be interactively solicited unless they can be determined programmatically. Those checked as **Dyn** (dynamic) should be left as null to allow their values to be computed. Note CME stands for CRUX Maintenance Environment (ME for short) and PCE stands for Production CRUX Environment (PC for short).

7.7.1.1. Variables Established By prtpkg_source

Variable	Opt	Dyn	Description and Notes
PPHOST			**host**name of the platform (must not be crux)
PPTYPE		✔	install/upgrade system **type**; i.e., ISO or CME
EDITOR			$EDITOR—only vi and nano are currently supported
ISOLABEL		✔	ISO partition **label**
MEDIA		✔	path to ISO **media**
PPBORU			**B**IOS **or U**EFI; i.e., BIOS or UEFI
PPPDRV			**p**rimary **dr**i**v**e; e.g., hda, sdb
PPESPL	✔		**E**FI **s**ystem **p**artition **l**abel; e.g., XA
PPBOTL	✔		**boot** fs **l**abel; e.g., XB
PPMERL	✔		**ME r**oot fs **l**abel; e.g., ME
PPMEMP	✔		**ME r**oot fs **m**ount **p**oint; e.g., /_/l/$PPMERL
PPTCRL			**t**arget **C**RUX **r**oot fs **l**abel; e.g., XD
PPTCMP			**t**arget **C**RUX root fs **m**ount **p**oint; e.g., /_/l/$PPTCRL
PPTARD		✔	**tar**get CRUX **d**rive; e.g., sda7

7.7.1.2. Variables Established By prtpkg_presetup

Variable	Opt	Dyn	Description and Notes
MUSTBUILD			Set to null (false) if you want to install pre-built non-ISO packages from the PP-stick even if space to build from crux.nu ports exists—true requires building needed packages downloaded from the Internet.
NEEDNFS			If single-platform commonwealth, set to null (false); otherwise, set to true to install and configure opt/nfstools.

```
#!/bin/bash
# CCIA_0.0    PP-stick base/PPvars ....4....:....5....:....6....:....7....:....8....:....9....:....0
# *****************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# **************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization...............
#---------------------------------------------------------------------------------------------------
# See the CRUX 3.4 Commonwealth Manual for the CCI identified in the second line for the primary
# documentation of this file.
#
# Set optional variables to '[N/A]' to decline the option.  Values for null variables will be
# interactively solicited unless they can be determined programmatically (those marked 'dynamic'
# should be allowed to be so determined).  Note CME stands for CRUX Maintenance Environment (or ME
# for short) and PCE stands for Production CRUX Environment (or PC for short).
#
# Variables established by prtpkg_source: ----------------------------------------------------------
export PPHOST=''                 # HOSTname of the platform (must not be 'crux')
export PPTYPE=''                 # install/upgrade system TYPE; i.e., 'ISO' or 'CME'         dynamic
export EDITOR=''                 # $EDITOR, only 'vi' and 'nano' are currently supported
export ISOLABEL=''               # ISO partition LABEL                                       dynamic
export MEDIA=''                  # path to ISO MEDIA                                         dynamic
export PPBORU=''                 # Bios OR Uefi; i.e., 'BIOS' or 'UEFI'
export PPPDRV=''                 # Primary DRiVe; e.g., 'hda', 'sdb'
export PPESPL=''                 # Efi System Partition Label; e.g., 'XA'                   optional
export PPBOTL=''                 # BOoT fs Label; e.g., 'XB'                                optional
export PPMERL=''                 # ME Root fs Label; e.g., 'ME'                             optional
export PPMEMP=''                 # ME root fs Mount Point; e.g., "/_/l/$PPMERL"             optional
export PPTCRL=''                 # Target Crux Root fs Label; e.g., 'XD
export PPTCMP=''                 # Target Crux root fs Mount Point; e.g., "/_/l/$PPTCRL"
export PPTARD=''                 # TARget crux Drive; e.g., sda7                             dynamic
# Variables established by prtpkg_presetup: --------------------------------------------------------
export MUSTBUILD='true'          # Set to '' if you want to install pre-built non-ISO
                                 # packages from the PP-stick even if space to build from
                                 # crux.nu ports exists--true requires building needed
                                 # packages from the Internet
export NEEDNFS='true'            # If single-platform commonwealth, set to '' (false)
# ------------------------------ Begin network variables
export NETDEV='enp?s0'           # Interface ID to network through
export IPV4='ddd.ddd.ddd.ddd'    # Interface's IPv4 address
export CIDR='dd'                 # Number of network bits in the IPV4 address
export IPV4GW='ddd.ddd.ddd.ddd'  # Default IPv4 gateway address
export DOMAINDOT='domainname.'   # /etc/resolv.conf domain/search
export DNSHOST='ddd.ddd.ddd.ddd' # /etc/resolv.conf nameserver
# ------------------------------ End of network variables, begin NTP variables
export NTPD='chrony'             # Choose chrony, ntpd, or ntpclient for NTP software
                                 # If using ntpd, enable contrib prtdir prt-get.conf file
export NTPHOST='ddd.ddd.ddd.ddd' # IPv4 of the nearby NTP server for NTP protocol software
#xport NTPHOST='92.242.140.21'   # IPv4 of the nearby NTP server (time-b.nist.gov)
#xport NTPHOST='104.245.32.240'  # IPv4 of the nearby NTP server (cns1.atlantic.net)
                                 # chrony variables:
                                 #    none (but edit the etc.chrony.conf file)
                                 # Mills' ntpd variables:
                                 #    none (but edit the etc.ntp.conf file)
                                 # Doolittle's' ntpclient variables:
#xport ADJFREQ=''                #   ntpclient -f value (null for unknown)
#------------------------------- End of NTP variables, begin rootfs space calculation variables
#                                Ensure the ISORAM root filesystem and platform memory have
#                                enough free space to install the software needed to prepare
#                                the ISO/CME to run the prtpkg_setup script, else quit.  If
#                                there is not enough space to properly build non-ISO packages
#                                from source, then we will simply install pre-built packages
#                                from the PP-stick.  All numeric variables are KiB numbers
#                                (units of 1024; e.g., 1024=1MiB=1024*1024 bytes).
export RAMFSBOOT=315476          # Minimum root fs space needed to boot the RAM root fs ISO system;
                                 # i.e., the RAM root fs space in use after booting the canonical
                                 # environment
export PRMFSBOOT=1289124         # Minimum root fs space needed to boot a permanent root filesystem
export RFSTOBUILD=288788         # Minimum free root filesystem space needed to install all
                                 # pre-built packages needed to build and install non-ISO pre-built
                                 # packages
export RFSNONISO=100000          # Minimum free root filesystem space needed after installing build
                                 # tools to build and install non-ISO packages
```

```
export MEMMINRAM=368580          # Used memory after booting the canonical environment
export MEMMINPRM=172588          # Used memory after booting the permanent environment
export MEMTOBUILD=429544         # Minimum available memory needed to install all pre-built packages
                                 # needed to build and install non-ISO pre-built packages
export MEMNONISO=100000          # Minimum available memory after installing build tools needed to
                                 # build and install non-ISO packages
# - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

7.7.2. etc.chrony.conf

```
# CCIA_0.0    PP-stick base/etc.chrony.conf :....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2000-2018 by Per Lidén and the CRUX development team, and licensed through the GNU
# * General Public License available at http://www.gnu.org/copyleft/gpl.html.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# This is the CRUX 3.4 configuration file for port opt/chrony with added commentary/samples by
# David L. Craig <dlc.usa@gmail.com> for the CRUX Commonwealth Extension.
#----------------------------------------------------------------------------------------------------
# Provide nearby (preferably no router hops) time servers provisioned by the enterprise:
server $NTPHOST iburst
#--------
# You may find a public (open access) stratum 1 server nearby--visit
# http://support.ntp.org/bin/view/Servers/StratumOneTimeServers
# and comply with the rules of engagement it links to.  For each
# system therein, you can click on the entry's country code to see
# the details for using that particular system.  For example,
# time-b.nist.gov stratum 1 pubic server:
#   per sysadmin's request (http://support.ntp.org/bin/view/Servers/PublicTimeServer000278):
#     - USA only
#     - no need to notify them you will be using this server
#     - do not use the DNS name -- code IPv4 or IPv6 address
#     - request time no more frequently than once every 3 minutes per requesting
#       non-NATed address
#   per official NIST time service info at
#   https://www.nist.gov/pml/time-and-frequency-division/services/internet-time-service-its
#     - no locality preference noted
#     - no need to notify them you will be using this server
#     - use DNS name or IP address
#     - request time no more frequently than once every 4 seconds per requesting address
# server time-b.nist.gov iburst
#--------
# Very nearby stratum 2 public servers:
# server cns1.atlantic.net iburst
# server 104.245.32.240 iburst
#--------
# Use public NTP servers from the pool.ntp.org project.
pool 2.pool.ntp.org iburst
pool 2.pool.ntp.org iburst
#--------
# Record the rate at which the system clock gains/losses time.
driftfile /var/lib/chrony/drift
#--------
# Allow the system clock to be stepped in the first three updates
# if its offset is larger than 1 second.
makestep 1.0 3
#--------
# Enable kernel synchronization of the real-time clock (RTC).
rtcsync
# Define the hosts/subnets that may use this host as an NTP server
# allow 192.168.0.0/24
# allow 192.168.0.17/24
# allow 10.220.186.0/25
# allow 10.220.186.128/25
# Define the hosts/subnets that may send chronyc inquiries to this host
# bindcmdaddress 0.0.0.0
# cmdallow 10.220.186.128/25
```

7.7.3. etc.fstab

```
# CCIA_0.0    PP-stick base/etc.chrony.conf :....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2000-2018 by Per Lidén and the CRUX development team, and licensed through the GNU
# * General Public License available at http://www.gnu.org/copyleft/gpl.html.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# This is the CRUX 3.4 /etc/fstab configuration file with added commentary/samples by
# David L. Craig <dlc.usa@gmail.com> for the CRUX Commonwealth Extension.
#----------------------------------------------------------------------------------------------------
#
# /etc/fstab: static file system information
#
# <file system>        <dir>            <type>    <options>                  <dump> <pass>

LABEL=XD               /                ext4      defaults,noatime                1      1
/                      /_/l/ZD          none      bind
LABEL=XB               /_/l/ZB          ext4      defaults,noatime                0      0
/_/l/ZB                /boot            none      bind
LABEL=XA               /_/l/ZA          vfat      defaults                        0      0
/_/l/ZA                /boot/efi        none      bind
LABEL=X1               swap             swap      defaults                        0      0
LABEL=X2               swap             swap      defaults                        0      0
LABEL=X3               swap             swap      defaults                        0      0
LABEL=X4               swap             swap      defaults                        0      0
devpts                 /dev/pts         devpts    noexec,nosuid,gid=tty,mode=0620 0      0
shmfs                  /dev/shm         tmpfs     nodev,nosuid,noexec             0      0
tmp                    /tmp             tmpfs     noauto,defaults                 0      0
usb                    /proc/bus/usb    usbfs     noauto,defaults                 0      0
/dev/cdrom             /cdrom           iso9660   noauto,ro,user,unhide           0      0
/dev/dvd               /dvd             udf       noauto,ro,user,unhide           0      0
/dev/floppy/0          /floppy          vfat      noauto,user,unhide              0      0
LABEL=CRUX-EFI         /_/l/CRUX-EFI    vfat      noauto,defaults,ro,noatime      0      0
LABEL=CRUX-3.4         /_/l/CRUX-3.4    iso9660   noauto,defaults,ro,noatime      0      0
LABEL=CRUX-3.4-updated /_/l/CRUX-3.4    iso9660   noauto,defaults,ro,noatime      0      0
LABEL=FREEDOS2016      /_/l/FREEDOS2016 vfat      noauto,defaults,rw,noatime      0      0
LABEL=FD-SETUP         /_/l/FD-SETUP    vfat      noauto,defaults,rw,noatime      0      0

# End of file
```

7.7.4. etc.hosts

```
# CCIA_0.0    PP-stick base/etc.hosts .4....:....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2000-2018 by Per Lidén and the CRUX development team, and licensed through the GNU
# * General Public License available at http://www.gnu.org/copyleft/gpl.html.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# This is a CRUX 3.4 configuration file for port core/glibc with added commentary/samples by
# David L. Craig <dlc.usa@gmail.com> for the CRUX Commonwealth Extension.
#----------------------------------------------------------------------------------------------------
#
# /etc/hosts: static lookup table for host names
#

# IPv4
127.0.0.1     localhost
#<ip-address> <hostname.domain.org>      <aliases>

# IPv6
#::1                        ip6-localhost ip6-loopback
#fe00::0      ip6-localnet
#ff00::0      ip6-mcastprefix
#ff02::1      ip6-allnodes
#ff02::2      ip6-allrouters
#ff02::3      ip6-allhosts
```

```
# End of file
```

7.7.5. etc.ntpd.conf

```
# CCIA_0.0    PP-stick base/etc.ntpd.conf ..:....5....:....6....:....7....:....8....:....9....:....0
# ****************************************************************************************************
# * Copyright © 2000-2018 by Per Lidén and the CRUX development team, and licensed through the GNU
# * General Public License available at http://www.gnu.org/copyleft/gpl.html.
# ****************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# This is a CRUX 3.4 configuration file for port contrib/ntp with added commentary/samples by
# David L. Craig <dlc.usa@gmail.com> for the CRUX Commonwealth Extension.
#----------------------------------------------------------------------------------------------------
#--------
# Provide nearby (preferably no router hops) time servers provisioned by the enterprise:
server $NTPHOST prefer iburst
#--------
# You may find a public (open access) stratum 1 server nearby--see notes in chrony.conf above
# server time-b.nist.gov prefer iburst
# server 129.6.15.29 prefer iburst
#--------
# Very nearby stratum 2 public servers:
# server cns1.atlantic.net prefer iburst
# server 104.245.32.240 prefer iburst
#--------
# Random stratum 2 public servers:
server 2.pool.ntp.org
server 2.pool.ntp.org
#--------
# The local system clock (last ditch source):
server 127.127.1.0
fudge  127.127.1.0 stratum 10
#--- Define remote and localhost network services permissions/restrictions: ---------------
# Any host not covered below can only receive time service:
restrict default noquery nopeer notrap
#--------
# Only allow read-only access from localhost:
restrict 127.0.0.1
restrict ::1
#--------
# Edit the NTP clients/subnets which should be able to designate our server as a time
# source and/or direct read-only ntpq requests here (notice that the noquery has been
# implicitly removed) -- add noserve flags to decline time service requests
# restrict 192.168.1.0    mask 255.255.255.0   nomodify
# restrict 192.168.17.0   mask 255.255.255.0   nomodify
# restrict 10.220.186.128 mask 255.255.255.128 nomodify
# restrict 10.220.186.0   mask 255.255.255.128 nomodify
#--- Other configuration items: ----------------------------------------------------------
enable kernel ntp
# Default paths
logfile   /var/log/ntp.log
pidfile   /var/run/ntp/ntpd.pid
driftfile /var/lib/ntp/ntp.drift
statsdir  /var/lib/ntp/stats/
# Avoid udp spoofed ddos attacks
disable monitor
```

7.7.6. etc.pkgadd.conf

```
# CCIA_0.0    PP-stick base/etc.pkgadd.conf :....5....:....6....:....7....:....8....:....9....:....0
# ****************************************************************************************************
# * Copyright © 2000-2018 by Per Lidén and the CRUX development team, and licensed through the GNU
# * General Public License available at http://www.gnu.org/copyleft/gpl.html.
# ****************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# This is a CRUX 3.4 configuration file for port core/pkgutils.
#----------------------------------------------------------------------------------------------------
```

```
#
# /etc/pkgadd.conf: pkgadd(8) configuration
#

# Default rule (implicit)
#UPGRADE     ^.*$                                     YES

UPGRADE                    ^etc/.*$                   NO
UPGRADE                    ^var/log/.*$               NO
UPGRADE                    ^var/spool/\w*cron/.*$     NO
UPGRADE                    ^var/run/utmp$                          NO

UPGRADE                    ^etc/ports/drivers/.*$     YES
UPGRADE                    ^etc/X11/.*$               YES

UPGRADE                    ^etc/rc.*$                 YES
UPGRADE                    ^etc/rc\.local$                         NO
UPGRADE                    ^etc/rc\.modules$          NO
UPGRADE                    ^etc/rc\.conf$                          NO
UPGRADE                    ^etc/rc\.d/net$                         NO

UPGRADE                    ^etc/udev/rules.d/.*$      YES
UPGRADE                    ^etc/udev/rules.d/1.*$     NO
UPGRADE                    ^etc/udev/hwdb.d/.*$       YES
UPGRADE                    ^etc/udev/hwdb.bin$        YES

UPGRADE                    ^etc/ssl/cert.pem$         YES
# End of file
```

7.7.7. etc.pkgmk.conf

```
# CCIA_0.0    PP-stick base/etc.pkgmk.conf .:....5....:....6....:....7....:....8....:....9....:....0
# **************************************************************************************************
# * Copyright © 2000-2018 by Per Lidén and the CRUX development team, and licensed through the GNU
# * General Public License available at http://www.gnu.org/copyleft/gpl.html.
# **************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#---------------------------------------------------------------------------------------------------
# This is a CRUX 3.4 configuration file for port core/pkgutils with added commentary/samples by
# David L. Craig <dlc.usa@gmail.com> for the CRUX Commonwealth Extension.
#---------------------------------------------------------------------------------------------------
#
# /etc/pkgmk.conf: pkgmk(8) configuration
#

export CFLAGS="-O2 -march=x86-64 -pipe"
export CXXFLAGS="${CFLAGS}"

# export MAKEFLAGS="-j2"

case ${PKGMK_ARCH} in
             "64"|"")
                           ;;
             "32")
                           export CFLAGS="${CFLAGS} -m32"
                           export CXXFLAGS="${CXXFLAGS} -m32"
                           export LDFLAGS="${LDFLAGS} -m32"
                           export PKG_CONFIG_LIBDIR="/usr/lib32/pkgconfig"
                           ;;
             *)
                           echo "Unknown architecture selected! Exiting."
                           exit 1
                           ;;
esac

# PKGMK_SOURCE_MIRRORS=()
# PKGMK_SOURCE_DIR="$PWD"
# PKGMK_PACKAGE_DIR="$PWD"
# PKGMK_WORK_DIR="$PWD/work"
# PKGMK_DOWNLOAD="no"
# PKGMK_IGNORE_FOOTPRINT="no"
# PKGMK_IGNORE_NEW="no"
# PKGMK_NO_STRIP="no"
# PKGMK_DOWNLOAD_PROG="wget"
# PKGMK_WGET_OPTS=""
# PKGMK_CURL_OPTS=""
# PKGMK_COMPRESSION_MODE="gz"

# End of file
```

7.7.8. etc.prt-get.conf

```
# CCIA_0.0    PP-stick base/etc.prt-get.conf ....5....:....6....:....7....:....8....:....9....:....0
# **************************************************************************************************
# * Copyright © 2000-2018 by Per Lidén and the CRUX development team, and licensed through the GNU
# * General Public License available at http://www.gnu.org/copyleft/gpl.html.
# **************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#---------------------------------------------------------------------------------------------------
# This is a CRUX 3.4 configuration file for port core/prt-get with added commentary/samples by
# David L. Craig <dlc.usa@gmail.com> for the CRUX Commonwealth Extension.
#---------------------------------------------------------------------------------------------------
###
### prt-get conf
###

# note: the order matters: the package found first is used
prtdir /usr/ports/core
prtdir /usr/ports/opt
prtdir /usr/ports/xorg
# prtdir /usr/ports/compat-32
# prtdir /usr/ports/contrib

# the following line enables the multilib compat-32 collection
#prtdir /usr/ports/compat-32

# the following line enables the user maintained contrib collection
#prtdir /usr/ports/contrib

### use mypackage form local directory
# prtdir /home/packages/build:mypackage

### log options:
# writelog enabled         # (enabled|disabled)
# logmode  overwrite       # (append|overwrite)
# rmlog_on_success yes     # (no|yes)
logfile  /var/log/pkgbuild/%n.log
                           # path, %p=path to port dir, %n=port name
                           #       %v=version, %r=release

### use alternate cache file (default: /var/lib/pkg/prt-get.cache
# cachefile /mnt/nfs/cache

### print README information:
# readme verbose           # (verbose|compact|disabled)

### prefer higher versions in sysup / diff
# preferhigher no      # (yes|no)

### use regexp search
# useregex no        # (yes|no)

### run pre- and post-installs scripts; yes is equivalent to the
### --install-scripts option
# runscripts no            # (no|yes)


### EXPERT SECTION ###

### alternative commands
# makecommand      pkgmk
# addcommand       pkgadd
# removecommand    pkgrm
# runscriptcommand sh
```

7.7.9. etc.rc.conf

```
# CCIA_0.0    PP-stick base/etc.rc.conf ....:....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2000-2018 by Per Lidén and the CRUX development team, and licensed through the GNU
# * General Public License available at http://www.gnu.org/copyleft/gpl.html.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# This is a CRUX 3.4 configuration file for port core/rc with added commentary/samples by
# David L. Craig <dlc.usa@gmail.com> for the CRUX Commonwealth Extension.
#----------------------------------------------------------------------------------------------------
#
# /etc/rc.conf: system configuration
#

FONT=lat1-12
KEYMAP=us
TIMEZONE=UTC
HOSTNAME=$MEHOST # Replace $MEHOST with your platform's host name
SYSLOG=sysklogd
SERVICES=(lo net crond)

# End of file
```

7.7.10. etc.rc.d.net

```
#!/bin/sh
# CCIA_0.0    PP-stick base/etc.rc.d.net ...:....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2000-2018 by Per Lidén and the CRUX development team, and licensed through the GNU
# * General Public License available at http://www.gnu.org/copyleft/gpl.html.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# This is a CRUX 3.4 configuration file for port core/iproute2 with added commentary/samples by
# David L. Craig <dlc.usa@gmail.com> for the CRUX Commonwealth Extension.
#----------------------------------------------------------------------------------------------------
#
# /etc/rc.d/net: start/stop network interface
#

# Connection type: "DHCP" or "static"
TYPE="DHCP"

# For "static" connections, specify your settings here:
# To see your available devices run "ip link".
DEV=enp11s0
ADDR=192.168.1.100
MASK=24
GW=192.168.1.1

# Optional settings:
DHCPOPTS="-t 10"

case $1 in
        start)
                if [ "${TYPE}" = "DHCP" ]; then
                        /sbin/dhcpcd ${DHCPOPTS}
                else
                        /sbin/ip addr add ${ADDR}/${MASK} dev ${DEV} broadcast +
                        /sbin/ip link set ${DEV} up
                        /sbin/ip route add default via ${GW}
                fi
                ;;
        stop)
                if [ "${TYPE}" = "DHCP" ]; then
```

```
                                        /sbin/dhcpcd -x
                           else
                                        /sbin/ip route del default
                                        /sbin/ip link set ${DEV} down
                                        /sbin/ip addr del ${ADDR}/${MASK} dev ${DEV}
                           fi
                           ;;
             restart)
                           $0 stop
                           $0 start
                           ;;
             *)
                           echo "Usage: $0 [start|stop|restart]"
                           ;;
esac

# End of file
```

7.7.11. etc.resolv.conf

```
# CCIA_0.0    PP-stick base/etc.resolv.conf :....5....:....6....:....7....:....8....:....9....:....0
# ****************************************************************************************************
# * Copyright © 2000-2018 by Per Lidén and the CRUX development team, and licensed through the GNU
# * General Public License available at http://www.gnu.org/copyleft/gpl.html.
# ****************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. .............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# This is a CRUX 3.4 configuration file for port core/glibc with added commentary/samples by
# David L. Craig <dlc.usa@gmail.com> for the CRUX Commonwealth Extension.
#----------------------------------------------------------------------------------------------------
#
# /etc/resolv.conf: resolver configuration file
#

#search <domain.org>
#nameserver <ip-address>

# End of file
```

7.7.12. extlinF1.msg

```
^L^U#$PPHOST boot loader - Info via F1-F2 - F1: Boot Overview (default: ism_)
Global: ro LANG=en_US.UTF-8 console=ttyS0,38400 console=tty0
Kernels (F2 shows legend):
 hdt - SYSLINUX Hardware Detection Tool
 isR - CRUX-3.3 ISO kernel
 xaR - simple,fb_vesa,fbcon,!agpgart,!intel_GMA3150,vgaarb
where R is 'd' for [XD] /dev/sda8 or 'm' for [XM] /dev/sda7
Suffixes: -adds-    --screen----    --stty-columns-x-rows-provided-by-kernel---
       -string--    --provided--    --xx--
 a   - vga=0xF00     800  600        80x33
 b   - vga=0xF01     800  600        80x33
 c   - vga=0xF02     800  600        80x28
 d   - vga=0xF03     800  600        80x32
 e   - vga=0xF05     800  600        80x40
 f   - vga=0xF06     800  600        80x39
 g   - vga=0xF07     800  600        80x40
 h   - vga=0x312     640  480 32     80x40
 i   - vga=0x314     800  600 16    100x50
 j   - vga=0x315     800  600 32    100x50
 k   - vga=0x301     640  480  8     80x40
 l   - vga=0x303     800  600  8    100x50
 m   - vga=0x311     640  480 16     80x40
 _   - vga=ask
Press <Tab> for available selections, select one, add any parms, press <Enter>
```

7.7.13. extlinF2.msg

```
^L^U#$PPHOST boot loader - Info via F1-F2 - F2: Kernel String Legend

hdt is not a kernel, takes no suffix, and is not part of this table.
```

```
1st character:
 i - CRUX 3.3 Linux 4.9.6 ISO
 x - CRUX 3.3 Linux 4.9.6 (config notes in F1 page)

2nd character:
 a - production, gui=n    c - test, gui=n    s - standard ISO
 b - production, gui=y    d - test, gui=y    x - experimental

3rd character:
 d - [XD] root=/dev/sda8  m - [XM] root=/dev/sda7
```

7.7.14. extlinux.conf

```
SERIAL 0 38400
DISPLAY extlinF1.msg
DEFAULT ism_
PROMPT 1
APPEND ro LANG=en_US.UTF-8 console=ttyS0,38400 console=tty0

F1 extlinF1.msg
F2 extlinF2.msg

LABEL hdt
  KERNEL hdt.c32
  APPEND -

LABEL xad
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda8

LABEL xada
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda8 vga=0xF00

LABEL xadb
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda8 vga=0xF01

LABEL xadc
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda8 vga=0xF02

LABEL xadd
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda8 vga=0xF03

LABEL xade
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda8 vga=0xF05

LABEL xadf
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda8 vga=0xF06

LABEL xadg
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda8 vga=0xF07

LABEL xadh
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda8 vga=0x312

LABEL xadi
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda8 vga=0x314

LABEL xadj
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda8 vga=0x315

LABEL xadk
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda8 vga=0x301

LABEL xadl
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda8 vga=0x303

LABEL xadm
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda8 vga=0x311

LABEL xad_
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda8 vga=ask

LABEL isd
```

```
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda8

LABEL isda
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda8 vga=0xF00

LABEL isdb
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda8 vga=0xF01

LABEL isdc
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda8 vga=0xF02

LABEL isdd
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda8 vga=0xF03

LABEL isde
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda8 vga=0xF05

LABEL isdf
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda8 vga=0xF06

LABEL isdg
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda8 vga=0xF07

LABEL isdh
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda8 vga=0x312

LABEL isdi
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda8 vga=0x314

LABEL isdj
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda8 vga=0x315

LABEL isdk
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda8 vga=0x301

LABEL isdl
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda8 vga=0x303

LABEL isdm
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda8 vga=0x311

LABEL isd_
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda8 vga=ask

LABEL xam
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda7

LABEL xama
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda7 vga=0xF00

LABEL xamb
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda7 vga=0xF01

LABEL xamc
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda7 vga=0xF02

LABEL xamd
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda7 vga=0xF03

LABEL xame
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda7 vga=0xF05

LABEL xamf
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda7 vga=0xF06

LABEL xamg
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda7 vga=0xF07

LABEL xamh
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda7 vga=0x312
```

```
LABEL xami
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda7 vga=0x314

LABEL xamj
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda7 vga=0x315

LABEL xamk
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda7 vga=0x301

LABEL xaml
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda7 vga=0x303

LABEL xamm
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda7 vga=0x311

LABEL xam_
  KERNEL vmlinuz-4.9.6_xa  root=/dev/sda7 vga=ask

LABEL ism
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda7

LABEL isma
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda7 vga=0xF00

LABEL ismb
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda7 vga=0xF01

LABEL ismc
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda7 vga=0xF02

LABEL ismd
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda7 vga=0xF03

LABEL isme
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda7 vga=0xF05

LABEL ismf
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda7 vga=0xF06

LABEL ismg
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda7 vga=0xF07

LABEL ismh
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda7 vga=0x312

LABEL ismi
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda7 vga=0x314

LABEL ismj
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda7 vga=0x315

LABEL ismk
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda7 vga=0x301

LABEL isml
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda7 vga=0x303

LABEL ismm
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda7 vga=0x311

LABEL ism_
  KERNEL vmlinuz-4.9.6_iso root=/dev/sda7 vga=ask
```

7.7.15. prtpkg_altinstall

```
#!/bin/dash
# CCIA_0.0    PP-stick base/prtpkg_altinstall ...5....:....6....:....7....:....8....:....9....:....
# ***************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization.............
#---------------------------------------------------------------------------------------------------
# Installs the packages in the /root/prtpkg_altinstall.txt file outside the prtpkg_setup script.
#---------------------------------------------------------------------------------------------------
set +xv
test ! -d $PPMEMP/var/lib/pkg \
  && /bin/mkdir -p $PPMEMP/var/lib/pkg
test ! -f $PPMEMP/var/lib/pkg/db \
  && : > $PPMEMP/var/lib/pkg/db
test ! -d $PPMEMP/etc \
  && /bin/mkdir -p $PPMEMP/etc
test ! -f $PPMEMP/etc/pkgadd.conf \
  && /bin/cp -p /etc/pkgadd.conf $PPMEMP/etc
/bin/cat /root/prtpkg_altinstall.txt \
| while read a1 a2 a3 a4
  do    if   test $a2 = '#'
        then echo "Skipping $a3 $a4"
        else cmd="/usr/bin/pkgadd -r $PPMEMP -f $MEDIA/crux/$a1/$a2"
             echo "Command:$cmd" > /root/pkgadd/$a2.log 2>&1
             $cmd > /root/pkgadd/$a2.log 2>&1
             echo "rc $? for $a2"
        fi
  done 2>&1 | /usr/bin/tee /root/altinstall.log
```

7.7.16. prtpkg_altinstall.txt

```
core acl#2.2.52-2.pkg.tar.xz
core attr#2.4.47-2.pkg.tar.xz
core autoconf#2.69-2.pkg.tar.xz
core automake#1.15.1-1.pkg.tar.xz
core bash#4.4.12-1.pkg.tar.xz
core bc#1.07.1-1.pkg.tar.xz
core bin86#0.16.21-1.pkg.tar.xz
core bindutils#9.10.6-1.pkg.tar.xz
core binutils#2.27-1.pkg.tar.xz
core bison#3.0.4-2.pkg.tar.xz
core bzip2#1.0.6-2.pkg.tar.xz
core ca-certificates#20170606-1.pkg.tar.xz
core coreutils#8.28-1.pkg.tar.xz
core cpio#2.12-1.pkg.tar.xz
core curl#7.55.1-1.pkg.tar.xz
core dash#0.5.9.1-1.pkg.tar.xz
core db#5.3.28-1.pkg.tar.xz
core dcron#4.5-3.pkg.tar.xz
core dhcpcd#6.11.5-1.pkg.tar.xz
core diffutils#3.6-1.pkg.tar.xz
core e2fsprogs#1.43.6-1.pkg.tar.xz
core ed#1.14.2-1.pkg.tar.xz
core eudev#3.2.4-1.pkg.tar.xz
core exim#4.89-1.pkg.tar.xz
core file#5.32-1.pkg.tar.xz
core filesystem#3.3-1.pkg.tar.xz
core findutils#4.6.0-1.pkg.tar.xz
core flex#2.6.4-1.pkg.tar.xz
core gawk#4.1.4-1.pkg.tar.xz
core gcc#6.4.0-1.pkg.tar.xz
core gdbm#1.13-1.pkg.tar.xz
core gettext#0.19.8.1-1.pkg.tar.xz
core glibc#2.24-7.pkg.tar.xz
core glibc-32#2.24-7.pkg.tar.xz
core gperf#3.0.4-2.pkg.tar.xz
core grep#3.1-1.pkg.tar.xz
core groff#1.22.3-2.pkg.tar.xz
core gzip#1.8-1.pkg.tar.xz
core hdparm#9.52-1.pkg.tar.xz
core httpup#0.4.0m-1.pkg.tar.xz
core iana-etc#2.4-1.pkg.tar.xz
core inetutils#1.9.4-3.pkg.tar.xz
core iproute2#4.13.0-1.pkg.tar.xz
core iptables#1.6.1-1.pkg.tar.xz
core iputils#s20151218-2.pkg.tar.xz
core kbd#2.0.4-1.pkg.tar.xz
core kmod#24-1.pkg.tar.xz
core less#487-1.pkg.tar.xz
core libarchive#3.3.1-1.pkg.tar.xz
core libcap#2.25-2.pkg.tar.xz
core libdevmapper#1.02.142-1.pkg.tar.xz
core libgmp#6.1.2-1.pkg.tar.xz
core libmpc#1.0.3-1.pkg.tar.xz
core libmpfr#3.1.6-1.pkg.tar.xz
core libpcre#8.41-1.pkg.tar.xz
core libpipeline#1.4.1-1.pkg.tar.xz
core libtool#2.4.6-2.pkg.tar.xz
core libusb#1.0.21-1.pkg.tar.xz
core libusb-compat#0.1.5-1.pkg.tar.xz
core lilo#24.2-1.pkg.tar.xz
core lzo#2.10-1.pkg.tar.xz
core m4#1.4.18-1.pkg.tar.xz
core make#4.2.1-1.pkg.tar.xz
core man-db#2.7.6.1-1.pkg.tar.xz
core man-pages#4.12-1.pkg.tar.xz
core mlocate#0.26-2.pkg.tar.xz
core nasm#2.13.01-1.pkg.tar.xz
core ncurses#6.0-4.pkg.tar.xz
core openrdate#1.2-3.pkg.tar.xz
core openssh#7.5p1-2.pkg.tar.xz
core openssl#1.0.2l-1.pkg.tar.xz
core patch#2.7.5-2.pkg.tar.xz
core pciutils#3.5.5-1.pkg.tar.xz
core perl#5.24.2-1.pkg.tar.xz
core pkg-config#0.29.2-1.pkg.tar.xz
core pkgutils#5.40.3-1.pkg.tar.xz
core ports#1.5-22.pkg.tar.xz
core procps#3.3.12-1.pkg.tar.xz
core prt-get#5.19.1-1.pkg.tar.xz
core psmisc#23.1-1.pkg.tar.xz
core rc#2.30-1.pkg.tar.xz
core readline#7.0.3-1.pkg.tar.xz
core rsync#3.1.2-1.pkg.tar.xz
core sed#4.4-1.pkg.tar.xz
core shadow#4.5-1.pkg.tar.xz
core signify#0.1p1-1.pkg.tar.xz
core start-stop-daemon#20170812-1.pkg.tar.xz
core sudo#1.8.21p2-1.pkg.tar.xz
core sysfsutils#2.1.0-2.pkg.tar.xz
core sysklogd#1.5.1-6.pkg.tar.xz
core sysvinit#2.88-4.pkg.tar.xz
core tar#1.29-1.pkg.tar.xz
core time#1.7-3.pkg.tar.xz
core tzdata#2017b-1.pkg.tar.xz
core usbutils#008-2.pkg.tar.xz
core util-linux#2.30.1-1.pkg.tar.xz
core vim#8.0.1071-1.pkg.tar.xz
core wget#1.19.1-1.pkg.tar.xz
core which#2.21-2.pkg.tar.xz
core xz#5.2.3-1.pkg.tar.xz
core zlib#1.2.11-1.pkg.tar.xz
opt # alsa-lib#1.1.4.1-1.pkg.tar.xz
opt # at-spi2-atk#2.24.1-1.pkg.tar.xz
opt # at-spi2-core#2.24.1-1.pkg.tar.xz
opt # atk#2.24.0-1.pkg.tar.xz
opt # autoconf-2.13#2.13-1.pkg.tar.xz
opt btrfs-progs#4.13-1.pkg.tar.xz
opt # cairo#1.15.6-2.pkg.tar.xz
opt # cdrkit#1.1.11-3.pkg.tar.xz
opt # cmake#3.9.1-1.pkg.tar.xz
opt cryptsetup#1.7.5-1.pkg.tar.xz
opt dbus#1.10.22-1.pkg.tar.xz
opt dialog#1.3-20160828-1.pkg.tar.xz
opt dosfstools#4.0-1.pkg.tar.xz # UEFI_req
opt efibootmgr#14-1.pkg.tar.xz # UEFI_req
opt efivar#26-1.pkg.tar.xz # UEFI_req
opt elfutils#0.169-1.pkg.tar.xz
opt # expat#2.2.4-1.pkg.tar.xz
opt fakeroot#1.22-1.pkg.tar.xz prtpkg_req
opt # fetchmail#6.3.26-2.pkg.tar.xz
opt # firefox#52.3.0esr-1.pkg.tar.xz
opt # fontconfig#2.12.4-1.pkg.tar.xz
opt # freetype#2.8-1.pkg.tar.xz
opt # gdk-pixbuf#2.36.9-1.pkg.tar.xz
opt # glib#2.52.3-1.pkg.tar.xz
opt gnu-efi#3.0.5-2.pkg.tar.xz # UEFI_req
opt # gobject-introspection#1.52.1-1.pkg.tar.xz
opt # grub2#2.02-1.pkg.tar.xz
opt # grub2-efi#2.02-1.pkg.tar.xz
opt # gtk#2.24.31-1.pkg.tar.xz
opt # gtk3#3.22.19-1.pkg.tar.xz
opt # harfbuzz#1.5.0-1.pkg.tar.xz
opt # hicolor-icon-theme#0.17-1.pkg.tar.xz
opt # intltool#0.51.0-1.pkg.tar.xz
opt jfsutils#1.1.15-3.pkg.tar.xz
opt # keyutils#1.5.10-1.pkg.tar.xz # NFS_req
opt # libevent#2.1.8-1.pkg.tar.xz # NFS_req
opt # libffi#3.2.1-3.pkg.tar.xz
opt libgcrypt#1.8.1-1.pkg.tar.xz
opt libgpg-error#1.27-1.pkg.tar.xz
opt # libidl#0.8.14-1.pkg.tar.xz
opt # libjpeg-turbo#1.5.2-1.pkg.tar.xz
opt # libnfsidmap#0.27-1.pkg.tar.xz # NFS_req
opt # libnl#3.3.0-1.pkg.tar.xz
opt # libpng#1.6.32-1.pkg.tar.xz
opt # libtiff#4.0.8-1.pkg.tar.xz
opt # libtirpc#1.0.2-1.pkg.tar.xz # NFS_req
opt # libxml2#2.9.5-1.pkg.tar.xz
opt # libxml2-python#2.9.5-1.pkg.tar.xz
opt # libxslt#1.1.29-1.pkg.tar.xz
opt linux-firmware#20161122-1.pkg.tar.xz
opt # llvm#4.0.1-1.pkg.tar.xz
opt # lvm2#2.02.173-1.pkg.tar.xz
opt # mdadm#4.0-1.pkg.tar.xz
opt mtools#4.0.18-2.pkg.tar.xz
opt # mutt#1.9.0-1.pkg.tar.xz
opt nano#2.8.7-1.pkg.tar.xz
opt # nfs-utils#2.1.1-1.pkg.tar.xz # NFS_req
opt # nspr#4.15-1.pkg.tar.xz
opt # nss#3.31-2.pkg.tar.xz
opt # openbox#3.6.1-1.pkg.tar.xz
opt # p5-xml-parser#2.44-1.pkg.tar.xz
opt # pango#1.40.11-1.pkg.tar.xz
opt parted#3.2-1.pkg.tar.xz
opt # popt#1.16-2.pkg.tar.xz
opt # ppp#2.4.7-3.pkg.tar.xz
opt # python#2.7.13-1.pkg.tar.xz
opt reiserfsprogs#3.6.26-1.pkg.tar.xz
opt # rpcbind#0.2.4-2.pkg.tar.xz # NFS_req
opt # shared-mime-info#1.5-1.pkg.tar.xz
opt # sqlite3#3.20.1-1.pkg.tar.xz # NFS_req
opt # syslinux#6.03-3.pkg.tar.xz
opt # talloc#2.1.10-1.pkg.tar.xz
opt # unzip#6.0-6.pkg.tar.xz
opt wireless-tools#29-2.pkg.tar.xz
opt wpa_supplicant#2.6-2.pkg.tar.xz
opt # wvdial#1.61-7.pkg.tar.xz
opt xfsprogs#4.12.0-1.pkg.tar.xz
opt # xterm#327-1.pkg.tar.xz
opt # yasm#1.3.0-2.pkg.tar.xz
opt # zip#3.0-2.pkg.tar.xz
```

7.7.17. prtpkg_chroot

```
#!/bin/dash
# CCIA_0.0    PP-stick base/prtpkg_chroot ..:....5....:....6....:....7....:....8....:....9....:....0
# ***********************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***********************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#------------------------------------------------------------------------------------------------
# This script ensures the mount point is properly available, that all bind mounts necessary are
# established, then does the chroot.  It requires the mount point and label of the target root
# filesystem be provided as a sourcable dash file named /_/PP/$PPHOST/prtpkg_setup.vars, which is
# normally created by the prtpkg_presetup script.
#------------------------------------------------------------------------------------------------
set +xv
test ".$PPHOST" = '.' \
  && { echo 'The PPHOST environment variable has not been pre=defined--aborting!'
       exit 32
     }
test -f /_/l/PP/h/$PPHOST/PPvars \
  && .  /_/l/PP/h/$PPHOST/PPvars \
  || { echo "Did not find the /_/l/PP/h/$PPHOST/PPvars file."
       exit 32
     }
test ".$PPPCMP" = '.' \
  && { echo 'The PPPCMP environment variable has not been pre=defined--aborting!'
       exit 32
     }
test ".$PPMEMP" = '.' \
  && { echo 'The PPMEMP environment variable has not been pre=defined--aborting!'
       exit 32
     }
test    ".$PPTARM" != ".$PPPCMP" \
     -a ".$PPTARM" != ".$PPMEMP" \
  && { echo "The specified mount point <$PPTARM> does not match PPPCMP or PPMEMP--aborting!"
       exit 32
     }
/usr/bin/grep -q '^/dev/'$PPPDRV'[0-9]* '$PPTARM' ' /proc/mounts \
  || { echo "A PPPDRV <$PPPDRV> partition is not mounted at $PPTARM--aborting!"
       exit 32
     }
export PPRDEV=`/usr/bin/gawk '$2 == "'$PPTARM'" { sub("^.*[/]", "", $1) ; print $1 }' /proc/mounts`
export PPRLAB=`/bin/ls -lh /dev/disk/by-label \
               | /usr/bin/gawk '/[/]'$PPRDEV'$/ { split($0, a) ; print a[NF - 2] }' \
               `
echo "/dev/$PPRDEV (label $PPRLAB) is mounted on requested mount point $PPTARM"
test    ".$PPTARD" != ".$PPRDEV" \
     -o ".$PPTARL" != ".$PPRLAB" \
  && { echo "/_/l/PP/h/$PPHOST/PPvars variables disagree--aborting!"
       exit 32
     }
set -e
for ii in _ _/PP dev tmp proc sys
do  test -d $PPTARM/$ii \
      || /bin/mkdir $PPTARM/$ii
done
/usr/bin/grep -q '^[^ ]* '$PPTARM'/_/l/PP ' /proc/mounts \
  || /bin/mount --bind /_/PP $PPTARM/_/l/PP
/usr/bin/grep -q '^[^ ]* '$PPTARM'/dev ' /proc/mounts \
  || /bin/mount --bind /dev $PPTARM/dev
/usr/bin/grep -q '^[^ ]* '$PPTARM'/tmp ' /proc/mounts \
  || /bin/mount --bind /tmp $PPTARM/tmp
/usr/bin/grep -q '^[^ ]* '$PPTARM'/proc ' /proc/mounts \
  || /bin/mount -t proc proc $PPTARM/proc
/usr/bin/grep -q '^[^ ]* '$PPTARM'/sys ' /proc/mounts \
  || /bin/mount -t sysfs none $PPTARM/sys
/usr/bin/grep -q '^[^ ]* '$PPTARM'/dev/pts ' /proc/mounts \
  || /bin/mount -t devpts -o noexec,nosuid,gid=tty,mode=0620 devpts $PPTARM/dev/pts
# If UEFI platform, if $PPTARM/...efivars is not mounted, if /sys/.../efivars is mounted,
# then bind mount /sys/---/efivars as $PPTARM/sys/---/efivars
if   test -f /sys/firmware/efi/efivars
```

```
then /usr/bin/grep -s ' '$PPTARM'/sys/firmware/efi/efivars$' \
       && /usr/bin/grep -qs /sys/firmware/efi/efivars /proc/mounts \
       && /bin/mount --bind /sys/firmware/efi/efivars $PPTARM/sys/firmware/efi/efivars
fi
PS1SAVE="$PS1"
export PS1='\[\033[01;31m\]$PPHOST[$PPRLAB](chroot)\[\033[01;37m\]\w \[\033[01;33m\]\$ \[\033[00m\]'
echo "Environment variables:"
/usr/bin/env \
| /usr/bin/sort
cmd="/usr/bin/chroot $PPTARM /bin/bash"
echo "Invoking $cmd..."
$cmd
echo "chroot return status code $?"
export PS1="$PS1SAVE"
```

7.7.18. prtpkg_getdrives.awk

```
#!/usr/bin/gawk
# CCS1_0.0 -- PP-stick base/prtpkg_getdrives.awk 5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# This script identifies candidate installation drives on a platform and reports their status.
#----------------------------------------------------------------------------------------------------
function new_drv() {
  Nam = $4
  Siz = $3
  Drives = Drives Nam "\t"
  NumParts = 0
  next
}
function print_it() {
  printf("Found %7.1f GiB /dev/%s with %3.1u partitions [",
         Siz / 1048576., Nam, NumParts) >> "/dev/stderr"
  for ( Ii = 1 ; Ii <= NumDrvs ; Ii++ ) {
    if ( Nam == DrvNamArray[Ii] ) {
      printf("%s", DrvPartsArrays[DrvNamArray[Ii]]) >> "/dev/stderr"
    }
  }
  print "]" >> "/dev/stderr"
}
BEGIN {
  Drives = "\t"
  NumDrvs = 0
  DrivesList = " "
  # For each drive, DrvNamArray[], build string of its labels, DrvPartsArrays[DrvNamArray[]]
  Cmd = "/bin/ls -l /dev/disk/by-label"
  while (( Cmd | getline LsLine ) > 0 ) {
    if ( LsLine == "total 0" ) continue
    split( LsLine, LsLineArray )
    if ( LsLineArray[10] == "->" ) {
      LsLabel = LsLineArray[9]
      LsNam = LsLineArray[11]
    } else {
      LsLabel = LsLineArray[8]
      LsNam = LsLineArray[10]
    }
    LsNam = substr(LsNam, 7)
    LsDrv = substr(LsNam,1,3)
    if ( DrivesList !~ " " LsDrv " " ) {
      NumDrvs++
      DrvNamArray[NumDrvs] = LsDrv
      DrivesList = DrivesList LsDrv " "
    }
    DrvPartsArrays[LsDrv] = DrvPartsArrays[LsDrv] LsLabel " "
  }
  close(Cmd)
  for ( Ii = 1 ; Ii <= NumDrvs ; Ii++ ) {
    sub(" $", "", DrvPartsArrays[DrvNamArray[Ii]])
    # print DrvNamArray[Ii] ": " DrvPartsArrays[DrvNamArray[Ii]] >> "/dev/stderr"
  }
}
NR <  3 {
  next
}
NR == 3 {
  PpPdrv = $4
  new_drv()
}
substr($4, 1, 3) == Nam {
  NumParts++
  next
}
{
  print_it()
  new_drv()
}
END {
  print_it()
  print PpPdrv
  printf("%s\n", Drives) > "/root/getdrives.all"
}
```

7.7.19. prtpkg_locking.dash.source

```
# #!/bin/dash
# CCIA_0.0    PP-stick base/prtpkg_locking.dash.source ....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# CRUX prtpkg lock and signal functions
#
# See prtpkg_setup which sources this file to serialize CRUX platform processing within prtpkg
# commonwealths from boot cells during their installations and distribution upgrades.  Also consult
# the Commonwealth Manual for this CCI for details.
#
# Most lock_* functions expect the following variable have been defined before
# the first lock function is invoked (from prtpkg_setup, for example):
#   TMPDIR=the absolute pathname of the preferred temporary files directory;
#         e.g., /tmp/prtpkg
#   PPDBG=the pathname defined by the external driving script for appending tracing messages;
#         e.g., $TMPDIR/setup.debug
#   PRTPKG_CELL=$hostname[$rootfs]
#   BOOTOS=CRUX version string; e.g., CRUX-3.2
#   LOCKTYPE=either 'E' or 'S' for exclusive (update) or shared
#   LOCKID=the lock identification string; e.g., 'PRTPKG'
#   LOCKDIR=the pathname of the lock queue directory sans LOCKSFX
#   LOCKSFX=the lock queue directory extension: queued, retired, or canceled
# Some lock_* functions expect or create the following variables:
#   LOCKBID=the basename (includes semi-colons) of the lock being processed
#   LOCKBIDPATH=the pathname (includes semi-colons) of the lock being processed
#   LOCKCELL=host[rootfs] of system that owns the oldest update bid (if any)
#   LOCKPID=the PID that owns this bid
#   LOCKQFA=pathname of tempfile containing newline-separated pathnames of
#           all LOCKID bids
#   LOCKQFU=pathname of tempfile containing newline-separated pathnames of
#           only update LOCKID bids
# Variable names in uppercase are global but not necessarily exported.
# Variable names in lowercase are not shared outside of a function.
#----------------------------------------------------------------------------------------------------
set +xv
####################################################################################################
```

```
# Internal locking functions not intended to be called directly by outside scripts                 #
####################################################################################################
##########################
#   L O C K _ D U M P    #
##########################
lock_dump() { # Display lock queue LOCKDIR directories requested via $1
    # Uses    LOCKID LOCKDIR
    echo                                      "Entered lock_dump LOCKDIR<$LOCKDIR> \$1<$1>" >>$PPDBG
    case ".$1" in
       '.QRC' ) sq='queued'  ; sr='retired' ; sc='canceled' ;;
       '.QR'  ) sq='queued'  ; sr='retired' ; sc=''         ;;
       '.QC'  ) sq='queued'  ; sr=''        ; sc='canceled' ;;
       '.RC'  ) sq=''        ; sr='retired' ; sc='canceled' ;;
       '.Q'   ) sq='queued'  ; sr=''        ; sc=''         ;;
       '.R'   ) sq=''        ; sr='retired' ; sc=''         ;;
       '.C'   ) sq=''        ; sr=''        ; sc='canceled' ;;
       *      ) lock_fail "\$1<$1> is an invalid suffixes specification" ;;
    esac
    # Snapshot the requested directories
    test $sq \
      && { qfdq="$TMPDIR/lock_dump-q.$$.`/bin/date -u '+%Y%m%d-%T.%NUT'`" # temp file name
           /bin/ls -1 $LOCKDIR.$sq/* >$qfdq 2>/dev/null # Snapshot the complete directory
         }
    test $sr \
      && { qfdr="$TMPDIR/lock_dump-r.$$.`/bin/date -u '+%Y%m%d-%T.%NUT'`" # temp file name
           /bin/ls -1 $LOCKDIR.$sr/* >$qfdr 2>/dev/null # Snapshot the complete directory
         }
    test $sc \
      && { qfdc="$TMPDIR/lock_dump-c.$$.`/bin/date -u '+%Y%m%d-%T.%NUT'`" # temp file name
           /bin/ls -1 $LOCKDIR.$sc/* >$qfdc 2>/dev/null # Snapshot the complete directory
         }
    # Output each requested snapshot to $PPDBG
    test $sq && lock_dump_dir "$qfdq" '/usr/bin/head -33'
    test $sr && lock_dump_dir "$qfdr" '/usr/bin/tail -7'
    test $sc && lock_dump_dir "$qfdc" '/usr/bin/tail -7'
}
#################################
#   L O C K _ D U M P _ D I R   #
#################################
lock_dump_dir() { # Display lock queue snapshot requested via $1 and filtered using $2
    echo                                            "Entered lock_dump_dir \$1<$1> \$2<$2>" >>$PPDBG
    test ".$1" = . \
      && lock_fail "\$1 is not specified"
    test -s "$1" \
      && { /bin/ls                                                            -lh "$1" >>$PPDBG 2>&1
           /bin/cat "$1" 2>>$PPDBG \
           | while read LOCKBIDPATH
             do    /bin/ls                                              -lh "$LOCKBIDPATH" >>$PPDBG 2>&1
                   /bin/cat                                            "$LOCKBIDPATH" | $2 >>$PPDBG 2>&1
             done
         }
}
##########################
#   L O C K _ F A I L    #
##########################
lock_fail() { # Exit the process due to problem stated in $1
    # Abandon ship!  Let troubleshooting and clever programming under pressure ensue
    # Expects $1=a one-line string that explains the problem
    # Uses    LOCKID LOCKDIR
    echo                              "Entered lock_fail LOCKID<$LOCKID> LOCKDIR<$LOCKDIR>" >>$PPDBG
    echo "$1"                                                                               >>$PPDBG
    echo "$1"
    test ".$LOCKDIR" != . \
      && lock_dump 'QRC' \
      || echo "lock_dump cannot be invoked with a null LOCKDIR"                             >>$PPDBG
    echo "This version of prtpkg is unprepared to deal with this situation--aborting!"      >>$PPDBG
    echo "This version of prtpkg is unprepared to deal with this situation--aborting!"
    exit 32
}
#################################
#   L O C K _ V A L I D A T E   #
```

```
#################################
lock_validate() { # Ensure LOCKTYPE, LOCKID, and LOCKDIR are legitimate or perish
    # Uses    LOCKTYPE LOCKID LOCKDIR
    echo      "Entered lock_validate LOCKTYPE<$LOCKTYPE> LOCKID<$LOCKID> LOCKDIR<$LOCKDIR>" >>$PPDBG
    test    ".$LOCKTYPE" != '.E' \
         -a ".$LOCKTYPE" != '.S' \
      && lock_fail "LOCKTYPE is neither 'E' nor 'S'"
    case $LOCKID in
      'PRTPKG' )
         ;;
      * )
         lock_fail "LOCKID $LOCKID is unrecognized"
    esac
    test ".$LOCKDIR" = . \
      && lock_fail "Provided LOCKDIR is a null string"
    /bin/ls -L $LOCKDIR.queued >/dev/null 2>&1 \
      || { /bin/mkdir -p $LOCKDIR.queued                                   >>$PPDBG 2>&1 \
             && { echo                                    "Created $LOCKDIR.queued" >>$PPDBG
                  /bin/ls                                 -lhd $LOCKDIR.queued >>$PPDBG 2>&1
                } \
             || lock_fail "mkdir for $LOCKDIR.queued failed code $?"
         }
    /bin/ls -L $LOCKDIR.retired >/dev/null 2>&1 \
      || { /bin/mkdir -p $LOCKDIR.retired                                  >>$PPDBG 2>&1 \
             && { echo                                   "Created $LOCKDIR.retired" >>$PPDBG
                  /bin/ls                                -lhd $LOCKDIR.retired >>$PPDBG 2>&1
                } \
             || lock_fail "mkdir for $LOCKDIR.retired failed code $?"
         }
    /bin/ls -L $LOCKDIR.canceled >/dev/null 2>&1 \
      || { /bin/mkdir -p $LOCKDIR.canceled                                 >>$PPDBG 2>&1 \
             && { echo                                  "Created $LOCKDIR.canceled" >>$PPDBG
                  /bin/ls                               -lhd $LOCKDIR.canceled >>$PPDBG 2>&1
                } \
             || lock_fail "mkdir for $LOCKDIR.canceled failed code $?"
         }
    echo                                                  "Leaving lock_validate nominally" >>$PPDBG
}
#########################################
#   L O C K _ T E S T _ G E T _ P I D   #
#########################################
lock_test_get_pid() { # Ensure bid file is a single line and extract its PID
    # Returns LOCKPID
    # Uses    LOCKID LOCKDIR LOCKSFX LOCKBID
    echo    "Entered lock_test_get_pid LOCKID<$LOCKID> LOCKDIR<$LOCKDIR> LOCKSFX<$LOCKSFX>" >>$PPDBG
    echo                                  "                          LOCKBID<$LOCKBID>" >>$PPDBG
    test    ".$LOCKSFX" = '.queued' \
         -o ".$LOCKSFX" = '.retired' \
         -o ".$LOCKSFX" = '.canceled' \
      || lock_fail "LOCKSFX<$LOCKSFX> contains an invalid value"
  # lock_dump 'Q'                                                                           >>$PPDBG
    test -f "$LOCKDIR.$LOCKSFX/$LOCKBID" \
      || lock_fail "lock bid file <$LOCKBID> in $LOCKID $LOCKDIR.$LOCKSFX does not exist"
    test -s "$LOCKDIR.$LOCKSFX/$LOCKBID" \
      || lock_fail "lock bid file <$LOCKBID> in $LOCKID $LOCKDIR.$LOCKSFX is empty"
    test `/bin/cat "$LOCKDIR.$LOCKSFX/$LOCKBID" \
          | /usr/bin/wc -l \
         ` -ne 1 \
      && lock_fail "LOCKBID<$LOCKBID> in $LOCKID <$LOCKDIR.$LOCKSFX> contains multiple lines"
    LOCKPID=`/bin/cat "$LOCKDIR.$LOCKSFX/$LOCKBID" \
             | /usr/bin/gawk 'NR == 1{print $2}' \
            ` # Get the pid that owns this bid
    echo                                 "Leaving lock_test_get_pid with LOCKPID<$LOCKPID>" >>$PPDBG
}
#################################
#   L O C K _ M A K E _ B I D   #
#################################
lock_make_bid() { # Make a bid for the LOCKID update lock
    # Returns LOCKBID
    # Uses    LOCKTYPE LOCKID LOCKDIR
    echo      "Entered lock_make_bid LOCKTYPE<$LOCKTYPE> LOCKID<$LOCKID> LOCKDIR<$LOCKDIR>" >>$PPDBG
    ts=`/bin/date -u '+%Y%m%d-%T.%NUT'` # Get the current timestamp for the filename and pid record
    LOCKBID="$ts;$LOCKTYPE;$PRTPKG_CELL;$BOOTOS;setup" # Define new filename for bid
    echo $ts $$ >"$LOCKDIR.queued/$LOCKBID" # Create the bid request
    lock_dump 'Q'                                                                           >>$PPDBG
    echo                                     "Leaving lock_make_bid with LOCKBID<$LOCKBID>" >>$PPDBG
}
#################################################
#   L O C K _ G E T _ U P D A T E _ Q U E U E   #
#################################################
lock_get_update_queue() { # Slurp update filenames in LOCKID lock queue into temp file
    # Returns LOCKCELL LOCKQFU
    # Uses    LOCKID LOCKDIR
    echo                  "Entered lock_get_update_queue LOCKID<$LOCKID> LOCKDIR<$LOCKDIR>" >>$PPDBG
    LOCKQFU="$TMPDIR/lock_get_update_queue.$$.`/bin/date -u '+%Y%m%d-%T.%NUT'`"
    /bin/ls -1 $LOCKDIR.queued/*UT\;E\;* >$LOCKQFU 2>/dev/null # Get the update queue
    /bin/ls                                                               -lh $LOCKQFU >>$PPDBG 2>&1
    /bin/cat                                                                  $LOCKQFU >>$PPDBG 2>&1
    LOCKCELL=`/usr/bin/gawk -F\; 'NR == 1{print $3}' $LOCKQFU` # Get cell of oldest if any
    echo                                                              "LOCKCELL<$LOCKCELL>" >>$PPDBG
    echo              "Leaving lock_get_update_queue LOCKQFU<$LOCKQFU> LOCKCELL<$LOCKCELL>" >>$PPDBG
}
###################################
#   L O C K _ C H E C K _ B I D   #
###################################
lock_check_bid() { # Make sure this process holds the LOCKID update lock
    # For now, this either returns (indiating this process holds the LOCKID update lock)
    # or terminates the process (soon it will learn how to wait).
    # Expects $1='old'|'new' identifying post-bid or pre-bid processing is afoot, respectively
    # Returns LOCKQFA
    # Uses    LOCKID LOCKDIR LOCKQFU
    echo                 "Entered lock_check_bid LOCKID<$LOCKID> LOCKDIR<$LOCKDIR> \$1<$1>" >>$PPDBG
    echo                                         "                       LOCKBID<$LOCKBID>" >>$PPDBG
    test ".$1" != '.old' -a ".$1" != '.new' \
      && { echo "locl_check_bid called with invalid argument <$1>--aborting!"
           exit 32
         }
    prel=`/usr/bin/gawk -F\; 'NR == 1{print $4}' $LOCKQFU` # Get release of our bid
    test ".$prel" != ".$BOOTOS" \
      && { test $1 = 'old' \
             && { # Our earlier bid is for a different release than what we want
                  lock_fail "The oldest update lock bid is this cell's bid to get $prel (?)"
                } \
             || { # The new bid has a different prel?  What could have bid against us?
                  lock_fail "An update lock bid for $prel just beat us in the race to lock (?)"
                } \
         }
    # The oldest update bid is the right system and the right release.
    # Are there any shared grants ahead of it?
    LOCKQFA="$TMPDIR/lock_check_bid.$$.`/bin/date -u '+%Y%m%d-%T.%NUT'`"
    /bin/ls -1 $LOCKDIR.queued/* >$LOCKQFA 2>/dev/null # Get the complete queue
    /bin/ls                                                               -lh $LOCKQFA >>$PPDBG 2>&1
    /bin/cat                                                                       $LOCKQFA >>$PPDBG
    ptyp=`/usr/bin/gawk -F\; 'NR == 1{print $2}' $LOCKQFA` # Get type of oldest bid
    if   test ".$ptyp" != . -a "$ptyp" = s
    then # We are not first in line
         lock_fail "The $LOCKID update bid is waiting on one or more shared bids"
    fi
    # We hold the LOCKID update lock
}
###########################################
#   L O C K _ P R T P K G _ U P D A T E   #
###########################################
lock_prtpkg_update() { # Obtain the PRTPKG update lock to the commonwealth-wide resources
    # Uses    LOCKTYPE LOCKDIR LOCKBID LOCKQFU LOCKCELL
    echo                                     "Entered lock_prtpkg_update LOCKDIR<$LOCKDIR>" >>$PPDBG
    lock_get_update_queue # returns LOCKQFU and LOCKCELL of oldest update bid, if any
    if   test ".$LOCKCELL" != .
    then # There is at least one update bid in queue
         if   test  "$LOCKCELL" != $PRTPKG_CELL
         then # We are not alone--there are other systems (must be an update to a
              # commonwealth containing only this prtpkg platform) - TODO SOON!
              lock_fail "The PRTPKG update lock has been requested by another cell <$LOCKCELL>"
         else # This system bid the oldest LOCKID update
              echo                "PRTPKG update is already bid by $LOCKCELL (this system)" >>$PPDBG
              lock_check_bid old # We have the lock if this returns
         fi
    else # There are currently no PRTPKG update bids in the queue, so bid for
         # exclusive PRTPKG privilege in this commonwealth
         lock_make_bid # Append a bid file to the lock queue
         lock_get_update_queue # returns LOCKQFU and LOCKCELL of oldest update bid
         if   test "$LOCKCELL" != $PRTPKG_CELL # There must be at least one this time
         then # Our PRTPKG update bid is not the oldest, we have to wait - TODO SOON!
              lock_fail "The bid for the update lock is waiting on updates."
         else # This system bid the oldest PRTPKG update
              lock_check_bid new # We have the lock if this returns
         fi
    fi
    # We hold the PRTPKG update lock
}
###########################################
#   L O C K _ P R T P K G _ S H A R E D   #
###########################################
lock_prtpkg_shared() { # Obtain the PRTPKG shared lock to the commonwealth-wide resources
    # Uses    LOCKTYPE LOCKDIR LOCKBID LOCKQFA LOCKCELL
    # TODO - replace this stub
    lock_fail "The lock_PRTPKG_shared logic is not yet available"
}
#########################################
#   L O C K _ A W A K E N _ Q U E U E   #
#########################################
lock_awaken_queue() { # Inform LOCKDIR grantees of possibly newly granted requests
    #   Returns LOCKQFA
    #   Uses    LOCKID LOCKDIR
    echo                      "Entered lock_awaken_queue LOCKID<$LOCKID> LOCKDIR<$LOCKDIR>" >>$PPDBG
    LOCKQFA="$TMPDIR/lock_awaken_queue.$$.`/bin/date -u '+%Y%m%d-%T.%NUT'`"
    /bin/ls -1 $LOCKDIR.queued/* >$LOCKQFA 2>/dev/null # Get the complete queue
    /bin/ls                                                               -lh $LOCKQFA >>$PPDBG 2>&1
    /bin/cat                                                                       $LOCKQFA >>$PPDBG
    granted='true' # The first bid is always granted
    /bin/cat $LOCKQFA 2>/dev/null \
    | while read LOCKBIDPATH
    do  ntyp=`echo "$LOCKBIDPATH" \
              | /usr/bin/gawk -F \; 'NR == 1{print $2}' \
             ` # Get lock type
        nrel=`echo "$LOCKBIDPATH" \
              | /usr/bin/gawk -F \; 'NR == 1{print $4}' \
             ` # Get release
        LOCKPID=`/bin/cat "$LOCKBIDPATH" \
               | /usr/bin/gawk 'NR == 1{print $2}' \
             ` # Get shoulder-tap pid
        test $granted \
          && { test ".$LOCKPID" != ".$$" \
                 && lock_shoulder_tap $LOCKPID
               test ".$ntyp" = '.U' \
                 && granted='' # Subsequent bids are not yet granted
             }
    done
}
#########################################
#   L O C K _ S H O U L D E R _ T A P   #
#########################################
lock_shoulder_tap() { # Awaken grantee of lock if necessary
    # Expects $1=PID that owns the bid being activated
    echo                                             "Entered lock_shoulder_tap for PID $1" >>$PPDBG
    test ".$1" = '.' \
      && { echo "lock_shoulder_tap called with invalid PID argument <$1>--aborting!"
           exit 32
         }
    spid=`/bin/ps hc -o pid,comm --ppid $1 \
          | /usr/bin/gawk '$2 == "sleep"{print $1}' \
         `
    if   test ".$1" != . -a $1 != $$
    then /bin/kill -s USR1 $1 2>/dev/null         # Tell bidder to get to work
         echo                                                  "Sent USR1 signal to PID $1" >>$PPDBG
         if   test ".$spid" != .
         then /bin/kill -s TERM $spid 2>/dev/null # Kill sleeper to awaken bidder
              echo                                          "Sent TERM signal to PID $spid" >>$PPDBG
         fi
    fi
}
##############################
#   L O C K _ R E T I R E   #
##############################
lock_retire() { # Moves unlocked bid to the retired directory or perish
    # Uses    LOCKID LOCKDIR LOCKBID
    echo          "Entered lock_retire LOCKID<$LOCKID> LOCKDIR<$LOCKDIR> LOCKBID<$LOCKBID>" >>$PPDBG
    /bin/mv "$LOCKDIR.queued/$LOCKBID" $LOCKDIR.retired \
      || lock_fail "mv of $LOCKDIR.queued/$LOCKBID to .retired failed - code $?"
    lock_dump 'R'                                                                           >>$PPDBG
}
####################################################################################################
# External locking functions intended to be called directly by outside scripts                     #
####################################################################################################
##############################
#   L O C K _ O B T A I N   #
##############################
lock_obtain() { # Obtain the LOCKID LOCKTYPE lock in LOCKDIR
    # Uses    LOCKID LOCKTYPE LOCKDIR
    echo        "Entered lock_obtain LOCKID<$LOCKID> LOCKTYPE<$LOCKTYPE> LOCKDIR<$LOCKDIR>" >>$PPDBG
    lock_validate
    case $LOCKID in
      'PRTPKG' )
         test $LOCKTYPE = 'E' \
           && lock_prtpkg_update \
           || lock_prtpkg_shared
         ;;
       * )
         lock_fail "Unsupported LOCKID ($LOCKID) requested in lock_obtain"
    esac
}
##############################
#   L O C K _ A S S U M E   #
##############################
lock_assume() { # Obtain the LOCKID floating update lock in LOCKDIR
    # Returns LOCKTYPE
    # Uses    LOCKID LOCKDIR
    echo                            "Entered lock_assume LOCKID<$LOCKID> LOCKDIR<$LOCKDIR>" >>$PPDBG
    LOCKTYPE='E' # Exclusive (update)
    lock_validate
    # TODO - complete this stub
    lock_fail "The lock_assume logic is not yet available"
}
##############################
#   L O C K _ F R E E U P   #
##############################
lock_freeup() { # Changes the held LOCKID update lock into a shared lock
    # Also proesses any queued shared bids for the same distribution release behind ours
    # Uses    LOCKID LOCKDIR LOCKBID
    echo          "Entered lock_freeup LOCKID<$LOCKID> LOCKDIR<$LOCKDIR> LOCKBID<$LOCKBID>" >>$PPDBG
    lock_validate
    lock_dump 'Q'                                                                           >>$PPDBG
    /bin/ls                                                 -1 $LOCKDIR.queued/*\;E\;* >>$PPDBG 2>&1
    LOCKPPE=`echo $LOCKDIR.queued/*\;E\;* \
             | /usr/bin/head -1 \
            ` # Get the current update lock pathname
    echo                                                                "LOCKPPE<$LOCKPPE>" >>$PPDBG
    LOCKPPS=`echo "$LOCKPPU" \
             | /bin/sed -e 's/;E;/;S;/' \
            ` # Create the shared replacement pathname
    echo                                                                "LOCKPPS<$LOCKPPS>" >>$PPDBG
    echo                                               "Attempting cp -p $LOCKPPU $LOCKPPS" >>$PPDBG
    /bin/cp -p "$LOCKPPE" "$LOCKPPS" \
      && { LOCKBID=`/usr/bin/basename "$LOCKPPE"` # Define the variable lock_retire expects
```

```
           lock_retire # retire the original update bid, leaving the shared bid in its place
         } \
      || lock_fail "cp -p $LOCKPPE $LOCKPPS failed - code $?"
    lock_awaken_queue
    echo                                                              'Leaving lock_freeup' >>$PPDBG
}
#############################
#   L O C K _ G I V E U P   #
#############################
lock_giveup() { # Give up the held LOCKID update lock (let it float)
    # This process cannot unlock the LOCKID update lock because the prtpkg
    # universe will be harmed until something this process cannot perform has
    # been performed by a different process (probably an interactive shell).
    # Thus this process "gives up" its ownership of the update lock before it
    # terminates leaving the lock granted to the cell but "floating" without
    # any owning process.  Subsequent processes can bid to assume lock ownership
    # (real soon now) via the coming assume_prtpkg function.
    # Uses    LOCKID LOCKDIR LOCKBID LOCKPID
    echo          "Entered lock_giveup LOCKID<$LOCKID> LOCKDIR<$LOCKDIR> LOCKBID<$LOCKBID>" >>$PPDBG
    lock_validate
    lock_dump 'Q'                                                                           >>$PPDBG
    LOCKSFX='queued'
    lock_test_get_pid
    test .$LOCKPID = .$$ \
      && { ts=`/bin/date -u '+%Y%m%d-%T.%NUT'`    # get the current timestamp for time of float
           echo $ts 0 >"$LOCKDIR.queued/$LOCKBID" # Rewrite the bid request with pid 0 (no pid)
         }
    lock_dump 'Q'                                                                           >>$PPDBG
    echo                                                              'Leaving lock_giveup' >>$PPDBG
}
#############################
#   L O C K _ U N L O C K   #
#############################
lock_unlock() { # Removes the oldest lock in the LOCKID lock queue
    # Only iff it is owned by this proess.  It also proesses any other bids behind
    # this bid that might now be granted due to this unlocking.
    # Uses    LOCKID LOCKDIR LOCKBID LOCKPID LOCKMVRC
    echo          "Entered lock_unlock LOCKID<$LOCKID> LOCKDIR<$LOCKDIR> LOCKPID<$LOCKPID>" >>$PPDBG
    echo                                                                "LOCKBID<$LOCKBID>" >>$PPDBG
    lock_validate
    LOCKSFX='queued'
    lock_test_get_pid
    if   test .$LOCKPID = .$$
    then lock_retire
    else lock_fail "PID $$ cannot unlock bid owned by PID $LOCKPID ($LOCKDIR.queued/$LOCKBID)"
    fi
    lock_dump 'QR'                                                                          >>$PPDBG
    lock_awaken_queue
    echo                                                              'Leaving lock_unlock' >>$PPDBG
}
#############################
#   L O C K _ C A N C E L   #
#############################
lock_cancel() { # For this process, remove any LOCKDIR pending bids and unlock any granted bids
    # Uses    LOCKID LOCKDIR LOCKPID
    echo                   "Entered lock_cancel LOCKID<$LOCKID> LOCKDIR<$LOCKDIR> PID <$$>" >>$PPDBG
    lock_validate
    LOCKQFC="$TMPDIR/lock_cancel.$LOCKID.$$.`/bin/date -u '+%Y%m%d-%T.%NUT'`"
    /bin/ls -1 $LOCKDIR.queued/* >$LOCKQFC 2>/dev/null # Snapshot the complete queue
    /bin/ls                                                               -lh $LOCKQFC >>$PPDBG 2>&1
    /bin/cat                                                                       $LOCKQFC >>$PPDBG
    granted='true' # The first bid is always granted
    /bin/cat $LOCKQFC 2>/dev/null \
    | while read LOCKBIDPATH
      do    ntyp=`echo "$LOCKBIDPATH" \
                  | /usr/bin/gawk -F \; 'NR == 1{print $2}' \
                 ` # Get lock type
            LOCKPID=`/bin/cat "$LOCKBIDPATH" \
                     | /usr/bin/gawk 'NR == 1{print $2}' \
                    ` # Get shoulder-tap pid
            echo                                           "Loop LOCKBIDPATH<$LOCKBIDPATH>" >>$PPDBG
            echo                     "     ntyp<$ntyp> LOCKPID<$LOCKPID> granted<$granted>" >>$PPDBG
            test $granted \
              && test ".$LOCKPID" = .0 \
                   && echo                                        "Ignoring floated lock" >>$PPDBG \
                   || { test ".$LOCKPID" = .$$ \
                          && { LOCKBID=`/usr/bin/basename "$LOCKBIDPATH"`
                               lock_unlock
                             } \
                      } \
              || { test ".$LOCKPID" = .$$ \
                     && { # Cancel this ungranted bid
                          echo                   "Attempting mv to .cancelled directory..." >>$PPDBG
                          /bin/mv "$LOCKDIR.queued/$LOCKBID" "$LOCKDIR.canceled" \
                            && { ts=`/bin/date -u '+%Y%m%d-%T.%NUT'` # get time of cancellation
                                 echo                   "$LOCKDIR $LOCKBID canceled at $ts" >>$PPDBG
                                 lock_dump 'QC'                                             >>$PPDBG
                               } \
                            || lock_fail "Cancellation of $LOCKDIR $LOCKBID failed (mv code $?)"
                        } \
                 }
            test ".$ntyp" = ".U" \
              && granted='' # Subsequent bids are not yet granted
        done
}
####################################################################################################
# Internal signal functions not intended to be called directly by outside scripts                  #
####################################################################################################
#######################
#   S I G N A L _ 0   #
#######################
signal_0() { # EXIT signal handling routine
    # Normal termination should occur after cleaning up all locks
    # Uses    PRTPKG_LOCKDIR
    echo      "`date -u '+%Y%m%d-%T.%NUT'` -- Signal  EXIT( 0) has been trapped in pid<$$>" >>$PPDBG
    /bin/ps                                                         hc -F -w --ppid $$ >>$PPDBG 2>&1
    signal='EXIT'
    for i in 0 1 2 3 10 12 15 # Inactivate all previously set traps
    do  trap - $i
    done
    for LOCKID in 'PRTPKG' # TODO - Add all lock queues in reverse order of locking
    do  case $LOCKID in
          'PRTPKG' )
             LOCKDIR=$LOCKDIR_PRTPKG
             lock_cancel # cancel or unlock any bids owned by this PID (floating locks are unowned)
             lock_dump 'QRC'                                                                >>$PPDBG
             ;;
        esac
    done
    echo      "`date -u '+%Y%m%d-%T.%NUT'` -- Leaving EXIT( 0) signal handling environment" >>$PPDBG
}
#######################
#   S I G N A L _ 1   #
#######################
signal_1() { # HUP signal handling routine
    echo      "`date -u '+%Y%m%d-%T.%NUT'` -- Signal  HUP ( 1) has been trapped in pid<$$>" >>$PPDBG
    /bin/ps                                                                     $id $$ >>$PPDBG 2>&1
    signal='HUP'
    echo      "`date -u '+%Y%m%d-%T.%NUT'` -- Leaving HUP ( 1) signal handling environment" >>$PPDBG
}
#######################
#   S I G N A L _ 2   #
#######################
signal_2() { # INT signal handling routine
    echo      "`date -u '+%Y%m%d-%T.%NUT'` -- Signal  INT ( 2) has been trapped in pid<$$>" >>$PPDBG
    /bin/ps                                                         hc -F -w --ppid $$ >>$PPDBG 2>&1
    signal='INT'
    echo      "`date -u '+%Y%m%d-%T.%NUT'` -- Leaving INT ( 2) signal handling environment" >>$PPDBG
}
#######################
#   S I G N A L _ 3   #
#######################
```

```
signal_3() { # QUIT signal handling routine
    echo      "`date -u '+%Y%m%d-%T.%NUT'` -- Signal  QUIT( 3) has been trapped in pid<$$>" >>$PPDBG
    /bin/ps                                                        hc -F -w --ppid $$ >>$PPDBG 2>&1
    signal='QUIT'
    echo      "`date -u '+%Y%m%d-%T.%NUT'` -- Leaving QUIT( 3) signal handling environment" >>$PPDBG
}
#########################
#   S I G N A L _ 1 0   #
#########################
signal_10() { # USR1 signal handling routine
    echo      "`date -u '+%Y%m%d-%T.%NUT'` -- Signal  USR1(10) has been trapped in pid<$$>" >>$PPDBG
    /bin/ps                                                        hc -F -w --ppid $$ >>$PPDBG 2>&1
    signal='USR1'
    echo      "`date -u '+%Y%m%d-%T.%NUT'` -- Leaving USR1(10) signal handling environment" >>$PPDBG
}
#########################
#   S I G N A L _ 1 2   #
#########################
signal_12() { # USR2 signal handling routine
    echo      "`date -u '+%Y%m%d-%T.%NUT'` -- Signal  USR2(12) has been trapped in pid<$$>" >>$PPDBG
    /bin/ps                                                        hc -F -w --ppid $$ >>$PPDBG 2>&1
    signal='USR2'
    echo      "`date -u '+%Y%m%d-%T.%NUT'` -- Leaving USR2(12) signal handling environment" >>$PPDBG
}
#########################
#   S I G N A L _ 1 5   #
#########################
signal_15() { # TERM signal handling routine
    echo      "`date -u '+%Y%m%d-%T.%NUT'` -- Signal  TERM(15) has been trapped in pid<$$>" >>$PPDBG
    /bin/ps                                                        hc -F -w --ppid $$ >>$PPDBG 2>&1
    signal='TERM'
    echo      "`date -u '+%Y%m%d-%T.%NUT'` -- Leaving TERM(15) signal handling environment" >>$PPDBG
}
####################################################################################################
# External signal functions intended to be called directly by outside scripts                     #
####################################################################################################
#######################################
#   S I G N A L _ S E T _ T R A P S   #
#######################################
signal_set_traps() { # activate all traps to disentangle from prtpkg locking
echo                                                         'Entered signal_set_traps' >>$PPDBG
for i in 0 1 2 3 5 10 12 15
do  trap "signal_$i" $i
done
}
```

7.7.20. prtpkg_mkfs

```
#!/bin/dash
# CCIA_0.0    PP-stick base/prtpkg_mkfs ....:....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# Create an ext4 filesystem with
#   $1 (required) specifying the label of the partition to be formatted
#   $2 (required) specifying the number of the partition to be formatted
#   $3 (optional) specifying:
#      'r' if the read-only badblocks test is desired
#      'w' if the read-write badblocks test is desired
#      ''  if no badblocks test is desired (the default)
#----------------------------------------------------------------------------------------------------
set +xv
if   test `whoami` != root
then echo 'Only root can use the script.'
     exit 2
fi
```

```
test    $# -lt 2 \
     -o $# -gt 3 \
  && { echo "Syntax: $0  partition_label  partition_number [r|w]"
       echo "  e.g., $0  XB                3"
       exit 1
     }
test ".$PPHOST" = . \
  && { echo 'The PPHOST environment variable has not been pre=defined--aborting!'
       exit 32
     }
test ".$PPPDRV" = . \
  && { echo 'The PPPDRV environment variable has not been pre=defined--aborting!'
       exit 32
     }
test    ".$PPBORU" != '.BIOS' \
     -a ".$PPBORU" != '.UEFI' \
  && { echo "PPBORU <$PPBORU> is not valid--aborting!"
       exit 32
     }
L="$1"             # e.g., XB
P="$2"             # e.g., 3
if   test ".$3" = '.r' ; then CC='-c '
elif test ".$3" = '.w' ; then CC='-c -c '
elif test ".$3" = '.'  ; then CC=''
else echo "The third argument <$3> must be 'r', 'w', or omitted--aborting!"
     exit 32
fi
D="/dev/$PPPDRV$P" # e.g., /dev/sda1
test -b $D \
  || { echo "Did not find block device $D--aborting!"
       exit 32
     }
# Extract the partition label for the partition number per parted:
PL=`/usr/sbin/parted /dev/$PPPDRV unit MiB print \
    | /usr/bin/gawk '$1 == "'$P'" {if ( $5 ~ "^ext" ) print $6 ; else print $5}' \
   `
test ".$PL" != ".$L" \
  && { echo "The partition label for partition $P <$PL> is not <$L>--aborting!"
       exit 32
     }
set -xv
# Ensure the partition is not mounted:
test `/bin/df \
      | /usr/bin/grep "^$D " \
      | /usr/bin/wc -l \
     ` -gt 0 \
  && umount $D
# If necessary, create the partition's canonical mount point:
test ! -d /_/l/$L \
  && /bin/mkdir -p /_/l/$L
# Format the filesystem, set the tuneable attributes, and mount it
# (this is a long string of escaped newlines down to the next comment
# to simplify editing attributes desired):
/sbin/mke2fs -v -t ext4 $CC -r 1 -L $L -T default\
 \
-O\
 \
dir_index\
,\
extent\
,\
filetype\
,\
flex_bg\
,\
has_journal\
,\
^journal_dev\
,\
large_file\
,\
^resize_inode\
,\
sparse_super\
,\
uninit_bg\
 \
-E \
lazy_itable_init=1\
 \
-b 4096 -I 256 -i 16384 -m 1 -J size=8 -G 1024 $D\
 \
&&\
 \
/sbin/tune2fs -c 256 -C 0 -e remount-ro -E hash_alg=tea -i 6m -r 1024 -u 0\
 \
-o\
 \
^acl\
,\
^block_validity\
,\
^discard\
,\
^debug\
,\
^bsdgroups\
,\
journal_data\
,\
journal_data_ordered\
,\
^journal_data_writeback\
,\
nobarrier\
,\
^nodelalloc\
,\
uid16\
,\
user_xattr\
 \
-O\
 \
dir_index\
,\
filetype\
,\
flex_bg\
,\
has_journal\
,\
large_file\
,\
^resize_inode\
,\
sparse_super\
,\
uninit_bg\
 \
LABEL=$L\
 \
&&\
 \
/bin/mount -L $L -o noatime /_/l/$L
# If the mount was nominal, create the labelfile, report the
# mounted state, and unmount it again:
if   test `/usr/bin/grep " /_/l/$L" /proc/mounts \
          | /usr/bin/wc -l \
         ` -eq 1
then test ! -f "/_/l/$L/[$L]" \
       && echo $PPHOST                   > /_/l/$L/\[$L\]
       && /bin/touch -r /_/l/$L/lost\+found /_/l/$L/\[$L\]
     /bin/df    -alHT \
     | /usr/bin/grep $D
     /bin/mount -lnf \
     | /usr/bin/grep $D
     /bin/ls -lah /_/l/$L
     /bin/umount  /_/l/$L
fi
# Report filesystem state per tune2fs:
/sbin/tune2fs -l LABEL=$L
```

7.7.21. prtpkg_mount

```
#!/bin/dash
# CCIA_0.0    PP-stick base/prtpkg_mount ...:....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# The script is intended to be run in the CRUX ISO installation/upgrade system booted with a root
# filesystem (RAMdisk if booting the CRUX release ISO) populated from the ISO's rootfs.tar.xz file.
# The environment is established primarily to install or upgrade CRUX on a target root filesystem
# set which, for prtpkg commonwealths, likely involves exported NFS filesystems.  Also, it can be
# used as a rescue system to deal with situations when the target system must not be operational for
# the non-primary maintenance procedure or cannot be booted nominally.
#
# This script is a subordinate function of prtpkg_pdrive and prtpkg_presetup that determines the
# target root filesystem(s) to be mounted for processing by the CRUX canonical setup script or the
# prtpkg_setup script.  It makes an educated guess at the desired target but asks when it it not
# clearcut.  The variables PPTARD, PPTARL, and PPTARM in the /_/l/PP/h/$PPHOST/PPvars file are
# added or edited as needed.
#----------------------------------------------------------------------------------------------------
set -e +xv
# Ensure this was invoked by root:
test `/usr/bin/whoami` != 'root' \
  && { echo 'You must be root to run this script--aborting!'
       exit 8
     }
# Ensure prtpkg_source was previously and successfully sourced:
test    ! \( $PPHOST \) \
     -o \(    ".$PPTYPE" != '.ISO' \
           -a ".$PPTYPE" != '.CME' \
        \) \
  && { echo "The PPHOST<$PPHOST> and/or PPTYPE<$PPTYPE> variables are improper--aborting!"
       echo "The prtpkg_source file must be successfully sourced before running this script."
       exit 32
     }
test    ".$PPMERL" = '.' \
     -o ".$PPPCRL" = '.' \
     -o ".$PPMEMP" = '.' \
     -o ".$PPPCMP" = '.' \
     -o ! -f /_/l/PP/h/$PPHOST/PPvars \
  && { echo "The /_/l/PP/h/$PPHOST/PPvars file does not seem to have been sourced--aborting!"
       echo "This script must be invoked by prtpkg_pdrive or prtpkg_presetup."
       exit 32
     }
if   test    $PPTYPE = 'ISO'
then
     if   test \( $PPMERL \)
     then echo "Choose 'c' to mount the CME environment or 'p' to mount the PCE"
          read -p "environment as the target for prtpkg install or upgrade: " ASK
          while test    ".$ASK" != '.c' \
                     -a ".$ASK" != '.p'
          do    read -p "Invalid response <$ASK>--enter 'c' or 'p':  " ASK
          done
          echo "The canonical ISO environment is ready to run prtpkg_setup"
          test $ASK = 'p' \
            && echo "on the production CRUX environment (PCE)." \
            || echo "on the permanent CRUX maintenance environment (CME)."
     else echo "The canonical ISO environment is ready to run prtpkg_setup"
          echo "on the production CRUX environment."
          ASK='p'
     fi
else echo "The permanent maintenance environment is ready to run prtpkg_setup"
     echo "on the production CRUX environment."
     ASK='p'
fi
test $ASK = 'm' \
  && { export PPTARL="$PPMERL"
       export PPTARM="$PPMEMP"
     } \
  || { export PPTARL="$PPPCRL"
       export PPTARM="$PPPCMP"
     }
set -e
test ! -f $PPTARM/\[$PPTARL\] \
  && /bin/mount -L $PPTARL $PPTARM
if   test \( $PPBOTL \)
then test ! -f $PPTARM/boot/\[$PPBOTL\] \
       && /bin/mount -L $PPBOTL $PPTARM/boot
     if   test \( $PPESPL \)
     then test ! -f $PPTARM/boot/efi/\[$PPESPL\] \
            && /bin/mount -L $PPESPL $PPTARM/boot/efi
     fi
else if   test \( $PPESPL \)
     then test ! -f $PPTARM/boot/\[$PPESPL\] \
            && /bin/mount -L $PPESPL $PPTARM/boot
     fi
fi
export PPTARD=`/usr/bin/gawk '$2 == "'$PPTARM'" {print substr($1, 6)}' /proc/mounts`
/bin/cat <<'EOGAWK' > /tmp/prtpkg_mount.awk
function prt_it() {
  print "export " NAM "='" VAL "'" ;
# print "prt_it<export " NAM "='" VAL "'" >> "/dev/stderr" ;
}
BEGIN {
  existed = "F" ;
}
{
# print "Read<" $0 "> existed<" existed ">" >> "/dev/stderr" ;
  if ( substr($2, 1, 6) == NAM ) {
    existed = "T" ;
    prt_it() ;
  } else {
    print ;
  }
}
END {
# print "END existed<" existed ">" >> "/dev/stderr" ;
  if ( existed == "F" )
    prt_it() ;
}
EOGAWK
PPvarsBad='' # Assume false
for ii in PPTARL PPTARM PPTARD
do  # echo "/_/l/PP/h/$PPHOST/PPvars before $ii:"
    # /bin/cat /_/l/PP/h/$PPHOST/PPvars
    eval echo \${$ii} > /tmp/prtpkg_mount.eval
    val=`/bin/cat /tmp/prtpkg_mount.eval \
         | /usr/bin/tr -d '\012' \
        `
    /bin/rm -f /tmp/prtpkg_mount.eval
    /usr/bin/gawk -v NAM=$ii -v VAL=$val -f /tmp/prtpkg_mount.awk \
                  /_/l/PP/h/$PPHOST/PPvars \
                > /_/l/PP/h/$PPHOST/PPvars.new 2> /_/l/PP/h/$PPHOST/PPvars.err
    if   test -s /_/l/PP/h/$PPHOST/PPvars.err
    then echo "gawk error(s):"
         /bin/cat /_/l/PP/h/$PPHOST/PPvars.err
        PPvarsBad="T"
    else /bin/mv -f /_/l/PP/h/$PPHOST/PPvars.new /_/l/PP/h/$PPHOST/PPvars
         # echo "/_/PP/l/$PPHOST/h/PPvars after  $ii:"
         # /bin/cat /_/l/PP/h/$PPHOST/PPvars
    fi
    /bin/rm -f /_/l/PP/h/$PPHOST/PPvars.err
done
/bin/rm -f /tmp/prtpkg_mount.awk
test \( $PPvarsBad \) \
  && { echo "There was a problem processing /_/l/PP/h/$PPHOST/PPvars"
       exit 32
     } \
  || echo "Set PPTARD<$PPTARD> PPTARL<$PPTARL> PPTARM<$PPTARM> in /_/l/PP/h/$PPHOST/PPvars"
echo "These filesystems are mounted (prtpkg_chroot will handle any needed bind mounts)."
/bin/df -alHT
```

7.7.22. prtpkg_pdrive

This ***dash*** script examines and optionally (re)partitions and (re)formats the primary SSD/HDD for a platform that has booted either:

- the CRUX ISO using its canonical RAM disk root filesystem or
- a persistent CRUX Maintenance Environment (CME) that boots from the platform's primary SSD/HDD and uses a small root filesystem based upon the canonical CRUX ISO system’s RAM disk root filesystem

and then run the ***prtpkg_source*** script (see Section **7.6.16** for details). Refer to Section **6** for more information about the processes using this script.

If $1 is build, (re)partition the platform's primary SSD/HDD drive per the PPBORU environment variable (UEFI for creating an ESP or BIOS for not including an ESP) for a target CRUX (TC) root filesystem installation with or without a persistent maintenance environment (ME) root filesystem, and including one or more swap partitions.

If $1 is omitted, only verify the primary SSD/HDD drive is acceptably partitioned. For this option only, the PPBORU environment variable may be preset to the null string.

This script requires the PPHOST, PPTYPE, and PPBORU environment variables to be predefined, which is normally accomplished by the ***prtpkg_source*** file prior to invoking this script the first time.

For all options, identify, with input from the user as needed, what drive is the primary SSD/HDD as well as what partitions are to be used for what by the current invocation of the CRUX canonical or permanent maintenance environment, and define the following environment variables to be used by prtpkg maintenance environment scripts (others are defined by ***prtpkg_source***):

PPPDRV	the basename of the Primary DRiVe; e.g., sda

PPESPL	EFI System Partition Label (if null, there isn't any ESP)
PPBOTL	legacy BOoT partition Label (if null, there isn't any—if PPESPL is also null, /boot is in the root partition)
PPTCRL	Target Crux Root filesystem Label
PPMERL	Maintenance Environment Root filesystem Label
PPTCMP	Target Crux root filesystem Mount Point
PPMEMP	Maintenance Environment root filesystem Mount Point

These will be preserved in the /root/PPvars file by this script. The file will be sourced by the ***prtpkg_source*** file after the script concludes to provide the values to all subsequent child processes. If the /root directory exists in a persistent (vis-à-vis RAMdisk) partition, those values will be available to subsequent bootups of the maintenance environment.

If the PPBORU environment variable was set to BIOS and the PPESPL variable is set to the null string by this script, unmodified execution of this script's build option will produce a primary SSD/HDD drive looking like this according to execution of a ***/usr/sbin/parted unit MiB print free*** command (hardware specifics and total space will almost certainly be different):

```
Model: ATA Hitachi HTS54501 (scsi)
Disk /dev/sda: 152628MiB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start     End       Size     File system     Name  Flags
 1      0.02MiB   1.00MiB   0.98MiB                  X0    bios_grub
 6      1.00MiB     32MiB     31MiB  ext4            XB    legacy_boot
 2        32MiB    544MiB    512MiB  linux-swap(v1)  X1
 3       544MiB   1056MiB    512MiB  linux-swap(v1)  X2
 4      1056MiB   1568MiB    512MiB  linux-swap(v1)  X3
 5      1568MiB   2080MiB    512MiB  linux-swap(v1)  X4
 7      2080MiB   2920MiB    800MiB  ext4            XM
 8      2920MiB 152628MiB 149708MiB  ext4            XD
```

Note ***grub*** is not used by either the unmodified BIOS or the unmodified UEFI flavors, so technically a bios_grub partition need not be allocated (and that is true even if ***grub*** *is* being used). However, it is more tidy to ensure that extent remains reserved—it is quite small and maybe someday ***grub*** will want to make use of it, perhaps when live-booting some non-prtpkg system.

If the PPBORU environment variable was set to UEFI, the PPESPL variable must be defined, as a functional ESP is required for UEFI operation. In this case, unmodified execution of this script's build option will instead produce a primary SSD/HDD drive looking like this:

```
Model: ATA Hitachi HTS54501 (scsi)
Disk /dev/sda: 152628MiB
Sector size (logical/physical): 512B/512B
Partition Table: gpt
Disk Flags:

Number  Start      End        Size     File system     Name  Flags
 1      0.02MiB    1.00MiB   0.98MiB                   X0    bios_grub
 6      1.00MiB     256MiB    255MiB   fat32           XA    boot, esp
 7       256MiB     320MiB     64MiB   ext4            XB
 2       320MiB     832MiB    512MiB   linux-swap(v1)  X1
 3       832MiB    1344MiB    512MiB   linux-swap(v1)  X2
 4      1344MiB    1856MiB    512MiB   linux-swap(v1)  X3
 5      1856MiB    2368MiB    512MiB   linux-swap(v1)  X4
 8      2368MiB    3168MiB    800MiB   ext4            XM
 9      3168MiB  152628MiB 149460MiB   ext4            XD
```

The above layout differs from the BIOS version by inserting a 255MiB ESP named XA as partition number 6. Note that, if desired, the XB partition's contents can instead be maintained in the ESP and the XB partition can be omitted. Also, the syslinux package is not deployed within a UEFI platform and the kernels therein must be built for direct UEFI loading.

```
#!/bin/dash
# CCIA_0.0    PP-stick base/prtpkg_pdrive ..:....5....:....6....:....7....:....8....:....9....:....0
# **************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# **************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization...............
# 19290131 xyzzy      Host plugh SSD Hx: 0 ESP, 1 boot, 4 swaps, 1 CM, 1 CT
#---------------------------------------------------------------------------------------------------
# See the CRUX 3.4 Commonwealth Manual for the CCI identified in the second line for the main
# documentation of this script.
#---------------------------------------------------------------------------------------------------
set -e +xv
if   test `whoami` != root
then echo 'Only root can use the script.'
     exit 2
fi
test ".$PPHOST" = '.' \
  && { echo "PPHOST is not set--aborting!"
       exit 32
     }
test    ".$PPTYPE" != '.ISO' \
     -a ".$PPTYPE" != '.CME' \
  && { echo "PPTYPE <$PPTYPE> is not valid--aborting!"
       exit 32
     }
test    ".$PPBORU" != '.BIOS' \
     -a ".$PPBORU" != '.UEFI' \
     -a ".$1" != . \
  && { echo "PPBORU <$PPBORU> is not valid--aborting!"
       exit 32
     }
if   test -f /root/PPvars
then . /root/PPvars
else #----------------------------------------------------------------------------------------------
```

```
     test ".$PPPDRV" = '.' \
       && export PPPDRV='' # No primary SSD/HDD device identified yet (coming soon)--we want to
                           # always ask about this one showing current state to help the user think
                           # about choosing the right drive.
     # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     # EDIT AS NEEDED these global variables:
     # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                           # Efi System Partition Label        needed by prtpkg_chroot for PCE & CME
                           # Set PPESPL to the null string if not using an ESP
     test ".$PPESPL" = '.' \
       && export PPESPL='XA'
     echo "export PPESPL='$PPESPL'" >>/root/PPvars
     # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                           # BOoT fs Label                     needed by prtpkg_chroot for PCE & CME
                           # Set PPBOTL to the null string if not using a boot partition
     test ".$PPBOTL" = '.' \
       && export PPBOTL='XB'
     echo "export PPBOTL='$PPBOTL'" >>/root/PPvars
     # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                           # Target CRUX Root fs Label         needed by prtpkg_chroot for TC only
     test ".$PPTCRL" = '.' \
       && export PPTCRL='XD'
     echo "export PPTCRL='$PPTCRL'" >>/root/PPvars
     # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                           # Production CRUX root Mount Point  needed by prtpkg_chroot for TC only
     test ".$PPTCMP" = '.' \
       && export PPTCMP="/_/l/$PPTCRL"
     echo "export PPTCMP='$PPTCMP'" >>/root/PPvars
     # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                           # Persistent CME Root fs Label      needed by prtpkg_chroot for CME only
     test ".$PPMERL" = . \
       && export PPMERL='XM'
     echo "export PPMERL='$PPMERL'" >>/root/PPvars
     # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
                           # Persistent CME root Mount Point   needed by prtpkg_chroot for CME only
     test ".$PPMEMP" = . \
       && export PPMEMP="/_/l/$PPMERL"
     echo "export PPMEMP='$PPMEMP'" >>/root/PPvars
     #---------------------------------------------------------------------------------------------
fi
# Show the available drives while getting the 1st, then ask which to use:
DRV1=`/usr/bin/gawk -f /root/prtpkg_getdrives.awk /proc/partitions`
if   test ".$PPPDRV" != '.'
then /usr/bin/grep -q '    '$PPPDRV'     ' /root/getdrives.all \
       && echo "Found pre-defined PPPDRV variable set to a valid drive." \
       || { DRVS=`/bin/sed -e 's/       / /g;s/^ //;s/ $//' /root/getdrives.all`
            echo "The pre-defined value for PPPDRV <$PPPDRV> is not among $DRVS--aborting!"
            exit 32
          }
else echo    'Enter the drive basename of the primary SSD/HDD or'
     read -p "enter a null string to take the default ($DRV1):  " ASK
     if   test .$ASK = .
     then PPPDRV=$DRV1
     else /usr/bin/grep -q '             '$ASK'        ' /root/getdrives.all \
            || { DRVS=`/bin/sed -e 's/  / /g;s/^ //;s/ $//' /root/getdrives.all`
                 echo "The response <$ASK> is not among $DRVS--aborting!"
                 exit 32
               }
          PPPDRV=$ASK
     fi
fi
/usr/bin/grep -q '^export PPPDRV=' /root/PPvars \
 || echo "export PPPDRV='$PPPDRV'" >>/root/PPvars
PPPDEV="/dev/$PPPDRV"
/bin/rm -f /root/getdrives.all
echo "`/bin/date -u` -- Using $PPPDEV as the $PPHOST platform's primary SSD/HDD drive"
#---------------------------------------------------------------------------------------------
# If 'build' was not specified, our work here is done:
test ".$1" != '.build' \
  && exit 0
#---------------------------------------------------------------------------------------------
```

```
# build was specified--begin (re)partitioning and (re)formatting processing:
test ".$PPBORU" = '.' \
  && { echo "Cannot build the primary SSD/HDD with a null PPBORU variable--aborting!"
       exit  32
     }
test    "$PPBORU" = 'UEFI' \
     -a ".$PPESPL" = '.' \
  && { echo "Cannot build the primary SSD/HDD without an ESP when PPBORU=UEFI--aborting!"
       exit  32
     }
test `/usr/bin/grep "^$PPPDEV" /proc/mounts \
      | /usr/bin/wc -l \
     ` -gt 0 \
  && { echo 'The specified drive has at least one filesystem mounted--aborting!'
       exit  32
     }
/usr/sbin/parted $PPPDEV unit MiB print free
read -p 'Last chance to prevent disaster--alter the drive reported above? [Y|n]:  ' ASK
test ".$ASK" != '.Y' \
  && exit 32
# Wipe any residual MBR/GRUB data:
/bin/dd if=/dev/zero of=$PPPDEV bs=8192 count=1024
if   test \( $PPESPL \)
then # (Re)partition the drive with an ESP:
     /usr/sbin/parted -s $PPPDEV mklabel gpt unit MiB \
       mkpart X0   ext2              34s     2047s                    toggle  1 bios_grub \
       mkpart X1   linux-swap       320       832  align-check opt  2 \
       mkpart X2   linux-swap       832      1344  align-check opt  3 \
       mkpart X3   linux-swap      1344      1856  align-check opt  4 \
       mkpart X4   linux-swap      1856      2368  align-check opt  5 \
       mkpart XA   fat32              1       256  align-check opt  6 toggle  6 boot \
       mkpart XB   ext4             256       320  align-check opt  7 toggle  7 legacy_boot \
       mkpart XM   ext4            2368      3168  align-check opt  8 \
       mkpart XD   ext4            3168 312581774s align-check opt  9 \
       unit s print free unit MiB print free unit GiB print free quit
     parts2clear='2 3 4 5 6 7 8 9' # These are partition numbers
else # (Re)partition the drive without an ESP:
     /usr/sbin/parted -s $PPPDEV mklabel gpt unit MiB \
       mkpart X0   ext2              34s     2047s                    toggle  1 bios_grub \
       mkpart X1   linux-swap        32       544  align-check opt  2 \
       mkpart X2   linux-swap       544      1056  align-check opt  3 \
       mkpart X3   linux-swap      1056      1568  align-check opt  4 \
       mkpart X4   linux-swap      1568      2080  align-check opt  5 \
       mkpart XB   ext4               1        32  align-check opt  6 toggle  6 legacy_boot \
       mkpart XM   ext4            2080      2920  align-check opt  7 \
       mkpart XD   ext4            2920 312581774s align-check opt  8 \
       unit s print free unit MiB print free unit GiB print free quit
     parts2clear='2 3 4 5 6 7 8' # These are partition numbers
fi
swapparts='1 2 3 4' # These are label numbers
set +e
set -xv
# Wipe any residual filesystem data in each partition to keep /dev/disk/by-label data
# from being inconsistent:
/bin/dd if=/dev/zero of=${PPPDEV}1 bs=512 count=2014 # Clear grub_config area
if   test ".$parts2clear" != '.'
then for ii in $parts2clear
     do  /bin/dd if=/dev/zero of=$PPPDEV$ii bs=8192 count=1024 # Clear 1st 8MiB
     done
fi
# Show the new partitions' udev status:
/bin/ls -lah $PPPDEV*
# Format any swap partitions:
set -e
if   test ".$swapparts" != '.'
then for ii in $swapparts
     do  jj=`/usr/bin/expr $ii \+ 1`
         /sbin/mkswap --pagesize 4096 --label X$ii $PPPDEV$jj
         /sbin/swapon -v $PPPDEV$jj # New labels can be slow to appear, so don't use -L
     done
     /sbin/swapon -s # Show active swap spaces
```

```
fi
do_mkfs() {
  set +e
  cmd="/root/prtpkg_mkfs $1 $2 $3"
  echo "Starting execution of $cmd... (you may run '/root_mkfsstat $1'"
  echo "from another session to cat its log while it's running."
  /root/prtpkg_mkfs "$1" "$2" "$3" > /_/l/PP/h/$PPHOST/prtpkg_mkfs.$1.log 2>&1
  rc=$?
  echo "mkfs $1 status<$rc>"
  test $rc -ne 0 \
    && exit 32
  set +e
}
if   test ".$PPESPL" = '.'
then do_mkfs XB 6 w
     do_mkfs XM 7 r
     do_mkfs XD 8
else do_mkfs XB 7 w
     do_mkfs XM 8 r
     do_mkfs XD 9
     # format and populate the ESP (this requires the dosfstools package):
     LANG=C /sbin/mkfs.vfat -F 32 -n $PPESPL ${PPPDEV}6
     /bin/mount -L $PPESPL /mnt
     ( cd /mnt \
         && /bin/tar xjf /_/l/PP/arcs/PA.tar.bz2
     )
     /bin/mv /mnt/\[PA\] /mnt/\[$PPESPL\]
     /bin/ls -lah /mnt
     /bin/umount /mnt
fi
setup_mount() { # $1 is the mount point, $2 is the label, $3 is the description string
  MOUNT="$1"
  LABEL="$2"
  /bin/mount -L $LABEL $MOUNT
  # Create root's .profile
  /bin/mkdir -p $MOUNT/root
  /bin/cat /root/prtpkg_root.profile \
           /root/prtpkg_shortcuts.bash.source > $MOUNT/root/.profile
  /bin/mkdir -p $MOUNT/usr/src # So setup can install just the kernel source
  /bin/mkdir $MOUNT/usr/bin
  # Add CRUX ISO modules and prtpkg modules to filesystems' /usr/bin:
  /bin/cp -p /usr/bin/serial_console \
             /usr/bin/setup \
             /usr/bin/setup-chroot \
             /usr/bin/setup-helper \
             /root/prtpkg_setup \
             /root/prtpkg_locking.dash.source  $MOUNT/usr/bin
  if   test    \( $PPBOTL \) \
            -o \( $PPESPL \)
  then test ! -d $MOUNT/boot \
         && /bin/mkdir $MOUNT/boot
       if   test  \( $PPBOTL \)
       then /bin/mount -L $PPBOTL $MOUNT/boot
            if   test \( $PPESPL \)
            then test ! -d $MOUNT/boot/efi \
                   && /bin/mkdir $MOUNT/boot/efi
                 /bin/mount -L $PPESPL $MOUNT/boot/efi
            fi
       else /bin/mount -L $PPESPL $MOUNT/boot
       fi
  fi
  echo "The $3 target environment mounts like so:"
  /bin/df -alHT
  /bin/mount -l
  test -f $MOUNT/boot/efi/\[$PPESPL\] \
    && /bin/umount $MOUNT/boot/efi
  test -f $MOUNT/boot/\[$PPBOTL\] \
    && /bin/umount $MOUNT/boot
  test -f $MOUNT/\[$LABEL\] \
    && /bin/umount $MOUNT
}
```

```
set +e
echo 'Primary Drive (SSD):'
/usr/sbin/parted -s $PPPDEV unit MiB print free
# If a persistent maintenance environment is requested, set up the mounted partitions to
# (re)install/upgrade the persistent maintenance environment filesystems, then unmount it all:
test \( $PPMERL \) \
  && setup_mount $PPMEMP $PPMERL "persistent maintenance"
# Set up the target CRUX environment's mounted partitions, then unmount it all:
setup_mount $PPTCMP $PPTCRL "target CRUX"
# Set up the mounted partitions to (re)install/upgrade the Production CRUX filesystems:
# Invoke prtpkg_mount to get the correct filesystem(s) mounted and report:
/root/prtpkg_mount ; rc=$?
if   test $rc -ne 0
then echo "The prtpkg_mount script encountered a problem"
else . /_/PP/$PPHOST/PPvars # Update the global variables
     test ".$PPTYPE" = '.ISO' \
       && { /bin/cp -p $PPTARM/root/.profile /root
            : >/\[ISORAM\]
          }
     if   test ".$PPTARL" = ".$PPMERL"
     then echo 'The persistent maintenance environment (CME) is ready for canonical setup.'
     else echo 'The target production CRUX environment (PCE) is mounted.  Either continue with'
          echo '/root/prtpkg_presetup if that has not been run yet; otherwise, you may'
          echo 'now run /root/prtpkg_setup to begin the prtpkg maintenance upon the'
          echo 'target CRUX environment.'
     fi
fi
```

7.7.23. prtpkg_presetup

```
#!/bin/dash
# CCIA_0.0    PP-stick base/prtpkg_presetup :....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#---------------------------------------------------------------------------------------------------
# The script is intended to be run in the CRUX ISO installation/upgrade system booted with a root
# filesystem (RAMdisk if booting the CRUX release ISO) populated from the ISO's rootfs.tar.xz file.
# The environment is established primarily to install or upgrade CRUX on a target root filesystem
# set which, for prtpkg commonwealths, likely involves exported NFS filesystems.  Also, it can be
# used as a rescue system to deal with situations when the target system must not be operational for
# the non-primary maintenance procedure or cannot be booted nominally.
#
# This script and the related not-yet-a-package prtpkg files are provided via an ext4-formatted
# USB-stick that will be presumed to be labeled 'PP' and referred to as the PP-stick.  These files
# are incorporated into the maintenance environment after the root prompt is initially proffered by
# running these commands:
#   /bin/mkdir -p /_/l/PP
#   /bin/mount -L PP /_/l/PP -o rw,noatime
#   . /_/l/PP/prtpkg_source
# where the sourced commands will:
#   1. Establish some short-cut alias commands,
#   2. Determine the host name, prompting for it if this is the canonical RAMDISK environment,
#   3. Export the host name into the PPHOST variable,
#   4. Export 'ISO' or 'CME' into the PPTYPE variable,
#   5. Copy files from the PP-stick into the root filesystem /root directory, and
#   6. Launch $EDITOR on the copy of this script in /root to enable you to customize it for your
#      needs before running it.
#   7. Launch $EDITOR on the copy of the prtpkg_pdrive source file in /root to enable you to
#      customize it for your needs before running it.
#   8. Source the prtpkg_pdrive file to identify the primary SSD/HDD and the partitions of interest
#      to the prtpkg maintenance environment, storing the information in environment variables.
#   9. If necessary, create the /root/PPvars source file containing export statements for all the
#      environment variables assembled for persistence across command invocations (and bootups in
#      the case of the permanent root filesystem).
# This script is then invoked via the command:
#   /root/prtpkg_presetup 2>&1 | /usr/bin/tee /_/l/PP/h/$PPHOST/prtpkg_presetup.log
```

```
#
# Function:
#
# Assuming sufficient space in the booted root RAMdisk filesystem and the platform's available
# memory permits, this script prepares the booted canonical setup environment for running the
# prtpkg_setup script by:
#   - Setting up the root filesystem for prt-get processing,
#   - Building and installing, for NTP protocol support, the proven-most-secure (non-ISO) opt/chrony
#     package,
#       Note: contrib/ntp and the non-packaged Doolittle ntpclient are supported and have been
#             tested, mostly as an example of how to go about doing such things
#   - Synchronizing the system's clock with a nearby NTP server, and
#   - Optionally configuring and starting the NFS software for using a multi-platform commonwealth.
# If space does not permit this preferred approach, but there is enough space to perform the above
# without building the needed software (by installing pre-built packages from the PP-stick), that
# route will be taken.  Otherwise, this script will promptly fail.  Then the canonical setup script
# must be used per the CRUX Handbook to create a permanent bootable SSD or HDD maintenance root
# filesystem.  That root filesystem is then booted to provide a maintenance environment that CAN run
# this script to prepare the real target filesystem for CRUX installation or upgrade within a prtpkg
# commonwealth.  The details for this process are explained in the
# permanent_maintenance_filesystem.txt file in the PP-stick.
#
# This script makes an educated guess at the target root filesystem(s) to be mounted for the setup
# or prtpkg_setup invocation to follow this script's nominal results.  The variables PPTARD, PPTARL,
# and PPTARM are written into the /_/l/PP/h/$PPHOST/prtpkg_setup.vars file which is intended to be
# sourced by whatever scripts need that information.
#
# Notes:
#
# Built-in dash commands are coded as basenames, while external commands are coded with absolute
# paths to remind us the commands are in the maintennce root filesystem and are not to be confused
# with commands installed into the target mounted root filesystem; e.g.,/mnt/bin/mount.  This script
# doesn't deal with that complication but prtpkg_setup invokes some commands specifying a $ROOT
# prefix.  As the PATH may be uncertain, it's safer to avoid errors by being explicit and
# consistent.
#
# The design of this script has prioritized customization ease by using variable
# definitions loosely up front for most of the choices, along with early definition of the
# configuration files to be deployed in the maintenance root filesystem.  Look for "EDIT AS NEEDED"
# in sections that may contain expected editing.  Even so, it is prudent to look at all the code
# with an eye for any changes that may be essential or merely useful for the particular platform.
# Don't forget you may need to rerun this script for future maintenance of this platform, possibly
# even a complete rebuilding.
#----------------------------------------------------------------------------------------------------
set -e +xv
# Ensure this was invoked by root:
test `/usr/bin/whoami` != 'root' \
  && { echo 'You must be root to run this script--aborting!'
       exit 8
     }
# Ensure prtpkg_source was previously and successfully sourced:
test    ! \( $PPHOST \) \
     -o \(    ".$PPTYPE" != '.ISO' \
           -a ".$PPTYPE" != '.CME' \
        \) \
  && { echo "The PPHOST<$PPHOST> and/or PPTYPE<$PPTYPE> variables are improper--aborting!"
       echo "The prtpkg_source file must be successfully sourced before running this script."
       exit 32
     }
#----------------------------------------------------------------------------------------------------
# EDIT AS NEEDED variables:  (prtpkg_source and prtpkg_pdrive defined many environment variables)
MUSTBUILD='true'          # Set to '' if you want to install pre-built non-ISO packages from the
                          # PP-stick even if space to build from crux.nu ports exists--the true
                          # setting requires building
NEEDNFS='true'            # If single-platform commonwealth, may be set to '' (false)
# ----------------------- Note: These networking variable are only needed if running the canonical
#                               ISO environment, If Running in the CME environment, the network has
#                               already been defined when the CME root filesystem was constructed
#                               and should be operating when this script is run.
NETDEV='enp?s0'           # Interface ID to network through
IPV4='ddd.ddd.ddd.ddd'    # Interface's IPv4 address
```

```
CIDR='dd'                  # Number of network bits in the IPV4 address
IPV4GW='ddd.ddd.ddd.ddd'   # Default IPv4 gateway address
DOMAINDOT='domainname.'    # /etc/resolv.conf domain/search
DNSHOST='ddd.ddd.ddd.ddd' # /etc/resolv.conf nameserver
# ------------------------ End of network variables
NTPD='chrony'              # Choose chrony, ntpd, or ntpclient for NTP protocol software
                           # If using ntpd, enable the prtdir for contrib in prt-get.conf below
NTPHOST='ddd.ddd.ddd.ddd' # IPv4 of the nearby NTP server for NTP protocol software
#TPHOST='92.242.140.21'    # IPv4 of the nearby NTP server (time-b.nist.gov)
#TPHOST='104.245.32.240'   # IPv4 of the nearby NTP server (cns1.atlantic.net)
# chrony variables: none (but edit the chrony.conf file input below)
# Mills' ntpd variables: none (but edit the ntp.conf file input below)
# Doolittle's' ntpclient variables:
ADJFREQ=''                 # ntpclient -f value for this platform (null for unknown)
#---------------------------------------------------------------------------------------------------
# Acquire any filesystem label information (create a filesystem label file if there is none):
FSLABEL=`/bin/mount -l \
         | /usr/bin/gawk '/ on [/] /{if($0 ~ "[[]"){sub("^.*[[]","");sub("]$","");print}}' \
        `
test ".$FSLABEL" = . \
  && echo "The root filesystem has no label according to the mount command." \
  || echo "Found filesystem label $FSLABEL mounted as root filesystem."
LABELFILE=`echo /\[*\] \
           | /usr/bin/gawk '{print substr($1,2)}NR>1{print "multiple files"}' \
          `
if   test ".$LABELFILE" = '.[*]'
then echo 'The root filesystem does not contain a label file.'
     LABELFILE=''
else echo $LABELFILE \
     | /usr/bin/grep -s 'multiple files' \
       && { echo "Found multiple label files on the root filesystem!"
            exit 32
          } \
       || echo "Found root label file $LABELFILE"
fi
if   test ".$LABELFILE" = .
then test ".$FSLABEL" = . \
       && LABELFILE='[MAINT]' \
       || LABELFILE="$FSLABEL"
     echo "Created root filesystem label file $LABELFILE"
     . >/"$LABELFILE"
else test    ".$FSLABEL"  != . \
          -a "[$FSLABEL]" != "$LABELFILE" \
       && { echo "Filesystem label $FSLABEL is not label file $LABELFILE--aborting!"
            exit 32
          }
fi
echo "`/bin/date -u` --"
echo "    $0 is running within the $PPTYPE $PPHOST$LABELFILE maintenance environment"
#---------------------------------------------------------------------------------------------------
get_stats() { # $1 is 'boot', 'min ', 'nfs ', 'bld', 'pkga', 'prew', 'posw', 'post', or 'last'
    echo "$1       Total_KiB      Used_KIB      Free_KiB     Avail_KiB"
    ROOTFSTOTAL=`/bin/df / | /usr/bin/gawk 'NR==2{print $2}'`
    ROOTFSUSED=`/bin/df  / | /usr/bin/gawk 'NR==2{print $3}'`
    ROOTFSFREE=`/bin/df  / | /usr/bin/gawk 'NR==2{print $4}'`
    MEMTOTAL=`/usr/bin/gawk '/^MemTotal:/{print $2}' /proc/meminfo`
    MEMFREE=`/usr/bin/gawk '/^MemFree:/{print $2}' /proc/meminfo`
    MEMAVAIL=`/usr/bin/gawk '/^MemAvailable:/{print $2}' /proc/meminfo`
    MEMUSED=`/usr/bin/expr $MEMTOTAL \- $MEMFREE`
    echo $ROOTFSTOTAL $ROOTFSUSED $ROOTFSFREE \
    | /usr/bin/gawk '{printf("rootfs: %12u %12u %12u\n",      $1, $2, $3)}'
    echo $MEMTOTAL $MEMUSED $MEMFREE $MEMAVAIL \
    | /usr/bin/gawk '{printf("memory: %12u %12u %12u %12u\n", $1, $2, $3, $4)}'
}
#---------------------------------------------------------------------------------------------------
# EDIT AS NEEDED if changing what needs to be built or (re)installed
# Ensure the RAMdisk root filesystem and platform memory have enough free space to install the
# software needed to prepare the maintenance environment to run the CRUX setup script within a
# prtpkg commonwealth, else quit.  If there is not enough space to properly build non-ISO packages
# from source, then we will simply install pre-built packages from the PP-stick.  All numeric
# variables are KiB numbers (units of 1024; e.g., 1024=1MiB=1024*1024 bytes).
# EDIT AS NEEDED variables:
RAMFSBOOT=315476  # Minimum root fs space needed to boot the RAM root fs ISO system; i.e., the
                  # RAM root fs space in use after booting the canonical environment
PRMFSBOOT=1289124 # Minimum root fs space needed to boot a permanent root filesystem
RFSTOBUILD=288788 # Minimum free root filesystem space needed to install all pre-built packages
                  # needed to build and install non-ISO pre-built packages
RFSNONISO=100000  # Minimum free root filesystem space needed after installing build tools to
                  # build and install non-ISO packages
MEMMINRAM=368580  # Used memory after booting the canonical environment
MEMMINPRM=172588  # Used memory after booting the permanent environment
MEMTOBUILD=429544 # Minimum available memory needed to install all pre-built packages needed
                  # to build and install non-ISO pre-built packages
MEMNONISO=100000  # Minimum available memory after installing build tools needed to build
                  # and install non-ISO packages
get_stats 'boot'
# Computed variables:
CANNOTBUILD='' # Assume the hardware will support a full building environment
MEMAVAILBOOT=`/usr/bin/gawk '/^MemAvailable:/{print $2}' /proc/meminfo`
RFSAVAILBOOT=`/bin/df / \
             | /usr/bin/gawk 'NR==2{print $4}' \
             `
test $MEMAVAILBOOT -lt `/usr/bin/expr $MEMTOBUILD \+ $MEMNONISO` \
  && { echo 'This platform does not have enough RAM to support building the needed'
       echo 'non-ISO packages.'
       CANNOTBUILD='true'
     }
test $RFSAVAILBOOT -lt `/usr/bin/expr $RFSTOBUILD \+ $RFSNONISO` \
  && { echo 'This platform does not have enough free space in the root filesystem to'
       echo 'support building the needed non-ISO packages.'
       CANNOTBUILD='true'
     }
test    \( $CANNOTBUILD \) \
     -a \( $MUSTBUILD   \) \
  && { echo 'Variable MUSTBUILD is true but we cannot build--aborting!'
       if   test $PPTYPE = 'ISO'
       then echo 'Consider creating a permanent maintenance root filesystem.'
            echo 'Read persistent_filesystem.txt in the PP-stick for details.'
       fi
       exit 32
     }
### Move this to prtpkg_postsetup:
### test       ".$ROOT" = '.' \
###      -o ! -f $ROOT/usr/prtpkg/type \
###      -o   -f $ROOT/usr/prtpkg/_under_construction_ \
###   && { echo "variable ROOT<$ROOT> is not the target root filesystem or"
###        echo "$ROOT/usr/prtpkg is not yet installed--aborting!"
###        exit 4
###      }
#----------------------------------------------------------------------------------------------------
# Show the setting of these booleans:
show_bool() { # $1 is shell variable name
    eval v=\$$1
    echo -n "${1}=<$v> " ; test $v && echo 'true' || echo 'false'
}
show_bool MUSTBUILD
show_bool CANNOTBUILD
show_bool NEEDNFS
#----------------------------------------------------------------------------------------------------
# Define ISO packages to be (re)installed into the ISO root RAM filesys (mostly automatic,
# but EDIT AS NEEDED):
PKGS='fakeroot' # All options need these packages
# UEFI needs these packages:
test $PPBORU = 'UEFI' \
  && PKGS="$PKGS dosfstools efibootmgr efivar gnu-efi"
# NFS needs these packages:
test \( $NEEDNFS \) \
  && PKGS="$PKGS nfs-utils keyutils libevent libnfsidmap libtirpc rpcbind sqlite3"
# The chrony program needs these packages to run:
test ".$NTPD" = '.chrony' \
  && PKGS="$PKGS shadow"
# The ntpd program needs these packages to run:
```

```
test ".$NTPD" = '.ntpd' \
  && PKGS="$PKGS shadow gcc"
# The build environment (for ntp anyway) needs these packages (re)installed:
test $NTPD = ntpclient \
  && PKGS="$PKGS gcc libmpc glibc pkg-config binutils libtool make libarchive file libcap"
test \( $CANNOTBUILD \) \
  || { PKGS="$PKGS httpup rsync ports perl ca-certificates prt-get"
       PKGS="$PKGS gcc libmpc glibc pkg-config binutils libtool make"
       PKGS="$PKGS libarchive diffutils file flex openssl bison libcap"
     }
ISOPKGS=`echo $PKGS \
         | /usr/bin/tr '\040' '\012' \
         | /usr/bin/sort \
         | /usr/bin/uniq \
        `
echo "ISO packages to be installed into the maintenance environment:"
echo $ISOPKGS \
| /usr/bin/tr '\040' '\012' \
| /usr/bin/gawk '\
BEGIN{\
 x=0;\
 p=""\
}\
{p=p sprintf("%-15.15s ",$1);\
 if (++x==5){\
  print "  " p;\
  p="";\
  x=0\
 }\
}\
END{if(x>0)print "  " p\
}'
#----------------------------------------------------------------------------------------------------
# If installing chrony, EDIT AS NEEDED to define the ntp configuration that will synchronize this
# platform's system clock with the available NTP servers and/or peers and optionally support NTP
# clients:
test ".$NTPD" = '.chrony' \
  && /bin/cat <<EOD >/root/chrony.conf # Will replace /etc/chrony.conf after installing chrony
# Provide nearby (preferably no router hops) time servers provisioned by the enterprise:
server $NTPHOST iburst
#--------
# You may find a public (open access) stratum 1 server nearby--visit
# http://support.ntp.org/bin/view/Servers/StratumOneTimeServers
# and comply with the rules of engagement it links to.  For each
# system therein, you can click on the entry's country code to see
# the details for using that particular system.  For example,
# time-b.nist.gov stratum 1 pubic server:
#   per sysadmin's request (http://support.ntp.org/bin/view/Servers/PublicTimeServer000278):
#     - USA only
#     - no need to notify them you will be using this server
#     - do not use the DNS name -- code IPv4 or IPv6 address
#     - request time no more frequently than once every 3 minutes per requesting
#       non-NATed address
#   per official NIST time service info at
#   https://www.nist.gov/pml/time-and-frequency-division/services/internet-time-service-its
#     - no locality preference noted
#     - no need to notify them you will be using this server
#     - use DNS name or IP address
#     - request time no more frequently than once every 4 seconds per requesting address
# server time-b.nist.gov iburst
#--------
# Very nearby stratum 2 public servers:
# server cns1.atlantic.net iburst
# server 104.245.32.240 iburst
#--------
# Use public NTP servers from the pool.ntp.org project.
pool 2.pool.ntp.org iburst
pool 2.pool.ntp.org iburst
#--------
# Record the rate at which the system clock gains/losses time.
driftfile /var/lib/chrony/drift
```

```
#--------
# Allow the system clock to be stepped in the first three updates
# if its offset is larger than 1 second.
makestep 1.0 3
#--------
# Enable kernel synchronization of the real-time clock (RTC).
rtcsync
# Define the hosts/subnets that may use this host as an NTP server
# allow 192.168.0.0/24
# allow 192.168.0.17/24
# allow 10.220.186.0/25
# allow 10.220.186.128/25
# Define the hosts/subnets that may send chronyc inquiries to this host
# bindcmdaddress 0.0.0.0
# cmdallow 10.220.186.128/25
EOD
#----------------------------------------------------------------------------------------------------
# If installing ntpd, EDIT AS NEEDED to define the ntp configuration that will synchronize this
# platform's system clock with the available NTP servers and/or peers and optionally support NTP
# clients:
test ".$NTPD" = '.ntpd' \
  && /bin/cat <<EOD >/root/ntp.conf # Will replace /etc/ntp/ntp.conf after installing ntp
#--------
# Provide nearby (preferably no router hops) time servers provisioned by the enterprise:
server $NTPHOST prefer iburst
#--------
# You may find a public (open access) stratum 1 server nearby--see notes in chrony.conf above
# server time-b.nist.gov prefer iburst
# server 129.6.15.29 prefer iburst
#--------
# Very nearby stratum 2 public servers:
# server cns1.atlantic.net prefer iburst
# server 104.245.32.240 prefer iburst
#--------
# Random stratum 2 public servers:
server 2.pool.ntp.org
server 2.pool.ntp.org
#--------
# The local system clock (last ditch source):
server 127.127.1.0
fudge  127.127.1.0 stratum 10
#--- Define remote and localhost network services permissions/restrictions: ---------------
# Any host not covered below can only receive time service:
restrict default noquery nopeer notrap
#--------
# Only allow read-only access from localhost:
restrict 127.0.0.1
restrict ::1
#--------
# Edit the NTP clients/subnets which should be able to designate our server as a time
# source and/or direct read-only ntpq requests here (notice that the noquery has been
# implicitly removed) -- add noserve flags to decline time service requests
# restrict 192.168.1.0    mask 255.255.255.0   nomodify
# restrict 192.168.17.0   mask 255.255.255.0   nomodify
# restrict 10.220.186.128 mask 255.255.255.128 nomodify
# restrict 10.220.186.0   mask 255.255.255.128 nomodify
#--- Other configuration items: -----------------------------------------------------------
enable kernel ntp
# Default paths
logfile   /var/log/ntp.log
pidfile   /var/run/ntp/ntpd.pid
driftfile /var/lib/ntp/ntp.drift
statsdir  /var/lib/ntp/stats/
# Avoid udp spoofed ddos attacks
disable monitor
EOD
#----------------------------------------------------------------------------------------------------
# If running the canonical ISO, the network is not configured nor started, but it is both if using
# a permanent root filesystem.  If the former, uncomment and EDIT AS NEEDED to configure and start
# the network (note that /etc/resolv.conf is overlaid by pkgadd activity so it is backed up in
# /etc/resolv.conf.wait for recovery after the pkgadd activity has been completed):
```

```
if   test "$PPTYPE" = 'CME'
then /bin/cp -p /etc/resolv.conf /etc/resolv.conf.wait
else # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     # For ethernet IPv4:
     /sbin/ip a add $IPV4/$CIDR dev $NETDEV
     /sbin/ip l set             dev $NETDEV up
     /sbin/ip r add default     via $IPV4GW
     # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     # For wireless IPv4:
     # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     # For ethernet IPv6:
     # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     # For wireless IPv6:
     # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
     # Provision /etc/resolv.conf as /etc/resolv.conf.wait:
     /bin/cat <<EOD >/etc/resolv.conf.wait
domain $DOMAINDOT
search $DOMAINDOT
nameserver $DNSHOST
EOD
     /bin/cp -p /etc/resolv.conf.wait /etc/resolv.conf
fi
#----------------------------------------------------------------------------------------------------
# Ensure the network is properly configured and operational:
/sbin/ip a
/sbin/ip r
/bin/ping -c 3 $DNSHOST
/bin/ls -lh /etc/resolv.conf
/bin/cat    /etc/resolv.conf
#----------------------------------------------------------------------------------------------------
# Ensure the booted ISO's RAM filesystem is prepared for remaining pkgutils activity:
#
# Provision /etc/pkgmk.conf
#
test -f /etc/pkgmk.conf \
  || /bin/cat <<'EOD' >/etc/pkgmk.conf
#
# /etc/pkgmk.conf: pkgmk(8) configuration
#

export CFLAGS="-O2 -march=x86-64 -pipe"
export CXXFLAGS="${CFLAGS}"

# export MAKEFLAGS="-j2"

case ${PKGMK_ARCH} in
             "64"|"")
                          ;;
             "32")
                          export CFLAGS="${CFLAGS} -m32"
                          export CXXFLAGS="${CXXFLAGS} -m32"
                          export LDFLAGS="${LDFLAGS} -m32"
                          export PKG_CONFIG_LIBDIR="/usr/lib32/pkgconfig"
                          ;;
             *)
                          echo "Unknown architecture selected! Exiting."
                          exit 1
                          ;;
esac

# PKGMK_SOURCE_MIRRORS=()
# PKGMK_SOURCE_DIR="$PWD"
# PKGMK_PACKAGE_DIR="$PWD"
# PKGMK_WORK_DIR="$PWD/work"
# PKGMK_DOWNLOAD="no"
# PKGMK_IGNORE_FOOTPRINT="no"
# PKGMK_IGNORE_NEW="no"
# PKGMK_NO_STRIP="no"
# PKGMK_DOWNLOAD_PROG="wget"
# PKGMK_WGET_OPTS=""
# PKGMK_CURL_OPTS=""
# PKGMK_COMPRESSION_MODE="gz"

# End of file
EOD
#-----------------------------------------------------------------------------------------------------
# Install into the booted ISO's RAM filesystem the needed ISO-3-3 package files not pre-installed
# within the ISO's $MEDIA/rootfs.tar.xz root filesystem file:
#
for i in $ISOPKGS
do  /usr/bin/grep -s "^$i$" /var/lib/pkg/db 2>/dev/null \
      && echo "Package $i was previously installed" \
      || { j=`echo $MEDIA/crux/*/${i}#*`
           echo "`/bin/date -u` -- /usr/bin/pkgadd -f $j started..."
           /usr/bin/pkgadd -f $j 1>/root/pkgadd.$i.log 2>&1 ; rc=$?
           echo "`/bin/date -u` -- /usr/bin/pkgadd -f $j ended with status $rc"
           get_stats 'pkga'
         }
done
# Restore the customized DNS resolver file:
/bin/mv /etc/resolv.conf.wait /etc/resolv.conf
#-----------------------------------------------------------------------------------------------------
# Ensure the booted ISO's RAM filesystem is prepared for prt-get activity if RAM is not limited:
#
test \( $CANNOTBUILD \) \
  || { test -f /etc/ports/contrib.rsync.inactive \
         && /bin/mv /etc/ports/contrib.rsync.inactive /etc/ports/contrib.rsync
       test -d /usr/ports \
         || /bin/mkdir /usr/ports
       test -d /usr/ports/core \
         || /bin/mkdir /usr/ports/core
       test -d /usr/ports/opt \
         || /bin/mkdir /usr/ports/opt
       test -d /usr/ports/xorg \
         || /bin/mkdir /usr/ports/xorg
       test -d /usr/ports/contrib \
         || /bin/mkdir /usr/ports/contrib
       # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
       # Provision /var/lib/pkg/prt-get.aliases
       test -f /var/lib/pkg/prt-get.aliases \
         || /bin/cat <<'EOD' >/var/lib/pkg/prt-get.aliases
j2sdk: j2re
j2sdk: jre
jdk: jre
openmotif: lesstif
postfix: sendmail
exim: sendmail
qmail: sendmail
masqmail: sendmail
xorg: x11
EOD
       # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
       # Provision public signify keys
       # Unnecessary if the ports package is installed
       ## if   test ! -f /etc/ports/core.pub
       ## then for i in 6c37-dropin 6c37-git 6c37 compat-32 contrib core deepthought df \
       ##                j_v jaeger jmf jue opt therealfun timcow xfce xorg
       ##      do  /usr/bin/wget  -a /tmp/wget.log -c -nH -P /etc/ports \
       ##                   "https://crux.nu/keys/$i.pub"
       ##          echo "wget of $i.pub produced status code $?"
       ##      done
       ## fi
       /bin/ls -lahd /etc/ports/*
       # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
       # Provision /etc/prt-get.conf with prtdir for contrib
       /bin/cat <<'EOD' >/etc/prt-get.conf
###
### prt-get conf
###

# note: the order matters: the package found first is used
prtdir /usr/ports/core
prtdir /usr/ports/opt
prtdir /usr/ports/xorg
# prtdir /usr/ports/compat-32
# prtdir /usr/ports/contrib

# the following line enables the multilib compat-32 collection
#prtdir /usr/ports/compat-32

# the following line enables the user maintained contrib collection
#prtdir /usr/ports/contrib

### use mypackage form local directory
# prtdir /home/packages/build:mypackage

### log options:
# writelog enabled         # (enabled|disabled)
# logmode  overwrite       # (append|overwrite)
# rmlog_on_success yes     # (no|yes)
logfile  /var/log/pkgbuild/%n.log
                           # path, %p=path to port dir, %n=port name
                           #       %v=version, %r=release

### use alternate cache file (default: /var/lib/pkg/prt-get.cache
# cachefile /mnt/nfs/cache

### print README information:
# readme verbose           # (verbose|compact|disabled)

### prefer higher versions in sysup / diff
# preferhigher no      # (yes|no)

### use regexp search
# useregex no        # (yes|no)

### run pre- and post-installs scripts; yes is equivalent to the
### --install-scripts option
# runscripts no            # (no|yes)


### EXPERT SECTION ###

### alternative commands
# makecommand      pkgmk
# addcommand       pkgadd
# removecommand    pkgrm
# runscriptcommand sh
EOD
       # - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
       do_port() { # $1 is collection, $2 is package, and $3 is rsync or httpup
       # For any needed CRUX-3.3 package files not distributed within the ISO image, retrieve
       # the port from crux.nu, then build and install it into the booted ISO's RAM filesystem:
       #
           dp_olddir=$PWD
           cd /usr/ports/$1
           case ".$3" in
             '.rsync'  ) /usr/bin/rsync -aqz crux.nu::ports/crux-3.3/$1/$2/ $2
                         ;;
             '.httpup' ) echo 'The httpup driver is not yet supported by prtpkg_presetup'
                         exit 32
                         ;;
             *         ) echo "The driver argument for do_prtget <$3> is unrecognized"
                         exit 32
                         ;;
           esac
           test -f $2/pre-install \
             && { echo "Running $2 pre-install script"
                    /bin/bash $2/pre-install
                }
           echo "`/bin/date -u` -- /usr/bin/prt-get -is -if -kw $2 started..."
           /usr/bin/prt-get install -is -if -kw $2 >/root/prtget.$2.log 2>&1 ; rc=$?
           echo "`/bin/date -u` -- /usr/bin/prt-get -is -if -kw $2 ended with status $rc"
```

```
                get_stats 'prew'
                test $rc -eq 0 \
                  && ( cd   $2/work \
                         && /bin/tar cjf /root/$2.work.tar.bz2 . \
                         && cd .. \
                         && rm -rf work
                     )
                test    $rc -eq 0 \
                     -a -f $2/post-install \
                  && { echo "Running $2 post-install script"
                       /bin/bash $2/post-install
                     }
                cd $dp_olddir
                /usr/bin/bzip2 /root/prtget.$2.log
                get_stats 'posw'
            }
        } # End of ! CANNOTBUILD build environment preparation
    # Print post-buildprep rootfs and memory stats
    get_stats 'post'
    #----------------------------------------------------------------------------------------------------
    # Synchronize the platform's clocks after installing the desired software:
    #
    mk_ntp() { # $1 is collection, $2 is NTP package to install, $3 is 'rsync' or 'httpup'
        if   test \( $CANNOTBUILD \)
        then test -f /_/l/PP/${2}_pre-install \
               && /bin/bash /_/l/PP/${2}_pre-install
             j=`echo /_/l/PP/p/atom/${2}#*`
             echo "`/bin/date -u` -- /usr/bin/pkgadd $j started..."
             /usr/bin/pkgadd $j 1>/root/pkgadd.ntp.log 2>&1 ; rc=$?
             echo "`/bin/date -u` -- /usr/bin/pkgadd $j ended with status $rc"
             test    $rc -eq 0 \
                  -a -f /_/l/PP/${2}_post-install \
               && /bin/bash /_/l/PP/${2}_post-install
        else do_port $1 $2 $3
        fi
    }
    # Capture the boot time hwclock:
    echo "`date -u` -- /sbin/hwclock --show" 1> /var/log/hwclock
    /sbin/hwclock --show                     1>>/var/log/hwclock 2>&1
    # Install chrony, Mills' ntpd, or Doolittle's ntpclient:
    case $NTPD in
      'chrony' )
         mk_ntp opt chrony rsync
         # Activate the chronyd daemon and synchronize this platform's system clock:
         /bin/cp -p /root/chrony.conf /etc/chrony.conf
         echo "`/bin/date -u` -- Running /etc/rc.d/chronyd start"
         /etc/rc.d/chronyd start
         echo 'Sleeping 15 seconds to allow quick synchronization.'
         /bin/sleep 15s
         echo "System time is now `/bin/date -u '+%Y%m%d_%T.%NUT'`"
         /usr/bin/chronyc -m tracking sources
         echo 'Sleeping 30 seconds to allow better synchronization.'
         /bin/sleep 15s
         /usr/bin/chronyc -m tracking sources
         # If CME filesystem and not already done, insert chronyd into /etc/rc.conf services:
         if   test -f /etc/rc.conf
         then /bin/sed -e '/^SERVICES=(lo /s/(lo/(chronyd lo/' /etc/rc.conf >/etc/rc.conf.new
              test -s /etc/rc.conf.new \
                && /bin/mv /etc/rc.conf.new /etc/rc.conf \
                || /bin/rm -f /etc/rc.conf.new
         fi
         ;;
      'ntpd' )
         mk_ntp contrib ntp rsync
         # Activate the ntpd daemon and synchronize this platform's system clock:
         /bin/cp -p /root/ntp.conf /etc/ntp/ntp.conf
         # echo "`/bin/date -u` -- Running /etc/rc.d/ntpd start"
         # /etc/rc.d/ntpd start
         echo "`/bin/date -u` -- Running ntpd -u ntp:ntp -gqx"
         /usr/bin/ntpd -u ntp:ntp -gqx
         echo "System time is now `/bin/date -u '+%Y%m%d_%T.%NUT'`"
```

```
            echo "`/bin/date -u` -- Starting (ntpd -u ntp:ntp 1>/root/ntp.log 2>&1 &)"
            (ntpd -u ntp:ntp 1>/root/ntp.log 2>&1 &)
            # FIXME: local ntpq doesn't work, but ntpq from external platforms does
            # echo sysinfo | /usr/bin/ntpq 127.0.0.1
            # echo lpeers  | /usr/bin/ntpq $IPV4
            # If not already done, insert ntpd into /etc/rc.conf services:
            /bin/sed -e '/^SERVICES=(lo /s/(lo/(ntpd lo/' /etc/rc.conf >/etc/rc.conf.new
            test -s /etc/rc.conf.new \
              && /bin/mv /etc/rc.conf.new /etc/rc.conf \
              || /bin/rm -f /etc/rc.conf.new
            ;;
         'ntpclient' )
            cd /root
            if   test \( $CANNOTBUILD \)
            then echo "`/bin/date -u` -- Copying the pre-build Doolittle ntpclient executables."
                 cp -p /_/PP/atom/ntpclient /_/PP/atom/adjtimex /_/PP/rate.awk .
            else echo "`/bin/date -u` -- Building the Doolittle ntpclient."
                 tar xzf /_/PP/ntpclient_2015_365.tar.gz 1>>/root/build_ntpclient.log 2>&1
                 cd ntpclient-2015 # See the README and HOWTO files for this KISS clock synchronizer
                 /usr/bin/make                           1>>/root/build_ntpclient.log 2>&1
                 /usr/bin/make adjtimex                  1>>/root/build_ntpclient.log 2>&1
            fi
            # If the ntpclient -f value is not known, get a coarse approximation:
            if   test ! \( $ADJFREQ \)
            then # Change system clock close to that of the nearby NTP server:
                 echo "Forcing system clock to current time from NTP server at $NTPHOST:"
                 ./ntpclient -s -h $NTPHOST                  1> /var/log/ntpclient 2>&1
                 /bin/cat /var/log/ntpclient
                 # Set the RTC to the new system time:
                 echo "`/bin/date -u` -- /sbin/hwclock --systohc --noadjfile --utc --debug" \
                                                                  1>>/var/log/hwclock
                 /sbin/hwclock --systohc --noadjfile --utc --debug      1>>/var/log/hwclock 2>&1
                 echo "`/bin/date -u` -- /sbin/hwclock --show --debug"  1>>/var/log/hwclock
                 /sbin/hwclock --show --debug                           1>>/var/log/hwclock 2>&1
                 # Gather 20 data points in 5 minutes:
                 echo "Gathering samples of clock drift over the next 5 minutes--do not interrupt."
                 # Display ntpclient data headers:
                 echo 'DAYNO SEC_OF_DAY  ELAPSED    STALL      SKEW DISPERSE FREQUENCY' \
                 | /usr/bin/tee -a /var/log/ntpclient
                 ./ntpclient -i 15 -c 20 -q 100 -h $NTPHOST 2>&1 \
                 | /usr/bin/tee -a /root/ntpclient.dat
                 /bin/cat /root/ntpclient.dat            1>>/var/log/ntpclient
                 # Analyze data and extract coarse crystal adjustment value:
                 ADJFREQ=`/usr/bin/awk -f rate.awk /root/ntpclient.dat \
                          | /usr/bin/awk '/^new frequency/{print $3}' \
                         `
            fi
            # Reset system clock using coarse crystal adjustment;
            echo "Forcing system clock to current time from NTP server with frequency setting $ADJFREQ:"
            ./ntpclient -s -f $ADJFREQ -h $NTPHOST 2>&1 \
            | /usr/bin/tee /var/log/ntpclient
            # Start setting the RTC to the current system time every 11 minutes (since the kernel won't):
            ( echo "Starting 11-minute RTC update daemon logging to /var/log/hwclock:"
              ( while :
                do echo "`/bin/date -u` -- /sbin/hwclock --systohc --noadjfile --utc --debug"
                   /sbin/hwclock --systohc --noadjfile --utc --debug
                   echo "`/bin/date -u` -- /sbin/hwclock --show --debug"
                   /sbin/hwclock --show --debug
                   /bin/sleep 11m
                done
              ) </dev/null 1>>/var/log/hwclock 2>&1 &
            )
            # Sample NTPHOST time every 15 seconds and adjust frequency as needed forever, logging data
            # (in limited sampling, this has kept the clock within +-4 milliseconds of the NTP source
            # on the same LAN switch/subnet).  Note times are in microseconds--ntp shows milliseconds.
            echo "Starting phase locking ntpclient daemon logging to /var/log/ntpclient:"
            echo 'DAYNO SEC_OF_DAY  ELAPSED    STALL      SKEW DISPERSE FREQUENCY' >>/var/log/ntpclient
            ( $PWD/ntpclient -f $ADJFREQ -l -i 15 -q 100 -g 0 -h $NTPHOST \
                </dev/null 1>>/var/log/ntpclient 2>&1 &
            )
            /bin/ps -A -F -w | /usr/bin/grep "$PWD\/ntpclient " | /usr/bin/grep -v grep
```

```
    ;;
esac
# Show if the RTC has been updated yet by ntp
echo "`/bin/date -u` -- /sbin/hwclock --show" 1> /var/log/hwclock
/sbin/hwclock --show                          1>>/var/log/hwclock 2>&1
#----------------------------------------------------------------------------------------------------
# Configure packages and start new daemons as needed:
#
# Print final rootfs and memory stats and terminate
get_stats 'last'
```

7.7.24. prtpkg_root.profile

```
#!/bin/bash
# CCIA_0.0    PP-stick base/prtpkg_root.profile .5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# This file is designed to be sourced when logging on as root from bash or dash sessions within a
# CRUX maintenance environment to establish command aliases, environment variables, etc. in the
# usual shell profile manner.  The prtpkg_pdrive script copies this file to /root/.profile in the
# CME and PCE root filesystem(s) it creates and appends the /_/PP/prtpkg_shortcuts.bash.source file
# to it.
#----------------------------------------------------------------------------------------------------
export ROOTFS=`/bin/ls -1 /\[*\] \
              | /usr/bin/tr -d '/[]' \
              `
export PS1='\[\033[01;31m\]\h[${ROOTFS}]\[\033[01;37m\]\w \[\033[01;33m\]\$ \[\033[00m\]'
if   test -f /root/PPvars
then . /root/PPvars
fi
```

7.7.25. prtpkg_setup

```
#!/bin/dash
# CCIA_0.0    PP-stick base/prtpkg_setup ...:....5....:....6....:....7....:....8....:....9....:....0
# ***************************************************************************************************
# * Original setup        Copyright © 2001-2005 by Per Liden <per@fukt.bth.se>
# * Original setup-helper Copyright © [unknown] by Johannes Winkelmann <jw@tks6.net>
# * Enhancements          Copyright © 2006-2018 by CRUX Development Team
# * prtpkg enhancements   Copyright © 2015-2018 by David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ***************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# CRUX ISO Setup Supporting prtpkg Maintenance Methodology
#
# This script includes the original CRUX 3.4 setup and setup-helper scripts with setup-helper in
# front of the foundational setup.  The mainline sections of setup-helper were merged as the new
# setup_helper() function at the end of the setup-helper section of this combined script.  Because
# the setup script is the major original component, it is used as the basis of this new dash
# version.  The overall structure of both scripts is very close to the originals.  There are some
# clearly identified tweaks in places as well as a lot of new functions in the setup script's
# section.  Hopefully, it can be matched against the original but as the style was changed to make
# everything consistent and more easily parsed by programmers, diffs even character-by-character
# are not very much help.  However, the actual pipelines are unchanged except where there are
# timestamped comments.  The long strings used as dialog arguments were broken up into variable
# "msg" assembling logic.  Debugging/tracing code was added but, except for the leftmost (command)
# tokens, are aligned on the right side to help make it plain they are not part of the script's
# functional logic, making it easier to keep them separate in one's mind.
#
# Remember: This script runs under the CRUX ISO's barebones ramdisk root filesys.  Any extra
#           required packages such as nfs-utils must be somehow installed into the ramdisk before
#           running this script.
#----------------------------------------------------------------------------------------------------
set +xv
####################################################################################################
# Begin CRUX 3.4 setup-helper script inclusion section...                                          #
####################################################################################################
#!/bin/bash
# post setup
#
# Johannes Winkelmann, jw at tks6 dot net
###############################
#   I S _ I N S T A L L E D   #
###############################
is_installed() {
    /usr/bin/printf "%s" $(pkginfo -r $ROOT -i \
    | /usr/bin/gawk -v r="^$1\$" '$1 ~ r {print $1}')
}
#####################################
#   R E N A M E _ P A C K A G E S   #
#####################################
rename_packages() {
    echo                                                          'Entered rename_packages' >>$PPDBG
    # Change one or more package names in the package database file
    # Each arg is oldname:newname tuple
    if   test -z "$1"
    then return
    fi
    pkg=$@
    filter="sed "
    runfilter="no"
    for tuple in $@
    do  oldn=`echo $tuple \
            | /usr/bin/gawk -F : '{print $1}' \
           `
        newn=`echo $tuple \
            | /usr/bin/gawk -F : '{print $2}' \
           `
        if      test -n "$newn" \
             && test `is_installed $oldn`
        then # Given tuple xyzzy:plugh, append " -e 's|^xyzzy$|plugh|'" to filter
             echo "Package renamed:  $oldn -> $newn"
             filter="$filter -e 's|^$oldn\$|$newn|'"
             runfilter="yes"
        fi
    done
    if   test "$runfilter" = "no"
    then return
    fi
    :> $TMPDB
    if   test -f $DB
    then nextIsName=1 # Initial state is expecting a package name line next
         /bin/cat $DB \
         | while read l
         do if   test $nextIsName -eq 1
            then # This is a package name line--filter it
                 nextIsName=0
                 echo $l \
                 | eval $filter >> $TMPDB
            else if   test ".$l" = . # If package delimiter line (null), change state
                 then nextIsName=1
                 fi                  # Either way, pass it through
                 echo $l >> $TMPDB
            fi
         done
    fi
    $MV $DB $BACKUPDB
    $MV $TMPDB $DB
    # Note this leaves $BACKUPDB as cruft after setup ends
}
#####################################
#   R E M O V E _ P A C K A G E S   #
```

```
#####################################
remove_packages() {
    echo                                                          'Entered remove_packages' >>$PPDBG
    for package in $@
    do  if   test `is_installed $package`
        then echo "Package removed:  $package"
             $PKGRM -r $ROOT $package
        fi
    done
}
#####################################
#   I N J E C T _ P A C K A G E S   #
#####################################
inject_packages() {
    echo                                                          'Entered inject_packages' >>$PPDBG
    for package in $@
    do  pkg="`/usr/bin/find core opt xorg \
                            -name $package\#*.pkg.tar.[bgx]z* \
                            &2> /dev/null \
             `"
        echo                                                                        "pkg<$pkg>" >>$PPDBG
        if   test -z "$pkg"
        then echo "  ERROR: package $package not found on ISO"
        else if   test -z `is_installed $package`
             then echo "Package injected: $package"
                  $PKGADD -r $ROOT -f $pkg
             else echo "Package upgraded: $package"
                  $PKGADD -r $ROOT -u -f $pkg
             fi
        fi
    done
}
###############################
#   S E T U P _ H E L P E R   #
###############################
setup_helper() {
    echo                                                             'Entered setup_helper' >>$PPDBG
    ROOT=$1
    DB=$ROOT/var/lib/pkg/db
    ## those values should be changed:
    TMPDB=$DB.tmp
    BACKUPDB=$DB.backup
    PKGADD="/usr/bin/pkgadd"
    PKGRM="/usr/bin/pkgrm"
    MV="/bin/mv"
    if   test -z "$1"
    then echo "Usage: $0 <root>"
         exit -1
    fi
    # ##### # 3.2 -> 3.3 #####
    # echo "* CRUX 3.2 -> 3.3 setup_helper"
    # inject_packages signify
    # if   test `is_installed firefox`
    # then inject_packages autoconf-2.13
    # fi
    # if   test `is_installed xorg-server`
    # then inject_packages xorg-libxfont2
    # fi
    ##### # 3.3 -> 3.4 #####
    echo "* CRUX 3.3 -> 3.4 setup-helper"
    remove_packages xorg-bigreqsproto \
                    xorg-compositeproto \
                    xorg-damageproto \
                    xorg-dmxproto \
                    xorg-dri2proto \
                    xorg-dri3proto \
                    xorg-fixesproto \
                    xorg-fontsproto \
                    xorg-glproto \
                    xorg-inputproto \
                    xorg-kbprot \
                    xorg-presentproto \
                    xorg-randrproto \
                    xorg-recordproto \
                    xorg-renderproto \
                    xorg-resourceproto \
                    xorg-scrnsaverproto \
                    xorg-videoproto \
                    xorg-xcmiscproto \
                    xorg-xextproto \
                    xorg-xf86bigfontproto \
                    xorg-xf86dgaproto \
                    xorg-xf86driproto \
                    xorg-xf86miscproto \
                    xorg-xf86vidmodeproto \
                    xorg-xineramaproto xorg-xproto
    for package in xorg-libxv \
                   xorg-libx11 \
                   xorg-libxfont2 \
                   xorg-libxfixes \
                   xorg-server \
                   xorg-libxres \
                   xorg-rgb \
                   xorg-libxcomposite \
                   xorg-libxxf86vm \
                   xorg-libxtst \
                   xorg-libxrender \
                   xorg-libice \
                   xorg-libxdmcp \
                   libvdpau \
                   xorg-libxfont \
                   xorg-libxdamage \
                   xorg-libxxf86dga \
                   xorg-libxrandr \
                   xorg-libdmx \
                   mesa3d \
                   xorg-libfontenc \
                   xorg-libxinerama \
                   xorg-libxshmfence \
                   xorg-libxau
    do
        if   test `is_installed ${package}`
        then if   test ! `is_installed xorg-xorgproto`
             then inject_packages xorg-xorgproto
             fi
        fi
    done
    if   test `is_installed firefox`
    then inject_packages rust
    fi
    if   test `is_installed python3`
    then inject_packages libtirpc \
                         mpdecimal
    fi
}
####################################################################################################
# Begin CRUX 3.4 setup script inclusion section (with interspersed new prtpkg logic)...            #
####################################################################################################
VERSION="3.4"
#########################
#   D O _ D I A L O G   #
#########################
do_dialog() {
    echo                                                                      'Entered do_dialog' >>$PPDBG
    /usr/bin/dialog --backtitle "CRUX $VERSION Setup with prtpkg" --no-shadow "$@" ; rc=$?
    echo                                                             "Leaving do_dialog status $rc" >>$PPDBG
    return $rc
}
#######################
#   D O _ A B O R T   #
#######################
do_abort() {
    echo                                                                 'Entered do_abort' >>$PPDBG
    do_dialog --aspect 50 --defaultno --yesno "Abort installation?" 0 0 \
      && exit 1
    echo                                                        "Leaving do_abort status 0" >>$PPDBG
    return 0
}
#########################
#   D O _ S E L E C T   #
#########################
do_select() {
    echo                                                                'Entered do_select' >>$PPDBG
    while true
    do    do_dialog "$@" ; rc=$?
          if   test $rc != 0
          then do_abort
          else break
          fi
    done
    echo                                                     "Leaving do_select status $rc" >>$PPDBG
    return $rc
}
#####################
#   W E L C O M E   #
#####################
welcome() {
    echo                                                                  'Entered welcome' >>$PPDBG
    msg="Welcome!\n"
    msg="${msg}\nThis is the superset version of setup that supports the prtpkg software"
    msg="${msg} maintenance facility, which, in addition to the canonical setup processing,"
    msg="${msg} deals with installing or updating the prtpkg commonwealth in which this"
    msg="${msg} platform will participate.\n"
    msg="${msg}\nBefore continuing make sure you have read and understood the prtpkg"
    msg="${msg} documentation as you will need to intelligently interact with this script to"
    msg="${msg} achieve a satisfactory outcome.  The requirements of the canonical setup script"
    msg="${msg} are in place with this script as well, and are explained using the canonical"
    msg="${msg} welcome which immediately follows:\n"
    msg="${msg}\nThis script will guide you through the installation of CRUX packages.\n"
    msg="${msg}\nBefore starting the installation make sure you have read and understood"
    msg="${msg} the \"CRUX Installation Guide\". If you have done that then please continue,"
    msg="${msg} else abort the installation and come back later.\n"
    msg="${msg}\nAre you really sure you want to continue?"
    do_select --aspect 5 --yesno "$msg" 0 0
}
#################################
#   S E L E C T _ A C T I O N   #
#################################
select_action() {
    echo                                                            'Entered select_action' >>$PPDBG
    do_select --menu "Install or upgrade?" 9 45 2 \
                     "1" "Install CRUX $VERSION" \
                     "2" "Upgrade to CRUX $VERSION" 2> $tmpfile
    ACTION=`/bin/cat $tmpfile`
    echo                                                                  "ACTION<$ACTION>" >>$PPDBG
    if   test "$ACTION" = "1"
    then ACTION="INSTALL"
    else ACTION="UPGRADE"
         msg="NOTE!\n"
         msg="{$msg}\nBefore upgrading make sure /etc/pkgadd.conf in the old installation is"
         msg="{$msg} tuned to fit your needs, important files might otherwise be overwritten."
         msg="{$msg} Further, when setup has completed the upgrade you need to go though the"
         msg="{$msg} rejected files in /var/lib/pkg/rejected/ and upgrade them manually if"
         msg="{$msg} needed (or use the rejmerge(8) tool). See the pkgadd(8) man page for"
         msg="{$msg} more information about /etc/pkgadd.conf.\n"
         msg="{$msg}\nAre you really sure you want to continue?"
         do_select --aspect 5 --yesno "$msg" 0 0
    fi
    echo                                                                  "ACTION<$ACTION>" >>$PPDBG
}
#############################
#   S E L E C T _ R O O T   #
#############################
```

```
select_root() {
    echo                                                                  'Entered select_root' >>$PPDBG
    while true
    do    msg="Enter directory where your CRUX root partition is mounted:"
          do_select --aspect 40 --inputbox "$msg" 0 0 "/" 2> $tmpfile
          ROOT=`/bin/cat $tmpfile`
          echo                                                                    "ROOT<$ROOT>" >>$PPDBG
          if   test    ".$ROOT" != '.' \
                    -a -d "$ROOT"
          then if      test "$ACTION" = "INSTALL" \
                    || test -f "$ROOT/var/lib/pkg/db"
               then break
               fi
               msg="Directory does not look like a CRUX root directory. Try again."
               do_dialog --aspect 50 --msgbox "$msg" 0 0
          else msg="Directory not found. Try again."
               do_dialog --aspect 50 --msgbox "$msg" 0 0
          fi
    done
}
################################################
#   G E T _ S C E N A R I O   (new for prtpkg) #
################################################
get_scenario() { # Validate target filesys, commonwealth situation, and return what-to-do info
    # Expects ROOT is set to mountpoint of target filesystem
    # Returns:
    #   NEWCOMW ACTION- CWTYPE Scenario----------------------------------------------------------
    #   false   UPGRADE U      Update CRUX in pre-existing commonwealth's pre-installed boot cell
    #   false   INSTALL I      Install CRUX into new boot cell of a pre-existing commonwealth
    #   true    INSTALL S      Install CRUX into new single-prtpkg-platform commonwealth
    #   true    INSTALL M      Install CRUX into new multi-prtpkg-platform commonwealth
    # Else aborts
    # Exports HOSTNAME ROOTFS PRTPKG_CELL
    echo                                            "Entered get_scenario with ROOT<$ROOT>" >>$PPDBG
    # 2017-11-20 dlcusa: Disable target system dialog--either the input file is right or its not:
    if   check_root_invalid
  # then select_root
  #      if   check_root_invalid
         then echo "The specified mountpoint $ROOT was not found in the mount table--aborting"
              exit 32
  #      fi
    fi
    echo          "On return MNT<$r_mnt> DEV<$r_dev> EXL<$r_exl> TYP<$r_typ> OPTS<$r_opts>" >>$PPDBG
    test ".$r_exl" = '.' \
      && echo                               "Warning: Root filesystem $r_dev is unlabeled." >>$PPDBG
    /bin/ls                                                                   -lh $ROOT >>$PPDBG 2>&1
    /bin/cat <<"           EOGAWK" >$TMPDIR/label.awk
             {
               sub(".*/", "") ;
               print ;
             }
             NR == 2 {
               print "Multiple [] files found in target root filesys--aborting" > "$TMPDIR/error" ;
             }
             EOGAWK
    /bin/rm -f $TMPDIR/error
    r_inl=`/bin/ls -1 $ROOT/\[*\] 2>/dev/null \
           | /usr/bin/tee -a $PPDBG 2>&1 \
           | /usr/bin/gawk -f $TMPDIR/label.awk \
          `
    echo                                                                    "r_inl<$r_inl>" >>$PPDBG
    if   test -f $TMPDIR/error
    then /bin/cat  $TMPDIR/error
         exit 32
    elif test    ".$r_inl" = '.' \
              -o \(    ".$r_exl" != '.' \
                    -a ".$r_exl" != ".$r_inl" \
                 \)
    then echo "External label: <$r_exl>    Internal label: <$r_inl>"
         echo "Target root filesystem mounted on $ROOT has unacceptable labeling--aborting!"
         exit 32
    fi
    echo      "Using $r_dev on $ROOT as target--labels: internal<$r_inl> external<$r_exl>." >>$PPDBG
    if   test ! -f "$ROOT/etc/rc.conf"
    then if   test $ACTION = 'UPGRADE'
         then echo 'Cannot upgrade an installed target cell lacking /etc/rc.conf--aborting!'
              exit 32
         else select_host
         fi
    else # There is an rc.conf file...
         export HOSTNAME=`/bin/sed -e '/^HOSTNAME=/!d' -e 's/^HOSTNAME=//' $ROOT/etc/rc.conf`
    fi
    export ROOTFS=$r_inl
    export PRTPKG_CELL=$HOSTNAME$ROOTFS
    echo                  "HOSTNAME<$HOSTNAME> ROOTFS<$ROOTFS> PRTPKG_CELL<$PRTPKG_CELL>" >>$PPDBG
    if   test    ! -d "$ROOT/usr/prtpkg" \
              -a ! -L "$ROOT/usr/prtpkg"
    then commonwealth_new
    elif test      -f "$ROOT/usr/prtpkg/type" \
              -a ! -f "$ROOT/usr/prtpkg/_under_construction_"
    then commonwealth_old
    else commonwealth_new
    fi
    echo                              "ACTION<$ACTION> NEWCOMW<$NEWCOMW> CWTYPE<$CWTYPE>" >>$PPDBG
}
############################################################
#   C H E C K _ R O O T _ I N V A L I D   (new for prtpkg) #
############################################################
check_root_invalid() { # Validate $ROOT is adequately labeled
    # Expects ROOT is set to mountpoint of target filesystem
    # Returns 0 (true) if $ROOT is not a valid target filesystem; otherwise return 1 (false)
    #     and r_dev r_mnt r_opts r_exl
    # May abort rc=32
    # Modifies $tmpfile
    echo                                                   'Entered check_root_invalid' >>$PPDBG
    /bin/mount -l \
    | while read r_dev r_l_on r_mnt r_l_typ r_typ r_opts r_exl r_xtra
      do echo      "mount -l: $r_dev $r_l_on $r_mnt $r_l_typ $r_typ $r_opts $r_exl $r_xtra" >>$PPDBG
         if   test    ".$r_l_on"  != '.on' \
                   -o ".$r_l_typ" != '.type'
         then echo '/bin/mount -l provided unexpected output--aborting'
              exit 32
         fi
         test ".$r_mnt" != ".$ROOT" \
           && continue
         echo                                       "Found device $r_dev mounted as $ROOT" >>$PPDBG
         echo "$r_dev $r_mnt $r_typ $r_opts $r_exl" >$tmpfile
         break
      done
    read r_dev r_mnt r_typ r_opts r_exl r_xtra <$tmpfile
    echo              "ROOT: MNT<$r_mnt> DEV<$r_dev> EXL<$r_exl> TYP<$r_typ> OPTS<$r_opts>" >>$PPDBG
    test ".$ROOT" != ".$r_mnt" \
      && return 0 \
      || return 1
}
##############################################
#   S E L E C T _ H O S T   (new for prtpkg) #
##############################################
select_host() {
    echo                                                          'Entered select_host' >>$PPDBG
    msg="Enter the name you want your CRUX cell to use; i.e., the"
    msg="${msg} string that the hostname command will return:"
    do_select --aspect 40 --inputbox "$msg" 0 0 "crux" 2> $tmpfile
    HOSTNAME=`/bin/cat $tmpfile`
    echo                                                          "HOSTNAME<$HOSTNAME>" >>$PPDBG
}
########################################################
#   C O M M O N W E A L T H _ O L D   (new for prtpkg) #
########################################################
commonwealth_old() {
    echo                                                     'Entered commonwealth_old' >>$PPDBG
    NEWCOMW='' # A commonwealth exists (set NEWCOMW to false)
```

```
        CWTYPE=`/bin/cat "$ROOT/usr/prtpkg/type" \
                | /usr/bin/tr -d '\012' \
                `
        test    ".$CWTYPE" != '.S' \
             -a ".$CWTYPE" != '.M' \
          && { echo "$ROOT/usr/prtpkg/type contains unexpected data <$CWTYPE>--aborting!"
               exit 32
             }
        if   test $ACTION = 'INSTALL'
        then /bin/ls                              -lah $ROOT/usr/prtpkg/cells/$PRTPKG_CELL >>$PPDBG 2>&1
             if   test    -d $ROOT/usr/prtpkg/cells/$PRTPKG_CELL \
                       -o -L $ROOT/usr/prtpkg/cells/$PRTPKG_CELL
             then echo 'Cannot install into existing PRTPKG_CELL--aborting!'
                  exit 32
             else echo 'This install will add a new boot cell to the existing commonwealth.'
                  CWTYPE='I' # This value is internal to prtpkg_setup only, not for the type file
             fi
        else # update existing cell--determine if this is a boot or chroot cell
             /bin/ls -L $ROOT/usr/prtpkg/cells/$PRTPKG_CELL >/dev/null 2>&1 \
               || { echo -n "Cannot update an installed target cell ($PRTPKG_CELL) lacking "
                    echo    'a /usr/prtpkg/cells directory--aborting!'
                    exit 32
                  }
             /bin/ls -L $ROOT/usr/prtpkg/cells/$PRTPKG_CELL/types >/dev/null 2>&1 \
               || { echo -n "Cannot update an installed target cell ($PRTPKG_CELL) lacking "
                    echo    "a /usr/prtpkg/cells/$PRTPKG_CELL/types file--aborting!"
                    exit 32
                  }
             if   /usr/bin/grep -q '^boot ' $ROOT/usr/prtpkg/cells/$PRTPKG_CELL/types
             then echo                       "This update will process boot cell $PRTPKG_CELL." >>$PPDBG
                  CWTYPE='U' # This value is internal to prtpkg_setup only, not for the type file
             else echo "Existing cell $PRTPKG_CELL is a chroot cell--aborting!"
                  echo 'Use the up_cell script while running in the associated boot cell'
                  echo 'instead of using the prtpkg_setup script while running in the ISO.'
                  exit 32
             fi
        fi
    }
    ########################################################
    #   C O M M O N W E A L T H _ N E W   (new for prtpkg) #
    ########################################################
    commonwealth_new() {
        echo                                                         'Entered commonwealth_new' >>$PPDBG
        NEWCOMW='T' # No existing commonwealth at present (set NEWCOMW to true)
        test $ACTION = 'UPGRADE' \
          && { echo 'Updating without existing commonwealth membership is unsupported--aborting.'
               echo 'Use the cannonical setup script instead of the prtpkg_setup script.'
               exit 32
             } \
          || echo                                       "A new commonwealth will be installed." >>$PPDBG
        test    ! -d "$ROOT/usr/prtpkg" \
             -a ! -L "$ROOT/usr/prtpkg" \
          && /bin/mkdir "$ROOT/usr/prtpkg"
        test ! -f "$ROOT/usr/prtpkg/_under_construction_" \
          && echo $PRTPKG_CELL >"$ROOT/usr/prtpkg/_under_construction_"
        if   test ! -f "$ROOT/usr/prtpkg/type"
        then select_cwtype
        else CWTYPE=`/bin/cat "$ROOT/usr/prtpkg/type" \
                     | /usr/bin/tr -d '\012' \
                     `
             test    ".$CWTYPE" != '.S' \
                  -a ".$CWTYPE" != '.M' \
               && { echo            "$ROOT/usr/prtpkg/type contains <$CWTYPE>--asking again..." >>$PPDBG
                    select_cwtype
                  }
        fi
    }
    ##################################################
    #   S E L E C T _ C W T Y P E   (new for prtpkg) #
    ##################################################
    select_cwtype() { # Determine whether new commonwealth is single- or multiple-prtpkg-platform
        # Returns CWTYPE = 'S' or 'M'
        echo                                                'Entered select_commonwealth' >>$PPDBG
        do_select --menu "Single- or multi-prtpkg-platform commonwealth?" 9 60 2 \
                         "1" "Single (no network filesystem(s)" \
                         "2" "Multi (at least /usr/prtpkg exported using NFS)" 2> $tmpfile
        CWTYPE=`/bin/cat $tmpfile`
        echo                                                            "CWTYPE<$CWTYPE>" >>$PPDBG
        if   test "$CWTYPE" = "1"
        then CWTYPE='S'
        else CWTYPE='M'
             msg="When installing a multi-commonwealth, the following packages in"
             msg="${msg} the opt collection become required:\n"
             msg="${msg}\n  nfs-utils\n"
             msg="${msg}\nand the packages it depends upon:\n"
             msg="${msg}\n  keyutils libevent libnfsidmap libtirpc rpcbind sqlite\n"
             msg="${msg}\nThese packages will be automatically selected for installation"
             msg="${msg} even if you do not specifically request them in the next set of"
             msg="${msg} dialogs."
             msg="{$msg}\nAre you really sure you want to continue?"
             do_select --aspect 5 --yesno "$msg" 0 0
        fi
        echo $CWTYPE >$ROOT/usr/prtpkg/type
        echo                                                            "CWTYPE<$CWTYPE>" >>$PPDBG
    }
    ###########################################
    #   S E L E C T _ C O L L E C T I O N S   #
    ###########################################
    select_collections() {
        echo                                                 'Entered select_collections' >>$PPDBG
        if   test "$ACTION" != "INSTALL"
        then return 0
        fi
        if      test ! -d core \
             || test ! -d opt \
             || test ! -d xorg \
             || test ! -d kernel
        then msg="Package directories (core, opt, xorg or kernel) were not found in"
             msg="$msg $crux_dir. Aborting."
             do_dialog --aspect 50  --msgbox "$msg" 0 0
             exit 1
        fi
        # 2017-02-17 dlcusa: Add any non-core ports prtpkg requires (note opt/openntpd is
        #                    strongly recommended for multi-platform commonwealths but is
        #                    not available via the ISO and so must be installed once the
        #                    cell's portdb has been initialized per the CRUX Handbook.
        REQOPTS='fakeroot' # All commonwealths (add prtpkg once there is a prtpkg package)
        test ".$CWTYPE" = '.M' \
          && { # for multi-platform commonwealths, add these packages:
               REQOPTS="$REQOPTS nfs-utils"
               # and those they depend upon:"
               REQOPTS="$REQOPTS keyutils libevent libnfsidmap libtirpc rpcbind sqlite"
             }
        TITLE="Select collections to install:\n(detailed package selection will follow)"
        do_select --separate-output --checklist "$TITLE" 13 78 6 \
                  core "The core packages (required)" ON \
                  opt "Optional packages (other than prtpkg reqs)" OFF \
                  xorg "X.org packages" OFF 2> $collfile
    }
    ###############################
    #   A S K _ D E T A I L E D   #
    ###############################
    ask_detailed() {
        echo                                                       'Entered ask_detailed' >>$PPDBG
        if   test "$ACTION" != "INSTALL"
        then return 0
        fi
        msg="Do you want the chance to select packages individually?  If you choose no,"
        msg="${msg} all packages in every collection you selected will be installed."
        do_dialog --aspect 50 --defaultno --yesno "$msg" 0 0 \
          && DO_DETAILED="yes"
        echo                                                  "DO_DETAILED<$DO_DETAILED>" >>$PPDBG
```

```
}
#####################################
#   S E L E C T _ P A C K A G E S   #
#####################################
select_packages() {
    echo                                                          'Entered select_packages' >>$PPDBG
    if      test ! -d core \
         || test ! -d opt \
         || test ! -d xorg \
         || test ! -d kernel
    then msg="Package directories (core, opt, xorg, and kernel) were not found"
         msg="${msg} in $crux_dir. Aborting."
         do_dialog --aspect 50 --msgbox "$msg" 0 0
         exit 1
    fi
    if   test "$ACTION" = "INSTALL"
    then select_packages_install
    else select_packages_upgrade
    fi
}
########################################################################
#   S E L E C T _ P A C K A G E S _ I N S T A L L   (new for prtpkg)   #
########################################################################
select_packages_install() {
    echo                                                        'Processing ACTION INSTALL' >>$PPDBG
    if   test "$DO_DETAILED" = "yes"
    then for collection in core opt xorg
         do  presel=`/usr/bin/grep $collection $collfile`
             if   test ".$presel" = ".$collection"
             then checked=ON
             else checked=OFF
             fi
             msg="Searching for packages, please wait..."
             do_dialog --aspect 50 --infobox "$msg" 0 0
             TITLE="Select $collection packages to install:"
             PKG_LIST=`/usr/bin/find $collection -name *.pkg.tar.[bgx]z* \
                                    -printf "%f ($collection) $checked\n" \
                       | /bin/sed 's/.pkg.tar.[^ ]*//g' \
                       | /usr/bin/sort \
                       | /usr/bin/xargs echo ' ' \
                      `
             echo                                                     "PKG_LIST<$PKG_LIST>" >>$PPDBG
             do_select --separate-output --checklist "$TITLE" 19 60 12 \
                       $PKG_LIST 2>> $pkgfile
         done
    else for collection in core opt xorg
         do  presel=`/usr/bin/grep $collection $collfile`
             if   test ".$presel" = ".$collection"
             then /usr/bin/find $collection -name *.pkg.tar.[bgx]z* \
                               -printf "%f\n" \
                  | /bin/sed 's/.pkg.tar.*//g' \
                  | /usr/bin/sort >> $pkgfile
             fi
         done
    fi
    echo                                                                      -n 'pkgfile<' >>$PPDBG
    /bin/cat                                                                       $pkgfile >>$PPDBG
    echo                                                                                '>' >>$PPDBG
    echo                                          "Adding REQOPTS<$REQOPTS> to $pkgfile..." >>$PPDBG
    : >$tmpfile 2>$TMPDIR/error
    for i in $REQOPTS
    do  /bin/ls -1 $PWD/opt/$i* 2>>$TMPDIR/error \
        | /bin/sed -e 's|^.*/||' -e 's/.pkg.tar.[bgxz2]*$//' >>$tmpfile
        echo                                                                  -n 'tmpfile<' >>$PPDBG
        /bin/cat                                                                   $tmpfile >>$PPDBG
        echo                                                                            '>' >>$PPDBG
        if   test -s $TMPDIR/error
        then /bin/cat $TMPDIR/error
             echo "Error attempting to add $i to list of packages to process--aborting!"
             exit 32
        fi
    done
    if   test -s $tmpfile
    then echo                                            'tmpfile<'`/bin/cat $tmpfile`'>' >>$PPDBG
         /bin/cat $tmpfile >>$pkgfile
         /usr/bin/sort $pkgfile \
         | /usr/bin/uniq >$tmpfile
         /bin/cat $tmpfile >$pkgfile
    fi
    echo                                                 'pkgfile<'`/bin/cat $pkgfile`'>' >>$PPDBG
}
########################################################################
#   S E L E C T _ P A C K A G E S _ U P G R A D E   (new for prtpkg)   #
########################################################################
select_packages_upgrade() {
    echo                                                      'Processing ACTION UPGRADE' >>$PPDBG
    msg="Searching for packages, please wait..."
    do_dialog --aspect 50 --infobox "$msg" 0 0
    TITLE="Select packages to upgrade:"
    INSTALLED_PACKAGES=`/usr/bin/pkginfo -r $ROOT -i \
                        | /usr/bin/gawk '{print $1}' \
                       `
    # 2017-08-10 dlcusa: Do not die silently if this occurs:
    echo                                        "INSTALLED_PACKAGES<$INSTALLED_PACKAGES>" >>$PPDBG
    if   test ".$INSTALLED_PACKAGES" = '.'
    then echo 'No installed packages were found in the target system--aborting!'
         exit 32
    fi
    for package in $INSTALLED_PACKAGES
    do  CORE_LIST="$CORE_LIST \
                  `/usr/bin/find core -name ${package}\#*.pkg.tar.[bgx]z* \
                                 -printf '%f (core) ON\n' \
                   | /bin/sed 's/.pkg.tar.[^ ]*/ /g' \
                   | /usr/bin/sort \
                   | /usr/bin/xargs echo ' ' \
                  `"
        echo                                                          "CORE_LIST<$CORE_LIST>" >>$PPDBG
        OPT_LIST="$OPT_LIST \
                 `/usr/bin/find opt -name ${package}\#*.pkg.tar.[bgx]z* \
                                -printf '%f (opt) ON\n' \
                  | /bin/sed 's/.pkg.tar.[^ ]*/ /g' \
                  | /usr/bin/sort \
                  | /usr/bin/xargs echo ' ' \
                 `"
        echo                                                            "OPT_LIST<$OPT_LIST>" >>$PPDBG
        XORG_LIST="$XORG_LIST \
                  `/usr/bin/find xorg -name ${package}\#*.pkg.tar.[bgx]z* \
                                 -printf '%f (xorg) ON\n' \
                   | /bin/sed 's/.pkg.tar.[^ ]*/ /g' \
                   | /usr/bin/sort \
                   | /usr/bin/xargs echo ' ' \
                  `"
        echo                                                          "XORG_LIST<$XORG_LIST>" >>$PPDBG
   done
   do_select --separate-output --checklist "$TITLE" 19 60 12 \
       $CORE_LIST $OPT_LIST $XORG_LIST 2> $pkgfile
}
###########################################
#   C H E C K _ D E P E N D E N C I E S   #
###########################################
check_dependencies() {
    echo                                                     'Entered check_dependencies' >>$PPDBG
    if   test "$ACTION" != "INSTALL"
    then return 0
    fi
    msg="Checking dependencies, please wait..."
    do_dialog --aspect 50 --infobox "$msg" 0 0
    get_missing_deps
    mpwc=`echo "$MISSING_DEPS" \
          | /usr/bin/wc -w \
         `
    echo                                  "MISSING_DEPS contains $mpwc words, PWD<$PWD>" >>$PPDBG
    findout='/tmp/prtpkg/find_results'
    sed1out='/tmp/prtpkg/sed1_results'
    sed2out='/tmp/prtpkg/sed2_results'
    if   test $mpwc -gt 0
    then for package in $MISSING_DEPS
         do  : >$findout 2>$sed1out 3>$sed2out
             MISSING_LIST="$MISSING_LIST \
                          `/usr/bin/find . -name ${package}#*.pkg.tar.[bgx]z* \
                                         -printf '%f %p ON\n' 2>>$findout \
                           | /usr/bin/tee -a $findout \
                           | /bin/sed 's/.pkg.tar.[^ ]*/ /g' 2>>$sed1out \
                           | /usr/bin/tee -a $sed1out \
                           | /bin/sed 's|./|(|;s|/.* O|) O|' 2>>$sed2out \
                           | /usr/bin/tee -a $sed2out \
                           | /usr/bin/sort \
                           | /usr/bin/xargs echo ' ' \
                          `"
             echo                               "For $package, MISSING_LIST<$MISSING_LIST>" >>$PPDBG
             echo                                                             -n 'findout<' >>$PPDBG
             /bin/cat                                                              $findout >>$PPDBG
             echo                                                                       '>' >>$PPDBG
             echo                                                             -n 'sed1out<' >>$PPDBG
             /bin/cat                                                              $sed1out >>$PPDBG
             echo                                                                       '>' >>$PPDBG
             echo                                                             -n 'sed2out<' >>$PPDBG
             /bin/cat                                                              $sed2out >>$PPDBG
             echo                                                                       '>' >>$PPDBG
         done
         echo                                                 "MISSING_LIST<$MISSING_LIST>" >>$PPDBG
         TITLE="The following packages are needed by the ones you selected"
         do_select --separate-output --checklist "$TITLE" 19 60 12 $MISSING_LIST 2>> $pkgfile
    fi
}
#######################################
#   G E T _ M I S S I N G _ D E P S   #
#######################################
get_missing_deps() {
    echo                                                         'Entered get_missing_deps' >>$PPDBG
    needed=""
    toinstall=`/bin/sed 's/\#.*//g' $pkgfile`
    for f in $toinstall
    do  pdeps=`/usr/bin/grep "^$f\:" $depsfile \
               | /bin/sed "s|^$f: ||g" \
              `
        echo                                                     "For f<$f>, pdeps<$pdeps>" >>$PPDBG
        for d in $pdeps
        do  needed="$needed $d"
        done
    done
    echo                                                                  "needed<$needed>" >>$PPDBG
    /bin/sed 's/\#.*//g' $pkgfile \
    | /usr/bin/sort -u > $markedfile
    echo                                                                   -n 'markedfile<' >>$PPDBG
    /bin/cat                                                                    $markedfile >>$PPDBG
    echo                                                                                '>' >>$PPDBG
    echo $needed \
    | /usr/bin/tr ' ' '\n' \
    | /usr/bin/sort -u > $neededfile
    echo                                                                   -n 'neededfile<' >>$PPDBG
    /bin/cat                                                                    $neededfile >>$PPDBG
    echo                                                                                '>' >>$PPDBG
    MISSING_DEPS=`/usr/bin/comm -1 -3 $markedfile $neededfile`
    echo                                                      "MISSING_DEPS<$MISSING_DEPS>" >>$PPDBG
}
#####################
#   C O N F I R M   #
#####################
confirm() {
    echo                                                                  'Entered confirm' >>$PPDBG
    if   test "$ACTION" = "INSTALL"
    then msg="Selected packages will now be installed. Are you sure you want to continue?"
```

```
             do_select --aspect 25 --yesno "$msg" 0 0
        else msg="Selected packages will now be upgraded. Are you sure you want to continue?"
             do_select --aspect 25 --yesno "$msg" 0 0
        fi
    }
    ################################################################
    #   P R E P A R E _ C O M M O N W E A L T H   (new for prtpkg) #
    ################################################################
    prepare_commonwealth() { # Top-level routine to create or update this cell's prtpkg commonwealth
        # This function is invoked following the final confirmation dialog to create or
        # update the commonwealth in which this cell does or will reside.  If the cell's
        # commonwealth pre-existed the execution of this script, the commonwealth will
        # be validated; otherwise, the new commonwealth as specified in the CWTYPE
        # variable will be instantiated.  Either way, the PRTPKG update lock will be
        # obtained to allow the entire commonwealth to wait until this cell's upgrade /
        # install process is completed following the first (re)boot of the cell.
        echo           "Entered prepare_commonwealth CWTYPE<$CWTYPE> PRTPKG_CELL<$PRTPKG_CELL>" >>$PPDBG
        BOOTOS=CRUX-$VERSION
        echo                                                               "BOOTOS<$BOOTOS>" >>$PPDBG
        DA=$ROOT/usr/prtpkg/$BOOTOS # Commonwealth-wide crux Distribution-specific files Anchor
        echo                                                                       "DA<$DA>" >>$PPDBG
        CA=$ROOT/usr/prtpkg/cells/$PRTPKG_CELL # Cell-specific files Anchor
        echo                                                                       "CA<$CA>" >>$PPDBG
        CDA="$CA/CRUX-$VERSION" # Cell-specific crux Distribution-specific files Anchor
        echo                                                                     "CDA<$CDA>" >>$PPDBG
        # Ensure these standard CRUX directories are available in the target rootfs
        for i in root etc/ports usr/ports
        do  /bin/ls -L ${ROOT}/$i >/dev/null 2>&1 \
              || /bin/mkdir -p ${ROOT}/$i
            /bin/ls                                                -lhd ${ROOT}/$i >>$PPDBG 2>&1
        done
        # Get the PRTPKG update lock (create its queue first if necessary)
        LOCKTYPE='E' # Obtain update (exclusive) lock
        LOCKID='PRTPKG'
        LOCKDIR_PRTPKG=$ROOT/usr/prtpkg/$LOCKID # For use by signal_0 lock cleanup
        LOCKDIR="$LOCKDIR_PRTPKG"
        lock_obtain # Request the PRTPKG update lock and wait for it
        signal_set_traps
        echo                 "`date -u` -- $0 (pid $$) now owns this commonwealth exclusively." >>$PPDBG
        # Establish or tweak the $ROOT/usr/prtpkg/cells branch for this host[rootfs]
        /bin/ls -L1 $CA >/dev/null 2>&1 \
          && setup_update_ca \
          || setup_create_ca
        # If necessary, create the prtpkg.txt file for this release
        prtpkgfile=$CDA/prtpkg.txt
        /bin/ls -L1 $prtpkgfile >/dev/null 2>&1 \
          || : > $prtpkgfile
        # Establish or tweak the $ROOT/usr/prtpkg tree branch for this distribution release
        /bin/ls -L $DA >/dev/null 2>&1 \
          && setup_update_da \
          || setup_create_da
        #
        # Archive the ISO package repos into the target cell's
        # /usr/prtpkg/$BOOTOS/packages/pkgs-ISO-$VERSION tree so
        # fallback to the ISO packages is available without
        # needing to boot up the ISO again.
        #
        ISODIR="$ROOT/usr/prtpkg/$BOOTOS/packages/pkgs-ISO-$VERSION"
        /bin/ls -L1 "$ISODIR" >/dev/null 2>&1 \
          || /bin/mkdir -p "$ISODIR"
        ( cd $MEDIA/crux \
            && /bin/cp -a core opt xorg "$ISODIR"
        )
        /bin/ls                                                     -lhR $ISODIR >>$PPDBG 2>&1
    }
    ######################################################
    #   S E T U P _ U P D A T E _ C A   (new for prtpkg) #
    ######################################################
    setup_update_ca() { # There is a cells tree for this host[rootfs]--update anything in need
       echo                                                     'Entered setup_update_ca' >>$PPDBG
       /bin/ls -L $CA/BOOT >/dev/null 2>&1 \
         && { PRTPKG_PREVBOOT=`/bin/cat   $CA/BOOT`
              test ".$PRTPKG_PREVBOOT" = '.' \
                && { echo                                    "$CA/BOOT found empty--aborting!" >>$PPDBG
                     exit 32
                   } \
                || echo                                    "PRTPKG_PREVBOOT<$PRTPKG_PREVBOOT>" >>$PPDBG
            } \
         || { echo                                            "$CA/BOOT not found--aborting!" >>$PPDBG
              exit 32
            }
       /bin/ls -L $CA/MAKE >/dev/null 2>&1 \
         && { PRTPKG_PREVMAKE=`/bin/cat   $CA/MAKE`
              test ".$PRTPKG_PREVMAKE" = '.' \
                && { echo                                    "$CA/MAKE found empty--aborting!" >>$PPDBG
                     exit 32
                   } \
                || echo                                    "PRTPKG_PREVMAKE<$PRTPKG_PREVMAKE>" >>$PPDBG
            } \
         || { echo                                            "$CA/MAKE not found--aborting!" >>$PPDBG
              exit 32
            }
       /bin/ls -L $CA/PORTSU >/dev/null 2>&1 \
         && { PRTPKG_PREVPORTSU=`/bin/cat   $CA/PORTSU`
              test ".$PRTPKG_PREVPORTSU" = '.' \
                && { echo                                  "$CA/PORTSU found empty--aborting!" >>$PPDBG
                     exit 32
                   } \
                || echo                                "PRTPKG_PREVPORTSU<$PRTPKG_PREVPORTSU>" >>$PPDBG
            } \
         || { echo                                          "$CA/PORTSU not found--aborting!" >>$PPDBG
              exit 32
            }
       echo                                                      "Conditional mkdir $CDA..." >>$PPDBG
       /bin/ls -L $CDA >/dev/null 2>&1 \
         || /bin/mkdir -p $CDA
       /bin/ls                                                                -Llhd $CDA >>$PPDBG 2>&1
       echo                                                  "Conditional mkdir $CDA/log..." >>$PPDBG
       /bin/ls -L $CDA/log >/dev/null 2>&1 \
         || /bin/mkdir -p $CDA/log
       /bin/ls                                                            -Llhd $CDA/log >>$PPDBG 2>&1
       echo                                             "Conditional mkdir $CDA/symports..." >>$PPDBG
       /bin/ls -L $CDA/symports >/dev/null 2>&1 \
         || /bin/mkdir -p $CDA/symports
       /bin/ls                                                       -Llhd $CDA/symports >>$PPDBG 2>&1
    }
    ######################################################
    #   S E T U P _ C R E A T E _ C A   (new for prtpkg) #
    ######################################################
    setup_create_ca() { # Create all new $ROOT/usr/prtpkg/cells/host[rootfs] tree
       echo                                                        'Entered setup_create_ca' >>$PPDBG
       /bin/mkdir -p $CA # Create this cell's anchor
       echo "CRUX-$VERSION"  > "$CA/BOOT"
       echo "CRUX-$VERSION"  > "$CA/MAKE"
       echo "CRUX-$VERSION"  > "$CA/PORTSU"
       /bin/mkdir $CDA # Create the cell's anchor for this distribution
       /bin/ls                                                                -Llhd $CDA >>$PPDBG 2>&1
       /bin/mkdir $CDA/log
       /bin/ls                                                            -Llhd $CDA/log >>$PPDBG 2>&1
       /bin/mkdir $CDA/symports
       /bin/ls                                                       -Llhd $CDA/symports >>$PPDBG 2>&1
    }
    ######################################################
    #   S E T U P _ U P D A T E _ D A   (new for prtpkg) #
    ######################################################
    setup_update_da() { # There is a distribution tree for this release--update anything in need
       echo                                                        'Entered setup_update_da' >>$PPDBG
    }
    ######################################################
    #   S E T U P _ C R E A T E _ D A   (new for prtpkg) #
    ######################################################
    setup_create_da() { # There is no tree branch yet for the distribution release--build it
```

```
    echo                                                         'Entered setup_create_da' >>$PPDBG
    /bin/mkdir -p $DA
    /bin/ls                                                                -Llhd $DA >>$PPDBG 2>&1
    /bin/mkdir $DA/MAKE   # Establish this release's MAKE maintenance
                          # serialization queue and to this system grant
                          # exclusive privilege as a floating lock for now
    ts=`/bin/date -u '+%Y%m%d-%T.%NUT'`
    echo $ts 0 >$DA/MAKE/$ts\;E\;$PRTPKG_CELL\;$BOOTOS\;setup
    /bin/mkdir $DA/BOOT   # Establish this release's BOOT maintenance
                          # serialization queue and to this system grant
                          # exclusive privilege as a floating lock for now
    ts=`/bin/date -u '+%Y%m%d-%T.%NUT'`
    echo $ts 0 >$DA/BOOT/$ts\;E\;$PRTPKG_CELL\;$BOOTOS\;setup
    /bin/mkdir $DA/builders       # Define the systems that may MAKE for this release
    : >$DA/builders/$PRTPKG_CELL  # and posit this system as the sole builder
    /bin/mkdir $DA/porters        # Define the systems that may PORTSU for this release
    : >$DA/porters/$PRTPKG_CELL   # and posit this system as the sole porter
    /bin/mkdir $DA/distfiles      # Establish this release's distfiles tree.
                                  # prtpkg creates package subdirectories to
                                  # prevent any possibility of collisions and
                                  # to document whence the distfile (not
                                  # always intuitive).  Consider making this
                                  # a symlink to HDD-backed storage if the
                                  # commonwealth's /usr/prtpkg resides in
                                  # SSD-backed storage.
    /bin/mkdir $DA/work           # Establish this release's package building
                                  # area.  Note prtpkg by default archives
                                  # compressed build work directories in
                                  # subtrees that have subpaths in the form of
                                  # build-ID/try-ID/package, and if the cell
                                  # has both SSD and HDD storage, work should
                                  # have symlinks to the batch-ID directories
                                  # backed by HDD storage.  Also note prtpkg
                                  # requires the use of fakeroot, so if hasn't
                                  # been installed yet, the following command
                                  # lists based on /etc/passwd and /etc/group
                                  # will need to be rerun after it has been
                                  # installed.
    test -f $ROOT/etc/passwd \
      && /usr/bin/grep -q '^pkgmk:'  $ROOT/etc/passwd \
      && /bin/chown pkgmk  $DA/work
    test -f $ROOT/etc/group \
      && /usr/bin/grep -q '^nobody:' $ROOT/etc/group \
      && /bin/chgrp nobody $DA/work
    prtpkgtxt=$CDA/prtpkg.txt
    : >$prtpkgtxt
}
#############################
#   P R O G R E S S B A R   #
#############################
progressbar() {
    echo                                                          'Entered progressbar' >>$PPDBG
    echo "XXX"
    /usr/bin/expr $COUNT '*' 100 / $TOTAL
    echo "\n$*"
    echo "XXX"
    COUNT=`/usr/bin/expr $COUNT \+ 1`
}
#######################################
#   I N S T A L L _ P A C K A G E S   #
#######################################
install_packages() {
    echo                            "Entered install_packages with PWD<`pwd`> ROOT<$ROOT>" >>$PPDBG
    # 2017-08-14 dlcusa: Show the state of the root directory and its var/lig/pkg directory
    echo                                                                     $ROOT: >>$PPDBG
    /bin/ls                                                            -lah $ROOT >>$PPDBG 2>&1
    if   test   -d     $ROOT/var/lib/pkg
    then echo                                                 $ROOT/var/lib/pkg before: >>$PPDBG
         /bin/ls                                         -lah $ROOT/var/lib/pkg >>$PPDBG 2>&1
    fi
    if   test ! -d     $ROOT/var/lib/pkg
```

```
then /bin/mkdir -p $ROOT/var/lib/pkg
     /bin/touch    $ROOT/var/lib/pkg/db
fi
if   test   -d     $ROOT/var/lib/pkg/rejected
then /bin/rm   -rf $ROOT/var/lib/pkg/rejected
fi
# 2017-08-14 dlcusa: Now that /mnt/var/lib/pkg exists, put a symlink for
#                    prtpkg.txt in there that redirects into $CDA
ln -s ../../../usr/prtpkg/cells/$PRTPKG_CELL/CRUX-$VERSION/prtpkg.txt \
      /mnt/var/lib/pkg/prtpkg.txt
# 2017-08-14 dlcusa: Show the amended state of the var/lig/pkg directory
echo                                                           $ROOT/var/lib/pkg after: >>$PPDBG
/bin/ls                                                     -lah $ROOT/var/lib/pkg >>$PPDBG 2>&1
if   test "$ACTION" = "INSTALL"
then PKGARGS=""
else # We use -f here since we want to avoid pkgadd conflicts.
     # Unwanted/Unexpected conflicts could arise if files are
     # moved from one package to another, or if the user added
     # the files himself. Instead of failing the whole upgrade
     # we force the upgrade. This should be fairly safe and it
     # will probably help to avoid some "semi-bogus" errors from
     # pkgadd. The rules in /etc/pkgadd.conf will still be used.
     PKGARGS="-f -u"
fi
( # Do all this in a subshell:
    # Log header
    echo "Log  ($logfile)" > $logfile
    echo "-----------------------------------------------------------" >> $logfile
    # Install packages
    KERNEL=./kernel/linux-*.tar.xz
    KERNEL_VERSION=`/usr/bin/basename $KERNEL .tar.xz \
                  | /bin/sed "s/linux-//" \
                  `
    TOTAL=`/bin/cat $pkgfile \
         | /usr/bin/wc -l \
         `
    TOTAL=`/usr/bin/expr $TOTAL \+ 1`
    COUNT=0
    ERRORS=0
    for FILE in `/bin/cat $pkgfile`
    do  progressbar "Installing $FILE..."
        echo -n "Installing $FILE....." >> $logfile
        PKG_FILE=`/usr/bin/find . -name $FILE.pkg.tar.[bgx]z*`
        echo                                               "PKGFILE<$PKGFILE>" >>$PPDBG
        # 2017-02-24 dlcusa: Capture pkgadd command and its status code for
        #                    prtpkg.txt record:
        pp_cmd="/usr/bin/pkgadd -r $ROOT $PKGARGS $PKG_FILE"
        # 2017-02-24 dlcusa: Add this package to the target system and latch its status:
        $pp_cmd > $tmpfile 2>&1 ; rc=$?
        # 2017-02-24 dlcusa: Construct prtpkg.txt record regarding this package
        #                    containing these fields separated by single tab
        #                    characters:
        #  1:timestamp  e.g., 20170130-15:07:58UT
        #  2:command-rc e.g., 0
        #  3:prtpkg-cmd e.g., U
        #  4:package    e.g., firefox
        #  5:version    e.g., 51.0-1
        #  6:collection e.g., opt
        #  7:build-ID   e.g., pkgsb3
        #  8:batch-ID   e.g., 00031
        #  9:try-ID     e.g., 06
        # 10:pwd        e.g., /var/log/pkgbuild/_/00031/06
        # 11:command    e.g., prt-cache update -if -kw -fr -ns --install-scripts -f firefox
        #
        pp_pkg=`echo $FILE \
               | /usr/bin/gawk -F \# '{print $1}' \
               `
        pp_ver=`echo $FILE \
               | /usr/bin/gawk -F \# '{print $2}' \
               `
        if   test -f core/$PKG_FILE
        then pp_col=core
        elif test -f  opt/$PKG_FILE
        then pp_col=opt
        else pp_col=xorg
        fi
        pp_bld=pkgs-ISO-$VERSION
        pp_bat=00000
        pp_try=00
        pp_mid="a      $pp_pkg       $pp_ver       $pp_col       $pp_bld       $pp_bat       $pp_try"
        # 2017-02-24 dlcusa: Append the prtpkg.txt record regarding this package
        echo "`/bin/date -u '+%Y%m%d-%TUT'`       $rc           $pp_mid       `pwd`         $pp_cmd" \
            >>$prtpkgtxt 2>&1
        if   test $rc = 0
        then echo "OK" >> $logfile
        else ERRORS=`/usr/bin/expr $ERRORS \+ 1`
             echo "ERROR" >> $logfile
             echo "" >> $logfile
             /bin/cat $tmpfile >> $logfile
             echo "" >> $logfile
        fi
    done
    # Install kernel
    if   test ! -d $ROOT/usr/src/linux-$KERNEL_VERSION
    then progressbar "Installing `/usr/bin/basename $KERNEL .tar.xz`..."
         echo -n "Installing `/usr/bin/basename $KERNEL .tar.xz`....." >> $logfile
         ( # Do this in a subshell piped into a tmpfile:
             set -e
             /bin/tar -C $ROOT/usr/src -xJf $KERNEL
             /bin/cp -f ./kernel/linux-$KERNEL_VERSION.defconfig \
                  $ROOT/usr/src/linux-$KERNEL_VERSION/.config
             /bin/chown -R root.root $ROOT/usr/src/linux-$KERNEL_VERSION
             /bin/chmod -R go-w $ROOT/usr/src/linux-$KERNEL_VERSION
           # shopt -s nullglob # 2017-02-17 dlcusa: Code around this bashism:
             # modified to be filename-agnostic
             #for patch in ./kernel/linux-$KERNEL_VERSION-*.patch; do
             for patch in ./kernel/*patch
             do  test ".$patch" != '../kernel/*patch' \
                   && { /usr/bin/patch -s -d $ROOT/usr/src/linux-$KERNEL_VERSION -p1 < $patch
                        /bin/cp -p $patch $ROOT/usr/src/
                      }
             done
             if   test ! -d $ROOT/lib/modules/$KERNEL_VERSION
             then /bin/mkdir  -p $ROOT/lib/modules/$KERNEL_VERSION
                  /sbin/depmod -b $ROOT -a $KERNEL_VERSION
             fi
         ) > $tmpfile 2>&1
         if   test $? = 0
         then echo "OK" >> $logfile
         else ERRORS=`expr $ERRORS \+ 1`
              echo "ERROR" >> $logfile
              echo "" >> $logfile
              /bin/cat $tmpfile >> $logfile
              echo "" >> $logfile
         fi
    else echo "Directory $ROOT/usr/src/linux-$KERNEL_VERSION already exists." >> $logfile
         echo "Assuming linux-$KERNEL_VERSION is already installed." >> $logfile
    fi
    # Log footer
    echo "----------------------------------------------------------" >> $logfile
    echo "$ERRORS error(s) found" >> $logfile
    /bin/cat $logfile > $tmpfile
    echo "" > $logfile
    if   test "$ERRORS" = "0"
    then echo "$ACTION COMPLETED SUCCESSFULLY!" >> $logfile
    else echo "$ACTION FAILED!" >> $logfile
    fi
    echo "" >> $logfile
    echo "" >> $logfile
    /bin/cat $tmpfile >> $logfile
) | do_dialog --title " Please wait " --gauge "" 8 60 0
# Show log
```

```
    do_dialog --exit-label "OK" --textbox $logfile 19 68
    # 2017-02-24 dlcusa: Preserve the setup logfile:
    /bin/cp -p $logfile $PRTPKG_SYSDIR/$PRTPKG_BOOT.log
}
###############
#   M A I N   #
###############
main() {
    echo                                                                  'Entered main' >>$PPDBG
    welcome ; echo                                         'Returned to main from welcome' >>$PPDBG
    select_action ; echo                             'Returned to main from select_action' >>$PPDBG
    # 2017-08-10 dlcusa: Determine target root filesys and viable scenario, or perish
    get_scenario ; echo                               'Returned to main from get_scenario' >>$PPDBG
    select_collections ; echo                   'Returned to main from select_collections' >>$PPDBG
    ask_detailed ; echo                               'Returned to main from ask_detailed' >>$PPDBG
    select_packages ; echo                         'Returned to main from select_packages' >>$PPDBG
    confirm ; echo                                         'Returned to main from confirm' >>$PPDBG
    # 2017-08-11 dlcusa: Setup commonwealth as needed for upgrade/install
    prepare_commonwealth ; echo               'Returned to main from prepare_commonwealth' >>$PPDBG
    # 2017-08-11 dlcusa: Development M not ready to continue yet
    check_dependencies ; echo                   'Returned to main from check_dependencies' >>$PPDBG
    echo                                                                   -n 'pkgfile<' >>$PPDBG
    /bin/cat                                                           $pkgfile >>$PPDBG 2>&1
    echo                                                                             '>' >>$PPDBG
    if   test "$ACTION" = "UPGRADE"
    then # 2017-02-24 dlcusa: The setup-helper script has been merged into this
         #                    script as functions
         ( setup_helper $ROOT > $helperlogfile 2>&1 ) \
         | do_dialog --title " Please wait [3.2 -> 3.3 check]" --gauge "" 8 60 0
    fi
    install_packages ; echo                       'Returned to main from install_packages' >>$PPDBG
    # one-time fix for .inactive ports files
    if   test -f $ROOT/etc/ports/contrib.rsync.inactive \
           -a -f $ROOT/etc/ports/contrib.rsync
    then /bin/mkdir -p $ROOT/var/lib/pkg/rejected/etc/ports
         /bin/mv -f $ROOT/etc/ports/contrib.rsync.inactive \
              $ROOT/var/lib/pkg/rejected/etc/ports/contrib.rsync
    fi
    if   test -f $ROOT/etc/ports/compat-32.rsync.inactive \
           -a -f $ROOT/etc/ports/compat-32.rsync
    then /bin/mkdir -p $ROOT/var/lib/pkg/rejected/etc/ports
         /bin/mv -f $ROOT/etc/ports/compat-32.rsync.inactive \
              $ROOT/var/lib/pkg/rejected/etc/ports/compat-32.rsync
    fi
    /bin/cat $helperlogfile 2> /dev/null
  # test $CWTYPE = 'M' \
  #   && LOCKID='PRTPKG'
  #      LOCKDIR="$LOCKDIR_PRTPKG"
  #      lock_freeup
  #      exit 128
    LOCKID='PRTPKG'
    LOCKDIR="$LOCKDIR_PRTPKG"
    lock_dump 'QRC'
    # Float PRTPKG update lock (assume after cell is booted to complete)
    lock_giveup ; echo                                 'Returned to main from lock_giveup' >>$PPDBG
}
###################################
#   M A I N   L I N E   C O D E   #
###################################
set +xv # tracing breaks dialog, so don't use it--instead record internal state
        # via stdout and stderr redirections to $PPDBG
TMPDIR=/tmp/prtpkg  # 2017-08-15 dlcusa - Define /tmp directory for prtpkg temporary files
if   test ! -d $TMPDIR
then /bin/mkdir -p $TMPDIR \
       || { echo "mkdir for $TMPDIR failed code $?--aborting"
            exit 32
          }
fi
PPDBG=$TMPDIR/setup.debug ; export PPDBG # 2017-08-15 dlcusa - For all ISO prtpkg_setup activity
echo                                                                'Entered main line code' >>$PPDBG
# 2017-08-11 dlcusa: Log start to $PPDBG:
```

```
echo '--------------------------------------------------------------------------------------' >>$PPDBG
echo                                  "`/bin/date -u` -- Running prtpkg_setup (\$0<$0>)..." >>$PPDBG
/bin/ls                                     -lh $0 /usr/bin/prtpkg_locking.dash.source >>$PPDBG 2>&1
/usr/bin/env \
| /usr/bin/sort                                                                        >>$PPDBG 2>&1
/bin/ls                                                                    -lh $TMPDIR >>$PPDBG 2>&1
# 2017-08-11 dlcusa: Source the locking functions or abort immediately:
/bin/ls -L   /usr/bin/prtpkg_locking.dash.source >>/dev/null 2>&1 \
  && .       /usr/bin/prtpkg_locking.dash.source \
  || { echo '/usr/bin/prtpkg_locking.dash.source was not found--aborting!'
       exit 32
     }
type lock_dump >>/dev/null 2>&1 \
  && echo                  "Source of /usr/bin/prtpkg_locking.dash.source was successful" >>$PPDBG \
  || { echo "Source of /usr/bin/prtpkg_locking.dash.source failed (type rc $?)--aborting!"
       exit 32
     }
# 2017-02-17 dlcusa:  Define global signal handling variables:
signal="" # signal is false -- will be set to a signal name
          #        (becoming true) when a signal is trapped
tmpfile=$TMPDIR/tmp.$$
collfile=$TMPDIR/collections.$$
pkgfile=$TMPDIR/packages.$$
logfile=$TMPDIR/log.$$
helperlogfile=$TMPDIR/log-helper.$$
neededfile=$TMPDIR/needed.$$
markedfile=$TMPDIR/marked.$$
# 2017-11-20 dlcusa: Verify the PPHOST environment variable was properly defined:
test ".$PPHOST" = '.' \
  && { echo "The PPHOST environment variable does not appear to have been properly defined."
       exit 32
     }
# 2017-11-20 dlcusa: Verify the /_/PP/$PPHOST/PPvars file was created and source it if so:
test -f /_/PP/$PPHOST/PPvars \
  && .  /_/PP/$PPHOST/PPvars \
  || { echo "Did not find the /_/PP/$PPHOST/PPvars file--aborting"
       exit 32
     }
/usr/bin/grep -q "^/dev/$PPTARD /_/$PPTARL " /proc/mounts \
  && { echo "The target root filesystem mounted on $PPTARM is [$PPTARL] on /dev/$PPTARD"
       read -p "Is this correct? Enter 'Y' is so to proceed or anything else to abort:  " ASK
       if   test ".$ASK" != '.Y'
       then exit 32
       fi
     }
ROOT="$PPTARM"
# 2017-11-20 dlcusa: Verify prtpkg_presetup defined the MEDIA environment variable:
test -f $MEDIA/crux-media \
  || { echo "The MEDIA environment variable does not appear to have been properly defined."
       exit 32
     }
/usr/bin/grep -q "prefix=*" /proc/cmdline
if   test $? -eq 0
then for opt in `/bin/cat /proc/cmdline`
     do  echo $opt \
         | /usr/bin/grep -q "prefix="
         if   test $? -eq 0
         then p=`echo $opt \
                 | /usr/bin/cut -d= -f2 \
                `
              crux_dir=$MEDIA/${p}/crux
         fi
     done
else crux_dir=$MEDIA/crux
fi
# Detailed selection of packages
DO_DETAILED="no"
MISSINGDEPS=""
test "$1" != "" \
  && crux_dir=$1
depsfile=$crux_dir/setup.dependencies
```

```
if   test -d $crux_dir
then cd $crux_dir
else do_dialog --aspect 50 --msgbox "Directory $crux_dir not found. Aborting." 0 0
     exit 1
fi
main
# 2017-08-15 dlcusa - Ensure the /tmp files are cleaned up when this process terminates normally:
/bin/rm -f $tmpfile $pkgfile $collfile $neededfile $markedfile
# End of file
```

7.7.26. prtpkg_shortcuts.bash.source

```
#!/bin/bash
# CCIA_0.0    PP-stick base/prtpkg_shortcuts.bash.source ..6....:....7....:....8....:....9....:....0
# ****************************************************************************************************
# * Copyright © 2015-2018 David L. Craig <mailto:dlc.usa@gmail.com>
# * Availibility subject to the terms and conditions contained in the file named
# * DLC_Copyright_License_and_Warranty_Information.txt included with this file or available at
# * http://dlc.casita.net/~dlc/DLC_Copyright_License_and_Warranty_Information.txt.
# ****************************************************************************************************
# Changes: [replace this part with an ordered list of applied CCI change ordinals; e.g., 1 5 13]
# YYYYMMDD ..USERID.. ..............Description of the Non-CCI-Changing Customization..............
#----------------------------------------------------------------------------------------------------
# This file is designed to be sourced from bash or dash sessions within a CRUX maintenance
# environment to establish command aliases, environment variables, etc. in a quick and dirty
# profile-ish manner.
#----------------------------------------------------------------------------------------------------
set -o vi
alias ls='/bin/ls'
alias la='/bin/ls -FaC'
alias ll='/bin/ls -lah'
alias lt='/bin/ls -laht'
alias df='/bin/df -alHT'
alias p='/bin/ps -A -F -w -H'
alias j='jobs -l'
alias h='history | tail -10'
```

7.7.27. var.lib.pkg.prt-get.aliases

```
j2sdk: j2re
j2sdk: jre
jdk: jre
openmotif: lesstif
postfix: sendmail
exim: sendmail
qmail: sendmail
masqmail: sendmail
xorg: x11
```

8. Command Information: Help, Prologs, Sample Outputs

The sample reports are edited with cherry-picked lines and column widths have been shortened to get them to fit better in the pages. They are presented to give you a flavor of what `prtpkg` has to offer. *Some* of the details of existing outputs have been edited for what they should soon look like, mostly the directory/file reorganization, and some of the latest enhancements are not showing in these samples. The point: the `prtpkg` package is *not* vaporware, but **is** work in progress. Really, everything at the moment should be marked [in transition to data reorg].

8.1. Output of Commands (or the *cat* of a `file`)

8.1.1. prtpkg h [contains TODO items]

```
prtpkg is the heart of a unified CRUX package maintenance and organization
toolset that logs maintenance activity in a prtpkg.txt file (a browsable log
that uses tabs as field separators).  That data, along with /var/lib/pkg/db,
can be processed by the prtpkginfo command to produce txt and log files of
installed ports sorted by package and by collection (log files are formatted
txt records without tabs and producing column alignment of the data).  Another
major feature of the toolset is its support for easily identifying how package
files were built, e.g., CFLAGS and all other environment variables at build
time, build uname -a and other system (and hypervisor) info, package roster,
tool configurations, port tree snapshots; i.e., all package metadata that can
be really useful in diagnosing strange package builds and behaviors.  Also,
the prtpkgbatch command (a driver for all commands, not just prtpkg commands,
needed to perform any given package maintenance task as defined in an input
.prtpkg file) by default archives compressed build work directory trees using
backgrounded nohup subprocesses that execute while subsequent prtpkg
subcommands in the batch are processed.

Of necessity, only the subset of canonical CRUX tools that build ports one
package per command or install or remove packages one package per command are
supported, but with some prohibitions (listed below), the rest can be invoked
outside of the prtpkg command to provide machine-readable information that can
be easily converted into .prtpkg files.

NOTE: Rather than require changes in the core system maintenance packages, it
      is thought better to hide the normal ports, prt-get, prt-cache, pkgadd,
      pkgrm, and pkgmk binaries on a prtpkg-maintained system and install in
      their places front-ending executables that filter out with appropriate
      explanation the verboten uses listed below before transferring control to
      the original binaries via execve(2) or the shell built-in exec command as
      appropriate.

      NEVER allow these command types to run in a prtpkg-maintained system
      UNDER ANY CIRCUMSTANCES:
        prt-get without --cache (always use prt-cache except for prt-get cache)
        prt-cache depinst without --test
        prt-cache grpinst without --test
        prt-cache sysup   without --test
      NEVER allow these command types to run in a prtpkg-maintained system
      UNLESS they have been forked within a prtpkg process tree:
        pkgadd
        pkgmk
        pkgrm
        prt-cache install without --test
        prt-cache remove  without --test
        prt-cache update  without --test
        ports -u
      Otherwise, you will have to manually update the prtpkg data files, a
      likely tedious, error-prone, and potentially impossible undertaking.

This version (of the new version) supports the pkg{mk,add,rm}, override, and
test extensions.

TODO: * Finish removing dlcisms and cleaning up the code and doc for the first
        initial release (port or maybe a git repo).
      * Implement TODO prtpkg subcommand TYPEs (see SYNTAX for details),
        including implementing build.conf and deploy.conf file handling.
      * Still a lot of development work to be done on implementing overrides.
      * The front-ending ports, prt-get, prt-cache, pkgmk, pkgadd, and pkgrm
        modules (or scripts) need to be developed and incorporated.
      * Implement prtpkg.conf file of base override definition sets and any
        other tweaks associated with particular porter and builder systems,
        special circumstance, etc.--perhaps supporting environment redefinition
        as part of prtpkg initialization.
      * Add the kernel .config file to the build environment snapshot.
      * Create a script to massage prt-cache sysup|depinst|grpinst –-test
        output into a new .prtpkg file.

For more help, run prtpkg h all|syntax|global|prt|pkg
```

8.1.2. prtpkg h syntax [contains TODO items]

```
SYNTAX:   prtpkg  TYPE  PACKAGE  BATCH  TRY  [OVERRIDES]
 where:
  TYPE of operation is <|>|B|D|E|I|M|P|R|S|U|a|b|c|d|e|f|h|r|s:
  Note:  -h and --help are the same as a naked h.
   TYPE  long-name-----     commands-invoked-or-other-description------------
    <    pre-install        [n/a--run the ports's pre-install script]     TODO
    >    post-install       [n/a--run the ports's post-install script]    TODO
    B    rebuild            pkgmk -f with -i or -u -- invokes pkgadd
    D    demote             [n/a—-move files: SSD>HDD>USB>offsite>null]   TODO
    E    establish          [n/a—-add a CRUX release]                     TODO
    I    install            prt-cache install      -- invokes pkg{mk,add}
    M    migrate            [n/a—-upgrade a CRUX release]                 TODO
    P    promote            [n/a—-move files: SSD<HDD<USB<offsite]        TODO
    R    remove             prt-cache remove       -- invokes pkgrm
    S    make_signature     signify                               CRUX-3.3-TODO
    U    update             prt-cache update       -- invokes pkg{mk,add}
    a    add_package        pkgadd                                        TODO
    b    build_package      pkgmk without -i -f -u -d -uf -um (just make) TODO
    c    clean_package      pkgmk -c                                      TODO
    d    download           pkgmk -do                                     TODO
    e    extract            pkgmk -eo                                     TODO
    f    make_footprint     pkgmk -uf                                     TODO
    h    help               [specify one or none: all|syntax|global|pkg|prt]
    r    remove_package     pkgrm
    s    make_md5sum        pkgmk -um                                     TODO
  PACKAGE is the name of the CRUX package to be processed.  Note: prtpkg
    does not permit multiple packages to be specified for any type of
    operation--each package must have its own prtpkg subcommand invocation.
  BATCH defines a chronological ordinal number starting with zero with
    leading zeros identifying the package maintenance task this prt-get
    invocation is supporting (see prtpkgbatch for further information).
    The new version of prtpkg uses a 5-digit number with a leading
    underscore (indicating it's a new version number), while the old
    version (which had no prtpkgbatch tool) accepts a naked 3-digit number
    (the underscore will be deprecated when support for the old version,
    never released into the wild, is dropped, hopefully before the first
    release).
  TRY is a 2-digit chronological ordinal number starting with zero for the
    attempt being made to complete the maintenance task identified by the
    BATCH argument, with a leading zero as needed.  The old version allowed
    alphanumeric values; e.g., t1, t23.
  OVERRIDES is a string of semi-colon-separated parameters in the
    form [!]OVERRIDE, where the optional ! causes disabling instead
    enabling, and OVERRIDE is one of the following as is pertinent
    to the TYPE of operation requested (most are currently TODO):
      [global]:  _if  _im  _r=  _uf  _um  redo test
      pkgadd:    acd= acf= af   ar=  au
      pkgmk:     mc   mcd= mcf= mcm  md   mdo  meo  mf   mi
                 mif  mim  min  mkw  mns  mu   mud  muf  mum
      pkgrm:     rr=
      prt-cache: paa= pca= pcd= pcf= pcp= pcs= pf   pfr  pi
                 pif  pig= pim  pins pkw  pma= pns  pnsd ppos
                 ppre pr=  pra= pt=  puf  pum
    The test override can be used to show how overrides will affect
    generated prt-cache, pkgmk, pkgadd, and pkgrm commands to support
    developing the desired prtpkg command before committing to its
    execution.
```

8.1.3. prtpkg h global [contains TODO items]

```
GLOBAL OVERRIDES sorted alphabetically within groups:
  _if : use this specification to set mif  and pif           [TODO]
  _im : use this specification to set mim  and pim           [TODO]
  _r= : use this specification to set ar=  and rr=           [TODO]
  _uf : use this specification to set muf  and puf           [TODO]
  _um : use this specification to set mum  and pum           [TODO]
  redo: force execution of this prtpkg command even if was successfully
        invoked earlier in any attempt to process this batch
  test: report what base default flags would be generated and what runnable
        prt-cache, pkgmk, pkgadd, or pkgrm command would be produced by
        any overrides in this prtpkg command, but do not execute that
        command or any others that support it (e.g., mkdir), only report
        the commands (or groups of commands) that would be run if test was
        not invoked.  While not intended for use in a batch file, this
        override can be applied globally to all batch commands in a
        prtpkgbatch input file via the prtpkgbatch command line test
        argument.  Note: This override is completely unrelated to the
        prt-cache --test flag which requires prt-cache be run to perform
        the test.
```

8.1.4. prtpkg h prt [entirely TODO items]

```
PRT-CACHE OVERRIDES (TYPEs I, U, R) sorted alphabetically within groups: -----------------
Note: prtpkg only permits flags in the aargs, margs, and rargs
      strings that have no equivalent prt-cache flags in order
      to simplify prtpkg's flag processing.
  paa=: Supply "--aargs=" string for any pkgadd invocation     (--aargs="flg[ flg]...")
          pkgadd flags not also supported as prt-cache flags:
            Specify alternative installation root                 (-r, --root <path>)
            Upgrade package with the same name                    (-u, --upgrade)
              Note: prt-cache update specifies --aargs=-u
                    automatically, and that cannot be disabled
                    (consider running pkgmk without -f or -i
                    instead of prt-cache)
          Unsupported pkgadd options (not permitted in prtpkg):   unsupported:
            Print help and exit                                   (-h, --help)
            Print version and exit                                (-v, --version)
  pca=: Append   string to this run's prt-cache configuration  (--config-append=string)
  pcd=: /usr/prtpkg subdirectory housing prt-get.conf          (uses --config=)
  pcf=: Supply alternative configuration file to prt-cache     (--config=filespec)
  pcp=: Prepend  string to this run's prt-cache configuration  (--config-prepend=string)
  pcs=: Override string in this run's prt-cache configuration  (--config-append=string)
  pf  : Force install by pkgadd                                (-f, -i, --aargs=-f)
  pfr : Force rebuild by pkgmk                                 (-fr, --margs=-f)
  pi  : Force install by pkgadd                                (-f, -i, --aargs=-f)
  pif : Ignore footprint by pkgmk                              (-if, --margs=-if)
  pig=: Prevent installation of listed dependencies            (--ignore=pkg[,pkg]...)
  pim : Ignore md5sum by pkgmk                                 (-im, --margs=-im)
  pins: Enable|disable the ppre and ppos overrides             (--pre-install,
                                                                --post-install,
                                                                --install-scripts)
  pkw : Prevent removal of the work directory by pkgmk         (-kw, --margs=-kw)
  pma=: Supply "--margs=" string for any pkgmk invocation      (--margs="flg[ flg]...")
          pkgmk flags not also supported as prt-cache flags:
            Remove package and downloaded files                   (-c, --clean)
            Do not build, only check md5sum                       (-cm, --check-md5sum)
            Download any missing distfiles before building        (-d, --download)
              Note: prt-cache sets --margs=-d automatically,
                    and that cannot be disabled (consider -do)
            Only download missing distfiles                       (-do, --download-only)
            Only extract distfiles into work directory            (-eo, --extract-only)
            Ignore new files in a footprint mismatch              (-in, --ignore-new)
            Only check for up-to-date                             (-utd, --up-to-date)
          Unsupported pkgmk options (not permitted in prtpkg):    unsupported:
            Search for and build packages recursively             (-r, --recursive)
            Print help and exit                                   (-h, --help)
            Print version and exit                                (-v, --version)
  pns : Prevent invocation of the strip command by pkgmk       (-ns, --margs=-ns)
  pnsd: Don't parse the default configuration file             (--no-std-config)
  ppos: Invoke post-install script after pkgadd in prt-cache   (--post-install,
                                                                --install-scripts)
  ppre: Invoke pre-install script before pkgmk in prt-cache    (--pre-install,
                                                                --install-scripts)
  pr= : Supply alternative installation root directory         (--install-root=dirspec)
  pra=: Supply "--rargs=" string for any pkgrm invocation      (--rargs="flg[ flg]...")
          pkgmk flags not also supported as prt-cache flags:
            Specify alternative installation root directory       (-r, --root <path>)
          Unsupported pkgrm options (not permitted in prtpkg):    unsupported:
            Print help and exit                                   (-h, --help)
            Print version and exit                                (-v, --version)
  puf : Update footprint by pkgmk                              (-uf, --margs=-uf)
  pum : Update md5sum by pkgmk                                 (-um, --margs=-um)

Unsupported prt-cache options (not permitted in prtpkg): --------------------------------
        Modifies scope of diff, quickdiff, and dependent       (--all)
        Use cache file for this run (always used by prtpkg)    (--cache)
        Modifies output of listinst                        (--depsort)
        Write build output to log (always used by prtpkg)  (--log)
        Show path info for search, dsearch, list, and depends  (--path)
        Modifies output of sysup                           (--nodeps)
        Modifies scope for diff, quickdiff, and sysup      (--prefer-higher, -ph)
        Modifies output of dependent                       (--recursive)
        Use filter, search patterns as regular expression  (--regex)
        Override the 'prefer-higher' option                (--strict-diff, -sd)
        Dry run, don't actually install anything           (--test)
        Modifies output of dependent                       (--tree)
        Verbose and more verbose for search and list       (-v, -vv)

Unsupported prt-cache commands (not permitted in prtpkg): ------------------------------
NOTE: depinst, grpinst, and sysup must NEVER be run in a
      prtpkg-maintained system UNLESS defanged using the
      --test option.  Also, install, remove, and update must
      NEVER be run outside of prtpkg on a prtpkg-maintained
      system.
  cache      cat        current    dependent  depends    depinst    deptree
  diff       dsearch    dumpconfig dup        edit       fsearch    grpinst
  help       info       isinst     list       listinst   listlocked listorphans
  lock       ls         printf     quickdep   quickdiff  readme     search
  sysup      unlock     version
```

8.1.5. prtpkg h pkg [entirely TODO items]

```
PKGMK OVERRIDES (TYPEs B, I, U, m) sorted alphabetically within groups: -----------------
  mc  : Remove package and downloaded files                 (-c, --clean)
  mcd=: /usr/prtpkg subdirectory housing pkgmk.conf         (uses --config-file)
  mcf=: Use alternative configuration file                  (-cf, --config-file <file>)
  mcm : Do not build, only check md5sum                     (-cm, --check-md5sum)
  md  : Download any missing distfiles before building      (-d, --download)
  mdo : Only download missing distfiles                     (-do, --download-only)
  meo : Only extract distfiles into work directory          (-eo, --extract-only)
  mf  : Build package even if it appears to be up to date   (-f, --force)
  mi  : Build and install package                           (-i, --install)
  mif : Build package without checking footprint            (-if, --ignore-footprint)
  mim : Build package without checking md5sum               (-im, --ignore-md5sum)
  min : Ignore new files in a footprint mismatch            (-in, --ignore-new)
  mkw : Keep temporary working directory                    (-kw, --keep-work)
  mns : Do not strip executable binaries or libraries       (-ns, --no-strip)
  mu  : Build and install package (as upgrade)              (-u, --upgrade)
  mud : Only check for up-to-date                           (-utd, --up-to-date)
  muf : Only update previously built package file's footprint  (-uf, --update-footprint)
  mum : Only update .md5sum file in the port directory      (-um, --update-md5sum)
Unsupported pkgmk options (not permitted in prtpkg):        unsupported:
Note: NEVER allow pkgmk to run outside of prtpkg in a
      prtpkg-maintained system!
        Print help and exit                                 (-h, --help)
        Search for and build packages recursively           (-r, --recursive)
        Print version and exit                              (-v, --version)

PKGADD OVERRIDES (TYPEs B, I, U, a) sorted alphabetically within groups: ----------------
  acd=: /usr/prtpkg subdirectory housing pkgadd.conf        (unsupported by pkgadd)
  acf=: the symlink value to use for /etc/pkgadd.conf       (unsupported by pkgadd)
  af  : force install, overwrite conflicting files          (-f, --force)
  ar= : specify alternative installation root               (-r, --root <path>)
  au  : upgrade package with the same name                  (-u, --upgrade)
Unsupported pkgadd options (not permitted in prtpkg):       unsupported:
Note: NEVER allow pkgadd to run outside of prtpkg in a
      prtpkg-maintained system!
        print help and exit                                 (-h, --help)
        print version and exit                              (-v, --version)

PKGRM OVERRIDES (TYPEs B, I, U, R, r) sorted alphabetically within groups: --------------
  rr= : specify alternative installation root               (-r, --root <path>)
Unsupported pkgrm options (not permitted in prtpkg):        unsupported:
Note: NEVER allow pkgrm to run outside of prtpkg in a
      prtpkg-maintained system!
        print help and exit                                 (-h, --help)
        print version and exit                              (-v, --version)
```

8.1.6. prtpkglog [in transition to data reorg]

```
00000000-00:00:00UT -rc  t package----- version--- coll build- batch try-
working_directory----------- command------------------ --------------------------------------
20170130-14:15:34UT   0  U xorg-libxi   1.7.9-1    xorg pkgsb3 00031  06
/var/log/pkgbuild/_/00031/06 prt-cache update -if -kw -fr -ns --install-scripts -f xorg-libxi
20170130-14:22:33UT   0  U mesa3d       12.0.6-1   xorg pkgsb3 00031  06
/var/log/pkgbuild/_/00031/06 prt-cache update -if -kw -fr -ns --install-scripts -f mesa3d
20170130-14:22:51UT   0  U libva        1.7.3-1    opt  pkgsb3 00031  06
/var/log/pkgbuild/_/00031/06 prt-cache update -if -kw -fr -ns --install-scripts -f libva
20170130-14:23:10UT   0  U freeglut     3.0.0-1    opt  pkgsb3 00031  06
/var/log/pkgbuild/_/00031/06 prt-cache update -if -kw -fr -ns --install-scripts -f freeglut
20170130-14:27:20UT   0  U fontforge    20161012-1 opt  pkgsb3 00031  06
/var/log/pkgbuild/_/00031/06 prt-cache update -if -kw -fr -ns --install-scripts -f fontforge
20170130-15:07:58UT   0  U firefox      51.0-1     opt  pkgsb3 00031  06
/var/log/pkgbuild/_/00031/06 prt-cache update -if -kw -fr -ns --install-scripts -f firefox
```

8.1.7. /usr/prtpkg/cells/dcz[ZD]/CRUX-3.2/pkgsb3/log/00031/06.log

```
==== prtpkgbatch is running batch 00031 try 06 in root [ZD] of host dcz at Mon Jan 30 13:32:31 UTC 2017
-rw-r--r-- 1 root root 1.9K Jan 30 13:31 /usr/prtpkg/systems/dcz[ZD]/CRUX-3.2/00031.prtpkg
# sysup 2017-01-29
## do_prtpkg U openssl
## do_prtpkg U libpcre
## do_prtpkg U flex
## do_prtpkg U libspiro
## do_prtpkg U kbd
## do_prtpkg U libyaml
## do_prtpkg U orc
## do_prtpkg U curl
## do_prtpkg U elfutils
## do_prtpkg U speex
## do_prtpkg U sed
## do_prtpkg U openldap
## do_prtpkg U nss
## do_prtpkg U nfs-utils
## do_prtpkg U btrfs-progs
## do_prtpkg U krb5
## do_prtpkg U glib
#
# The first try to U python3-setuptools failed for the situation described
# in the port's new README, so attempt the solution it documents...
#
## do_prtpkg I python3-pip
#
# The second try failed looking for module six -- since package six is
# installed, try installing python3-six (all dependencies already installed)...
#
## do_prtpkg I python3-six
#
# The 3rd try complains about module packaging not found--try installing all
# 6c37 python3-* ports with their uninstalled deps...
#
## do_prtpkg I log4cplus
## do_prtpkg I leptonica
## do_prtpkg I libwebp
#
# Alan's tesseract fails checksum, so select 6c37's by adding it to the 1st
# 6c37 prtdir...  Only multiple ports per prtdir is breaking prtlist somehow.
# Worry about that latter, instead create a _symlinks collection that holds
# symlinks to all the ports previously in "prtdir collection:port[, port]..."
# statements, which made prtlist happy...
#
# Only 6c37's tesseract won't build, either--meanwhile, there is a new
# pre-install script for python3-setuptools that needs to be run for it to
# install properly, so modified prtpkg to default to --install-scripts for
# updates, and get rid of the other unnecessary 6c37 ports deps...
#
do_prtpkg R libwebp
do_prtpkg R leptonica
do_prtpkg R log4cplus
do_prtpkg U python3-setuptools
do_prtpkg U llvm
do_prtpkg U taglib
do_prtpkg U gobject-introspection
do_prtpkg U mako
do_prtpkg U libvirt
do_prtpkg U xorg-libxi
do_prtpkg U mesa3d
do_prtpkg U libva
do_prtpkg U freeglut
do_prtpkg U fontforge
do_prtpkg U firefox
==== prtpkgbatch is now running the above commands:
Mon Jan 30 13:32:31 UTC 2017 -- whatprt libwebp => /usr/ports/contrib 0.5.1-1
Mon Jan 30 13:32:31 UTC 2017 -- whatpkg libwebp => contrib     0.5.1-1        pkgsb3         00031          04
Mon Jan 30 13:32:31 UTC 2017 -- Invoking /com/bin/prtpkg R libwebp _00031 06
Mon Jan 30 13:32:32 UTC 2017 -- Invocation succeeded
==== prtpkgbatch command <do_prtpkg R libwebp> returned 0
Mon Jan 30 13:32:32 UTC 2017 -- whatprt leptonica => /usr/ports/6c37 1.74.1-1
Mon Jan 30 13:32:32 UTC 2017 -- whatpkg leptonica => 6c37      1.74.1-1       pkgsb3         00031          04
Mon Jan 30 13:32:32 UTC 2017 -- Invoking /com/bin/prtpkg R leptonica _00031 06
Mon Jan 30 13:32:33 UTC 2017 -- Invocation succeeded
==== prtpkgbatch command <do_prtpkg R leptonica> returned 0
Mon Jan 30 13:32:33 UTC 2017 -- whatprt log4cplus => /usr/ports/6c37 1.2.0-1
Mon Jan 30 13:32:33 UTC 2017 -- whatpkg log4cplus => 6c37      1.2.0-1        pkgsb3         00031          04
Mon Jan 30 13:32:33 UTC 2017 -- Invoking /com/bin/prtpkg R log4cplus _00031 06
Mon Jan 30 13:32:34 UTC 2017 -- Invocation succeeded
==== prtpkgbatch command <do_prtpkg R log4cplus> returned 0
==== prtpkgbatch archived port _symlinks/python3-setuptools
====            in /usr/prtpkg/systems/dcz[ZD]/CRUX-3.2/log/_/00031/06/python3-setuptools.port.tar.bz2
```

[...]

```
==== prtpkgbatch while read EOF encountered
==== prtpkgbatch finished running the above commands with status 0
==== Running prtpkginfo -sup -dbg -lpf -lcf -bld -bat -try -dat -typ
The pkg/db and/or prtpkg.log files are more recent than any
prtpkg_by_pkg.*.txt file in the working directory.  Thus,
the -upd option is in effect even if it was not specified
Generating /var/lib/pkg/prtpkg_by_pkg.20170130-150758.txt...
Generating /var/lib/pkg/prtpkg_by_col.20170130-150758.txt...
Generating /var/lib/pkg/prtpkg_by_pkg.20170130-150758.log from /var/lib/pkg/prtpkg_by_pkg.20170130-150758.txt...
Generating /var/lib/pkg/prtpkg_by_col.20170130-150758.log from /var/lib/pkg/prtpkg_by_col.20170130-150758.txt...
*** prtpkg_by_pkg.20170130-055017.txt          2017-01-30 05:50:17.471873661 +0000
--- prtpkg_by_pkg.20170130-150758.txt          2017-01-30 15:07:58.752057202 +0000
***************
*** 86 ****
! firefox       50.1.0-1        [same]          opt             pkgsb3          00025           18              20170113-
17:59:55UT      B               /usr/ports/opt/firefox          pkgmk -d -f -u -if -kw -ns
--- 86 ----
! firefox       51.0-1          [same]          opt             pkgsb3          00031           06              20170130-
15:07:58UT      U               /var/log/pkgbuild/_/00031/06    prt-cache update -if -kw -fr -ns --install-scripts -f firefox
***************
*** 92 ****
! fontforge     20150824-1      [same]          opt             pkgsb3          00025           17              20170113-
17:12:51UT      B               /usr/ports/opt/fontforge        pkgmk -d -f -u -if -kw -ns
--- 92 ----
! fontforge     20161012-1      [same]          opt             pkgsb3          00031           06              20170130-
14:27:20UT      U               /var/log/pkgbuild/_/00031/06    prt-cache update -if -kw -fr -ns --install-scripts -f fontforge
***************
*** 94 ****
! freeglut      2.8.1-1         [same]          opt             pkgsb3          00025           13              20170113-
16:36:47UT      B               /usr/ports/opt/freeglut         pkgmk -d -f -u -if -kw -ns
--- 94 ----
! freeglut      3.0.0-1         [same]          opt             pkgsb3          00031           06              20170130-
14:23:10UT      U               /var/log/pkgbuild/_/00031/06    prt-cache update -if -kw -fr -ns --install-scripts -f freeglut
***************
*** 127 ****
! gobject-introspection         1.48.0-1        [same]          opt             pkgsb3          00025           11
                20170113-11:29:45UT             B               /usr/ports/opt/gobject-introspection            pkgmk -d -f -u -if
-kw -ns
--- 127 ----
! gobject-introspection         1.50.0-1        [same]          opt             pkgsb3          00031           06
                20170130-14:12:47UT             U               /var/log/pkgbuild/_/00031/06    prt-cache update -if -kw -fr -ns
--install-scripts -f gobject-introspection
***************
*** 187 ****
! leptonica     1.74.1-1        [same]          6c37            pkgsb3          00031           04              20170130-
05:09:46UT      I               /var/log/pkgbuild/_/00031/04    prt-cache install -if -kw -ns --install-scripts -f leptonica
--- 187 ----
! leptonica     [not installed] 1.74.1-1        6c37            pkgsb3          00031           04              20170130-
05:09:46UT      I               /var/log/pkgbuild/_/00031/04    prt-cache install -if -kw -ns --install-scripts -f leptonica
***************
*** 268 ****
! libva         1.7.2-1         [same]          opt             pkgsb3          00025           12              20170113-
12:21:29UT      B               /usr/ports/opt/libva            pkgmk -d -f -u -if -kw -ns
--- 268 ----
! libva         1.7.3-1         [same]          opt             pkgsb3          00031           06              20170130-
14:22:51UT      U               /var/log/pkgbuild/_/00031/06    prt-cache update -if -kw -fr -ns --install-scripts -f libva
***************
*** 271 ****
! libvirt       2.5.0-2         [same]          nullspoon       pkgsb3          00025           11              20170113-
11:29:10UT      B               /usr/ports/nullspoon/libvirt    pkgmk -d -f -u -if -kw -ns
--- 271 ----
! libvirt       3.0.0-1         [same]          nullspoon       pkgsb3          00031           06              20170130-
14:15:18UT      U               /var/log/pkgbuild/_/00031/06    prt-cache update -if -kw -fr -ns --install-scripts -f libvirt
***************
*** 277 ****
! libwebp       0.5.1-1         [same]          contrib         pkgsb3          00031           04              20170130-
05:10:09UT      I               /var/log/pkgbuild/_/00031/04    prt-cache install -if -kw -ns --install-scripts -f libwebp
--- 277 ----
! libwebp       [not installed] 0.5.1-1         contrib         pkgsb3          00031           04              20170130-
05:10:09UT      I               /var/log/pkgbuild/_/00031/04    prt-cache install -if -kw -ns --install-scripts -f libwebp
***************
*** 291,292 ****
! llvm          3.9.0-2         [same]          opt             pkgsb3          00025           07              20170113-
07:48:21UT      B               /usr/ports/opt/llvm             pkgmk -d -f -u -if -kw -ns
! log4cplus     1.2.0-1         [same]          6c37            pkgsb3          00031           04              20170130-
05:08:48UT      I               /var/log/pkgbuild/_/00031/04    prt-cache install -if -kw -ns --install-scripts -f log4cplus
--- 291,292 ----
! llvm          3.9.1-3         [same]          opt             pkgsb3          00031           06              20170130-
14:11:42UT      U               /var/log/pkgbuild/_/00031/06    prt-cache update -if -kw -fr -ns --install-scripts -f llvm
! log4cplus     [not installed] 1.2.0-1         6c37            pkgsb3          00031           04              20170130-
05:08:48UT      I               /var/log/pkgbuild/_/00031/04    prt-cache install -if -kw -ns --install-scripts -f log4cplus
***************
*** 303 ****
```

```
! mako          1.0.1-1         [same]          opt             pkgsb3          00025           11              20170113-
11:33:45UT      B               /usr/ports/opt/mako             pkgmk -d -f -u -if -kw -ns
--- 303 ----
! mako          1.0.4-1         [same]          opt             pkgsb3          00031           06              20170130-
14:12:53UT      U               /var/log/pkgbuild/_/00031/06   prt-cache update -if -kw -fr -ns --install-scripts -f mako
***************
*** 307 ****
! mesa3d        12.0.5-1        [same]          xorg            pkgsb3          00025           12              20170113-
12:15:30UT      B               /usr/ports/xorg/mesa3d          pkgmk -d -f -u -if -kw -ns
--- 307 ----
! mesa3d        12.0.6-1        [same]          xorg            pkgsb3          00031           06              20170130-
14:22:33UT      U               /var/log/pkgbuild/_/00031/06   prt-cache update -if -kw -fr -ns --install-scripts -f mesa3d
***************
*** 391 ****
! python3-setuptools            32.3.1-1        [same]          6c37            pkgsb3          00025           10
                20170113-11:11:40UT             I               /var/log/pkgbuild/_/00025/10   prt-cache install --margs="-if" -
kw -ns --install-scripts --aargs="-f"                python3-setuptools
--- 391 ----
! python3-setuptools            34.1.0-1        [same]          _symlinks       pkgsb3          00031           06
                20170130-13:32:44UT             U               /var/log/pkgbuild/_/00031/06   prt-cache update -if -kw -fr -ns
--install-scripts -f python3-setuptools
***************
*** 452 ****
! taglib        1.11-1          [same]          opt             pkgsb3          00025           07              20170113-
06:56:31UT      B               /usr/ports/opt/taglib           pkgmk -d -f -u -if -kw -ns
--- 452 ----
! taglib        1.11.1-1        [same]          opt             pkgsb3          00031           06              20170130-
14:12:04UT      U               /var/log/pkgbuild/_/00031/06   prt-cache update -if -kw -fr -ns --install-scripts -f taglib
***************
*** 591 ****
! xorg-libxi    1.7.8-1         [same]          xorg            pkgsb3          00025           12              20170113-
12:00:43UT      B               /usr/ports/xorg/xorg-libxi      pkgmk -d -f -u -if -kw -ns
--- 591 ----
! xorg-libxi    1.7.9-1         [same]          xorg            pkgsb3          00031           06              20170130-
14:15:34UT      U               /var/log/pkgbuild/_/00031/06   prt-cache update -if -kw -fr -ns --install-scripts -f xorg-libxi
*** prtpkg_by_col.20170130-055017.txt          2017-01-30 05:50:17.474873675 +0000
--- prtpkg_by_col.20170130-150758.txt          2017-01-30 15:07:58.755057215 +0000
***************
*** 23,24 ****
! leptonica     1.74.1-1        [same]          6c37            pkgsb3          00031           04              20170130-
05:09:46UT      I               /var/log/pkgbuild/_/00031/04   prt-cache install -if -kw -ns --install-scripts -f leptonica
! log4cplus     1.2.0-1         [same]          6c37            pkgsb3          00031           04              20170130-
05:08:48UT      I               /var/log/pkgbuild/_/00031/04   prt-cache install -if -kw -ns --install-scripts -f log4cplus
--- 23,24 ----
! leptonica     [not installed] 1.74.1-1        6c37            pkgsb3          00031           04              20170130-
05:09:46UT      I               /var/log/pkgbuild/_/00031/04   prt-cache install -if -kw -ns --install-scripts -f leptonica
! log4cplus     [not installed] 1.2.0-1         6c37            pkgsb3          00031           04              20170130-
05:08:48UT      I               /var/log/pkgbuild/_/00031/04   prt-cache install -if -kw -ns --install-scripts -f log4cplus
***************
*** 27 ****
- python3-setuptools            32.3.1-1        [same]          6c37            pkgsb3          00025           10
                20170113-11:11:40UT             I               /var/log/pkgbuild/_/00025/10   prt-cache install --margs="-if" -
kw -ns --install-scripts --aargs="-f"                python3-setuptools
--- 26 ----
***************
*** 29 ****
--- 29 ----
+ python3-setuptools            34.1.0-1        [same]          _symlinks       pkgsb3          00031           06
                20170130-13:32:44UT             U               /var/log/pkgbuild/_/00031/06   prt-cache update -if -kw -fr -ns
--install-scripts -f python3-setuptools
***************
*** 86 ****
! libwebp       0.5.1-1         [same]          contrib         pkgsb3          00031           04              20170130-
05:10:09UT      I               /var/log/pkgbuild/_/00031/04   prt-cache install -if -kw -ns --install-scripts -f libwebp
--- 86 ----
! libwebp       [not installed] 0.5.1-1         contrib         pkgsb3          00031           04              20170130-
05:10:09UT      I               /var/log/pkgbuild/_/00031/04   prt-cache install -if -kw -ns --install-scripts -f libwebp
***************
*** 258 ****
! libvirt       2.5.0-2         [same]          nullspoon       pkgsb3          00025           11              20170113-
11:29:10UT      B               /usr/ports/nullspoon/libvirt    pkgmk -d -f -u -if -kw -ns
--- 258 ----
! libvirt       3.0.0-1         [same]          nullspoon       pkgsb3          00031           06              20170130-
14:15:18UT      U               /var/log/pkgbuild/_/00031/06   prt-cache update -if -kw -fr -ns --install-scripts -f libvirt
***************
*** 299 ****
! firefox       50.1.0-1        [same]          opt             pkgsb3          00025           18              20170113-
17:59:55UT      B               /usr/ports/opt/firefox          pkgmk -d -f -u -if -kw -ns
--- 299 ----
! firefox       51.0-1          [same]          opt             pkgsb3          00031           06              20170130-
15:07:58UT      U               /var/log/pkgbuild/_/00031/06   prt-cache update -if -kw -fr -ns --install-scripts -f firefox
***************
*** 302,303 ****
! fontforge     20150824-1      [same]          opt             pkgsb3          00025           17              20170113-
17:12:51UT      B               /usr/ports/opt/fontforge        pkgmk -d -f -u -if -kw -ns
! freeglut      2.8.1-1         [same]          opt             pkgsb3          00025           13              20170113-
16:36:47UT      B               /usr/ports/opt/freeglut         pkgmk -d -f -u -if -kw -ns
--- 302,303 ----
! fontforge     20161012-1      [same]          opt             pkgsb3          00031           06              20170130-
14:27:20UT      U               /var/log/pkgbuild/_/00031/06   prt-cache update -if -kw -fr -ns --install-scripts -f fontforge
! freeglut      3.0.0-1         [same]          opt             pkgsb3          00031           06              20170130-
14:23:10UT      U               /var/log/pkgbuild/_/00031/06   prt-cache update -if -kw -fr -ns --install-scripts -f freeglut
***************
*** 318 ****
! gobject-introspection         1.48.0-1        [same]          opt             pkgsb3          00025           11
                20170113-11:29:45UT             B               /usr/ports/opt/gobject-introspection            pkgmk -d -f -u -if
-kw -ns
--- 318 ----
```

```
! gobject-introspection         1.50.0-1        [same]          opt             pkgsb3          00031           06
                20170130-14:12:47UT             U               /var/log/pkgbuild/_/00031/06    prt-cache update -if -kw -fr -ns
--install-scripts -f gobject-introspection
***************
*** 379 ****
! libva         1.7.2-1         [same]          opt             pkgsb3          00025           12              20170113-
12:21:29UT      B               /usr/ports/opt/libva            pkgmk -d -f -u -if -kw -ns
--- 379 ----
! libva         1.7.3-1         [same]          opt             pkgsb3          00031           06              20170130-
14:22:51UT      U               /var/log/pkgbuild/_/00031/06    prt-cache update -if -kw -fr -ns --install-scripts -f libva
***************
*** 389 ****
! llvm          3.9.0-2         [same]          opt             pkgsb3          00025           07              20170113-
07:48:21UT      B               /usr/ports/opt/llvm             pkgmk -d -f -u -if -kw -ns
--- 389 ----
! llvm          3.9.1-3         [same]          opt             pkgsb3          00031           06              20170130-
14:11:42UT      U               /var/log/pkgbuild/_/00031/06    prt-cache update -if -kw -fr -ns --install-scripts -f llvm
***************
*** 393 ****
! mako          1.0.1-1         [same]          opt             pkgsb3          00025           11              20170113-
11:33:45UT      B               /usr/ports/opt/mako             pkgmk -d -f -u -if -kw -ns
--- 393 ----
! mako          1.0.4-1         [same]          opt             pkgsb3          00031           06              20170130-
14:12:53UT      U               /var/log/pkgbuild/_/00031/06    prt-cache update -if -kw -fr -ns --install-scripts -f mako
***************
*** 454 ****
! taglib        1.11-1          [same]          opt             pkgsb3          00025           07              20170113-
06:56:31UT      B               /usr/ports/opt/taglib           pkgmk -d -f -u -if -kw -ns
--- 454 ----
! taglib        1.11.1-1        [same]          opt             pkgsb3          00031           06              20170130-
14:12:04UT      U               /var/log/pkgbuild/_/00031/06    prt-cache update -if -kw -fr -ns --install-scripts -f taglib
***************
*** 519 ****
! mesa3d        12.0.5-1        [same]          xorg            pkgsb3          00025           12              20170113-
12:15:30UT      B               /usr/ports/xorg/mesa3d          pkgmk -d -f -u -if -kw -ns
--- 519 ----
! mesa3d        12.0.6-1        [same]          xorg            pkgsb3          00031           06              20170130-
14:22:33UT      U               /var/log/pkgbuild/_/00031/06    prt-cache update -if -kw -fr -ns --install-scripts -f mesa3d
***************
*** 601 ****
! xorg-libxi    1.7.8-1         [same]          xorg            pkgsb3          00025           12              20170113-
12:00:43UT      B               /usr/ports/xorg/xorg-libxi      pkgmk -d -f -u -if -kw -ns
--- 601 ----
! xorg-libxi    1.7.9-1         [same]          xorg            pkgsb3          00031           06              20170130-
14:15:34UT      U               /var/log/pkgbuild/_/00031/06    prt-cache update -if -kw -fr -ns --install-scripts -f xorg-libxi
==== Results of revdep:
libreoffice
opera
syslinux
tesseract
==== End     of revdep results.
==== Do not forget to run rejmerge
```

8.1.8. /usr/prtpkg/CRUX-3.2/prtpkg_by_col.20170130-161126.log

```
package---------------------------- pkg/db_version- port_version collection-- build- batch try date_time---------- type
                 firefox-java-plugin 1.7.0-2         !?           !?           !?     !?    !?  !?                  !?
                                  gc 7.4.2-1         !?           !?           !?     !?    !?  !?                  !?
                           leptonica [not installed] 1.74.1-1     6c37         pkgsb3 00031 04  20170130-05:09:46UT I
                            log4cplus [not installed] 1.2.0-1      6c37         pkgsb3 00031 04  20170130-05:08:48UT I
                          python3-pip 9.0.1-1         [same]       6c37         pkgsb3 00031 01  20170129-18:05:33UT I
                     python3-pyflakes 1.5.0-1         [same]       6c37         pkgsb3 00031 05  20170130-05:50:10UT I
                           rubberband 1.8.1-1         [same]       6c37         pkgsb3 00025 11  20170113-11:27:17UT B
                       vamp-plugin-sdk 2.6-2           [same]       6c37         pkgsb3 00025 07  20170113-06:55:13UT B
                   python3-setuptools 34.1.0-1        [same]       _symlinks    pkgsb3 00031 06  20170130-13:32:44UT U
                            tesseract 3.04.01-1       [same]       _symlinks    pkgsb3 00031 05  20170130-05:49:59UT I
                                atkmm 2.24.2-1        [same]       contrib      pkgsb3 00025 18  20170113-20:52:49UT B
                                boost 1.63.0-1        [same]       contrib      pkgsb3 00025 07  20170113-07:10:43UT B
                         bridge-utils 1.5-1           [same]       contrib      pkgsb3 00025 01  20170109-01:48:19UT B
                              cairomm 1.12.0-1        [same]       contrib      pkgsb3 00025 12  20170113-12:07:19UT B
                             denyhost 2.9-1           [same]       contrib      pkgsb3 00025 07  20170113-07:49:25UT B
                                dev86 0.16.21-1       [same]       contrib      pkgsb3 00025 01  20170109-01:50:43UT B
                            dmidecode 2.12-1          [same]       contrib      pkgsb3 00025 01  20170109-01:51:11UT B
                         docbook-xml [not installed] 4.5-6        contrib      pkgsb3 00012 04  20161225-17:08:13UT I
```

8.1.9. pkg_basenames [in transition to data reorg]

```
BBS                         xfwm4-themes(4.10.0-1)   D '/usr/share/themes/BBS'
BC.3x.gz                    ncurses(6.0-3)           F '/usr/share/man/man3/BC.3x.gz'
'                                                    -> 'curs_termcap.3x.gz'
'                                                    => '/usr/share/man/man3/curs_termcap.3x.gz'
BCC.pm                      perl(5.22.3-1)           F '/usr/lib/perl5/5.22/ExtUtils/CBuilder/Platform/Windows/BCC.pm'
BDCE.h                      llvm(3.9.1-3)            F '/usr/include/llvm/Transforms/Scalar/BDCE.h'
BER.h                       gcj-jdk(5.4.0-1)         F '/usr/include/c++/5.4.0/gnu/java/security/ber/BER.h'
BEREncodingException.h      gcj-jdk(5.4.0-1)         F '/usr/include/c++/5.4.0/gnu/java/security/ber/BEREncodingException.h'
BERReader.h                 gcj-jdk(5.4.0-1)         F '/usr/include/c++/5.4.0/gnu/java/security/ber/BERReader.h'
BERValue.h                  gcj-jdk(5.4.0-1)         F '/usr/include/c++/5.4.0/gnu/java/security/ber/BERValue.h'
BF_cbc_encrypt.3ssl.gz      openssl(1.0.2k-1)        F '/usr/share/man/man3/BF_cbc_encrypt.3ssl.gz'
'                                                    -> 'blowfish.3ssl.gz'
'                                                    => '/usr/share/man/man3/blowfish.3ssl.gz'
BF_cfb64_encrypt.3ssl.gz    openssl(1.0.2k-1)        F '/usr/share/man/man3/BF_cfb64_encrypt.3ssl.gz'
'                                                    -> 'blowfish.3ssl.gz'
'                                                              => '/usr/share/man/man3/blowfish.3ssl.gz'
```

8.1.10. misspkglog [in transition to data reorg]

```
6c37              rubberband                       1.8.1-1           vamp-plugins-sdk
6c37-git          connman                          git-1             gnutils
6c37-git          nctelegram                       git-1             pytg3
6c37-git          od6config                        git-1             fglrx
6c37-git          urwid3                           1.3.1-1           setuptools3
alan              anki                             2.0.39-1          distribute
alan              docbook2x                        0.8.8-1           docbook
alan              dovecot                          2.2.25-6          tcp_wrappers
alan              freevo                           1.9.0-1           pygame
alan              gocr                             0.50-1            netpbm
alan              hplip                            3.11.10-1         foomatic-filters
alan              k2pdfopt                         2.32-1            netpbm
alan              kodi                             16.1-Jarvis-6     freetype2
alan              lirc-xmms-plugin                 1.4-1             xmms
alan              nfs-utils                        1.3.3-4           udev
alan              p5-calendar-japanese-holiday     0.03-1            p5-calendar
alan              p5-cam-pdf                       1.60-1            p5-crypt-rc4
alan              p5-dbd-sqlite                    1.46-1            sqlite
alan              p5-module-signature              0.70-1            p5-digest-sha
alan              p5-net-dbus                      1.0.0-1           p5-xml-twig
alan              p5-text-pdf                      0.29a-1           p5-crypt-rc4
alan              p5-xml-xql                       0.68-1            p5-date-manip
alan              p5-xml-xql                       0.68-1            p5-xml
alan              p5-yaml                          1.15-2            p5-spiffy
alan              p5-yaml                          1.15-2            p5-test-base
alan              py-send2trash                    1.3.0-1           distribute
alan              pyqt                             4.9.1-1           qt
alan              pysqlite                         2.4.1-1           sqlite
alan              spamassassin                     3.4.1-2           p5-lwp
alan              timidity-sgm                     2.01-1            timidity
```

8.1.11. prtlist [in transition to data reorg]

```
babl                                         /usr/ports/opt              0.1.16-1
babl"                                    /usr/ports/teatime              0.1.18-1     [no Depends On]
babl"                                /usr/ports/deepthought              0.1.18-1
backlight                               /usr/ports/6c37-git                 git-1
  git
baloo                                        /usr/ports/kde4             4.14.3-1
  kdepimlibs
  xapian
  kfilemetadata
baloo"                                        /usr/ports/kf5             5.30.0-1
  gtk
  kfilemetadata
  kidletime
  kio
  lmdb
  gtk
baloo-widgets                                 /usr/ports/kf5            16.12.1-1
  kdelibs4support
  baloo
bash                                         /usr/ports/core             4.3.48-1
  ncurses
  readline
bash-completion                               /usr/ports/opt                2.1-2
  bash
bash-completion-extras               /usr/ports/deepthought                 0.0-1
  bash-completion
bash-git-prompt                      /usr/ports/deepthought                 1.2-1     [no Depends On]
bashish                                  /usr/ports/romster               2.2.4-1
bashish"                                  /usr/ports/6c37
2.2.4-1
```

8.1.12. prtpkginfo -h [in transition to data reorg]

```
prtpkginfo: Create prtpkg_by_{pkg,col}.$tstmp.{txt,log} files
            from /usr/prtpkg/systems/$system/var.lib.pkg/db and
            /usr/prtpkg/system/$system/prtpkg.txt files.  The
            txt files are created if the input files have been
            modified since the most recent prtpkg_by_pkg.*.txt
            file in the working directory was created, if any (or
            this can be explicitly requested via the -upd flag).
            The log files are created from the txt files if requested.
            If there are current prtpkg_by_{pkg,col}.txt files
            in the working directory, diffs files against any new
            .txt files are created and the older .txt files are
            compressed and moved into the working directory's
            prtpkg_past subdirectory that will be created if
            necessary.
Note: Unknown flags are silently ignored.
Processing option flags:
  -sup: change the working directory to /usr/prtpkg/$PRTPKG_PORTS
        (requires superuser authority).
  -upd: requests update of the prtpkg_*.txt files in the working
        directory even if that does not seem to be necessary.
  -lpf: requests output of the by_pkg log as the file
        prtpkg_by_pkg.log file in the working directory.
  -lcf: requests output of the by_col log as the file
        prtpkg_by_col.log file in the working directory.
  -lps: requests output of the by_pkg log to stdout.
  -lcs: requests output of the by_col log to stdout.
  -dbg: log debug messages to /tmp/prtpkg.$USER.debug
  Note: -lpf and -lps are two sides of a coin--the file
        option prevails if both are specified; if neither
        is specified, the file/stream is not output.
        The same is true of the -lcf and -lcs options.
Optional log file fields selected by flags:
  -bld: include build ID in log files|streams
  -bat: include batch ID in log files|streams
  -try: include try ID in log files|streams
  -dat: include date in log files|streams
  -typ: include prtpkg record type in log files|streams
  -pwd: include cmd's working directory in log files|streams
  -cmd: include low-level command in log files|streams
  -all: include all of the preceding fields in log files|streams
```

8.2. Prologs of Scripts

8.2.1. prtpkgbatch [contains TODO items]

```
Process a prtpkg batch as specified by:
Required Arguments:
  $1: the batch_ID of the prtpkg file to process
  $2: the build_ID of the build configuration to run in [TODO]
Optional arguments that can be specified to be passed
as overrides into all prtpkg commands invoked by the
batch run.
  'redo'                                            [TODO]
  'test'
Environment Variables:
  PRTPKG_SYS    the prtpkg platform this command is     [TODO]
                running on                              [TODO]
  PRTPKG_PORTS  the ports_ID of the release of port     [TODO]
                collections to be used for building     [TODO]
```

8.2.2. prtpkglog [in transition to data reorg]

```
prtpkglog -- output /usr/prtpkg/$PRTPKG_PORTSU/prtpkg.txt in log format (no tabs,
fields columnized using blanks) piped into the command defined in the arguments
or by default into "less -S" (truncate each line at the screen's right boundary).
```

8.2.3. localize_ports [in transition to data reorg]

```
If no args, configures all ports after the initial portsu script for maintenance on this
system using the enhanced framework; otherwise, configures a new port for the enhanced
framework ($1=collection $2=package) and is meant to be called from the prtpkg script.
Prolog: missing_packages [in transition to data reorg]
Show packages depended on that do not exist, sorted by:
  name    of collection referencing missing package
  name    of package    referencing missing package
  version of package    referencing missing package
  name    of missing depended on package
The only argument is the target-ID of the running system to
confirm this invocation is not merely a help request.
```

8.2.4. missing_packages_doit (gawk) [in transition to data reorg]

```
from /usr/CRUX/prtlist.*.txt, write missing_ports records to
stdout (expected to be redirected to /tmp/missing_ports.pkgs.
The /tmp/missing_packages.pkgs file created by the
missing_packages script is read into an array to provide the
definitive list of all known packages.
```

8.2.5. pkgaddconf [targets contain TODO items]

```
            ensure the /etc/pkgadd.conf symlink points to the version defined by the
            specified target argument; e.g., for argument xyzzy[plugh], the symlink
            should be or become
               /etc/pkgadd.conf -> /usr/CRUX/xyzzy[plugh]/pkgadd.conf
            If successful, the script writes the target component of the previous
            symlink (may be the same) to stdout.
       -or- omit an argument to output the target component of the current symlink
            to stdout (this does not require the caller to be the superuser).
    return: Status code 0 is returned to indicate the data written to stdout is the
            expected value; otherwise, an error message is written to stdout and a
            non-zero status code is returned.

targets:    define a CRUX port/package building structure to support a particular
            set of port/package maintenance activities.  Targets come in three types
            having one of the following formats as appropriate:
              release: CRUX-$V.$R; e.g., CRUX-3.2
              system:  $HOST[$ROOT]; e.g., mydevbox[SSD3], where:
                 $HOST is the output of the hostname command for the targeted
                       system
                 $ROOT is the enterprise's identifier for the targeted root
                       filesystem of that system, ideally a volume, partition,
                       or filesystem label
              shared:  an enterprise-meaningful string without any brackets; e.g.,
                       shared, common, production, testing, marketing

            A release target is used for ports -u processing by one CRUX software
            maintenance system designated to do so on behalf of all other CRUX
            systems in the enterprise running the same version of CRUX.  It defines
            the standard /usr/ports tree of prtdirs on that system.

            A system target is used for building and installing/removing packages
            for the targeted system (which is not necessarily the system performing
            the port/package maintenance).  A system target's prt-get.config file
            defines the /usr/prtpkg/$system/ports tree of prtdirs that deploy symlinks
            to redirect to the release ports directories that the system uses (see
            the validate_system command for more information).  For example,
              /usr/prtpkg/systems/xyzzy[plugh]/ports/core/gcc -> /usr/ports.core/gcc

            A shared target is exactly like a system target except it identifies a
            grouping of systems sharing the same building configuration.  One system
            should be designated to perform the actual port/package maintenance on
            the behalf of the group--it need not even be in that group.
```

8.2.6. prtlist [in transition to data reorg]

```
For all packages defined within /etc/prt-get.conf (via prtdir statements),
produce a sorted list of the packages, with different ports of the same name
sorted in prtdir access order (the first is the port that prt-get will act
upon, and the others will have a single double quotation mark appended; i.e.,
a "ditto" indicator).  Each port line includes the prtdir path, the
version-release tuple, and an optional indicator that the package does not
contains a "Depends on:" record.  Each port line is followed by lines
containing the Depends on: values, indented by two spaces, and listed in
the order they are specified in the "Depends on:" record.

This script takes no parameters.  It should be run after a "ports -u" command
(which is one reason you should instead use the portsu script on this system)
or after the order and/or content of the prtdir statements in /etc/prt-get.conf
is modified.

The output is put into a file named as /usr/ports/prtlist.YYYYMMDD-HHMMSS.txt
and if an earlier file exists, the two are compared and the result is placed
into /usr/ports/prtlist.diffs, then the older file is compressed using bzip2
and moved into the /usr/ports/prtlist_past directory.

A /tmp/prtlist.$$.debug file contains details of processing helpful in diagnosing
any misbehavior.
```

8.2.7. prtlist_packages (gawk) [in transition to data reorg]

```
from /usr/prtpkg/$PRTPKG_PORTS/prtlist.*.txt, extract list of available packages
```

8.2.8. prtpkg_symlink (gawk)

```
from piped input for an "/bin/ls -lh" command for a single
symlink, extract the redirection string and write it to stdout
```

8.2.9. prtpkginfo [in transition to data reorg]

```
prtpkginfo -- see catted help below for documentation

input prtpkg.txt package maintenance record format (tab field separators):
$1 [YYYYMMDDhhmmss] $2 $3   $4  $5      $6         $7    $8    $9  $10 $11
00000000-00:00:00UT rc type pkg version collection build batch try pwd cmd

output txt record format (tab field separators) and associated log columns:
$1       $2      $3      $4      $5      $6      $7      $8      $9      $10     $11
pkg_name pkg_ver out_ver prt_col prt_bld prt_bat prt_try prt_dat prt_typ prt_pwd prt_cmd
always   always  always  always  -bld    -bat    -try    -dat    -typ    -pwd    -cmd
```

8.2.10. validate_builds [very early new program]

```
Ensure the target system's tree of ports used in the builds
version of /etc/prt-get.conf (referred to as the builds ports
tree) matches the packages currently installed (or uninstalled
but built) as recorded in /usr/prtpkg/$PRTPKG_PORTS/prtpkg.txt.
If multiple ports for an uninstalled package are available,
the one chosen is the first in the prtdir order according to
the portsu version of prt-get.conf, which uses the _symlinks
pseudo-collection to contain symlinks to any cherry-picked
ports; e.g.,
  /usr/ports/_symlinks/xyzzy -> /usr/ports/plugh/xyzzy
Inodes in builds ports collection subdirectories are symlinks
to the portsu ports tree; e.g.,
  /usr/CRUX/$hrid/ports/plugh/xyzzy -> /usr/ports/plugh/xyzzy
Installed ports that have no port history will be presumed
to be associated with the portsu config's normal selection
(the highest prtdir having a port for that package).
This script takes only optional '-h' and '--help' arguments.
```

8.2.11. validate_symports [in transition from varports]

```
Ensure the symports tree used in the builds version of
/etc/prt-get.conf matches the packages either currently
installed or uninstalled but built as recorded in
/usr/prtpkg/$PRTPKG_PORTS/prtpkg.txt.  If multiple ports
for an uninstalled package are available, the one chosen
is the first in the prtdir order according to the portsu
version of prt-get.conf, which uses the _symlinks pseudo-
collection to contain symlinks to any cherry-picked
ports; e.g.,
  /usr/ports/_symlinks/xyzzy -> /usr/ports/plugh/xyzzy
Inodes in symports collection subdirectories are symlinks
to the portsu ports; e.g.,
  /usr/prtpkg/systems/boxA[rootfsB]/ports/plugh/xyzzy →
  /usr/prtpkg/CRUX-3.2/ports/plugh/xyzzy
Installed ports that have no port history will be presumed
to be associated with the portsu config's normal selection
(the highest prtdir having a port for that package).
Note the /var/ports tree is implicitly connected to the
root filesystem in which it resides (and the host thereof),
identified elsewhere in the prtpkg tools as `hostname`_$ROOT
where ROOT is set to the enterprise's label for the root
filesystem during boot.
This script takes only optional '-h' and '--help' arguments.
```

8.2.12. validate_symports_links (gawk) [in transition from varports]

```
from /tmp/validate_symports.links, identify all
packages that reside in multiple collections
(note pseudo-collection _symlinks ports should
never be seen by this script but the logic for
dealing with them remains in place).  Any multiples
found are recorded in /tmp/validate_varports.dups
for subsequent remediation by continuing
processing of the validate_symports script.
Messages of significance are written to stdout,
and the status code returned is the number of
packages needing remediation
```

8.2.13. whatpkg [in transition to WHATPKG_ variables]

```
Lookup the port collection and version for installed package $1
as recorded in the designated prtpkg.txt file.
```

8.2.14. whatpkg_2ndline (gawk) [in transition to WHATPKG_ variables]

```
from /dev/null, based upon the values of the WHATPKG_PKGINFO and WHATPKG_PRTPKGTXT
environment variable values, write as appropriate to stdout:
       "pkginfo reports uninstalled"
--or-- "pkginfo version mismatch: " followed by the value of WHATPKG_PKGINFO
--or-- nothing
Regardless, return status code zero is returned.
```

8.2.15. whatpkg_pkginfo (gawk) [in transition to WHATPKG_ variables]

```
find WHATPKG_PKG environment variable value in input stream from pkginfo -i
If found, write the installed version to stdout and return status code zero.
If not found, write "[uninstalled]" to stdout and return status code one.
Otherwise, write an error message to stdout and return greater than one status.
```

8.2.16. whatpkg_prtpkgtxt (gawk) [in transition to WHATPKG_ variables]

```
in input /var/log/prtpkg.txt, find the most recent successful
type B, I, or U record for the package defined in the WHATPKG_PKG
environment variable, if any.  If a suitable record is found,
write its following fields separated by tabs:
  collection  port-version buildID batchID tryID
and return a zero status code; if one is not found, write an
informational message to stdout and return status code one;
otherwise, write an error message to stdout and return a status
code greater than one.
```

8.2.17. whatprt [in transition to WHATPRT_ variables]

```
Lookup the port collection for package $1 that the current
/etc/prt-get.conf will cause to be selected.  Write that
as well as the version of the package to stdout and return
status code zero, write an informational message to stdout
and return status code two if the port is not found, or
write an error message to stdout and return a status code
greater than two.  Status code one is returned when help
information is written to stdout.
```

8.2.18. whatprt_doit (gawk) [in transition to WHATPRT_ variables]

```
from the current /etc/prt-get.conf file's prtdir statements,
determine what port will be selected for processing by prt-get
and write the collection's path and the Pkgfile's version-release
information to stdout.  The package to lookup is predefined in
the WHATPRT_PKG environment variable and WHATPRT_DBG is defined
as null (no debugging output) or the string to use in the
name of the debugging output file; i.e.,
/tmp/whatprt.$WHATPRT_DBG.debug.
If the search is successful, status code zero is returned.
```

9. Appendices

9.1. Typographic Character Markup Legend

9.1.1. Monospace Font

- Style `builddef` marks the name of a builddef component
- Style `celltype` marks the name of a cell type or attribute
- Style ***`cmd`*** marks a command name and/or its parameters

- Style `collection` marks the name of a collection
- Style `computer` marks the name of a miscellaneous IT term
- Style CRUX marks the distribution (except in a pathname)
- Style **distro** marks a non-CRUX operating system distribution
- Style `idname` marks the name of an instance of user or other keyname
- Style `keyname` marks the name of a parameter, keyword, etc.
- Style `lock` marks the name of a lock type or class
- Style `package` marks the name of a package
- Style `pathname` marks path name components not otherwise marked up
- Style `prtpkg` marks the enhancement name when not otherwise marked up
- Style `SenSep` marks a space separating intra-paragraph sentences
- Style `service` marks the name of a service
- Style `signal` marks the name of a signal
- Style *`variable`* marks the name of a variable to be substitued with a value

9.1.2. Proportional Font

- Style Default Style marks all unremarkable text
- Style *Emphasis* marks italic characters
- Style **Index Link** marks a clickable cross-reference (note the shadows)
- Style Internet Link marks a clickable URL (note the underline)
- Style **Strong Emphasis** marks bold characters
- Style *Title* marks a title

9.2. Provisioning a PP-stick

[Describe the `PP-stick` download and customization process.]

9.3. Provisioning an EFI System Partition

[Describe the `PP-stick`'s ESP construction process.]

1 Within this document, the Commonwealth Extention (`CEX` for super-short) is often simply called the Extension. The Extension is also referred to as the `prtpkg` software or architecture when focusing on the Extension's software maintenance capabilities, while emphasizing *commonwealth* tends to occur when the focus is more on the Extension's inter-platform design and facilities.

2 Usage of the *noun* **system** in this document strives to hold to a precise definition that differentiates it from the *noun* **cell** (that is also precisely defined in Endnote **4**). A *system* is a really or virtually bootable entity that provides more services to the enterprise than merely running `prtpkg` processes (and perhaps runs no such processes and perhaps not even CRUX) and thus has reliability, availability, security, user data handling/storage, and system administration support requirements beyond those of a mere `prtpkg`-only bootable entity. In short, `prtpkg` and its documentation tries to ignore computing outside its mission and focuses on its cells. There is possible overlap when a system runs software maintained by `prtpkg`, of course, because then it is most likely simultaneously a system and a cell. Also, the *noun* **prtpkg platform** should be understood to be a particular `boot` cell plus all `chroot` cells in its root tree that are not mounted in a networked filesystem, and is explicitly meant to *not* include any virtual machine cells hosted via the `boot` cell.

3 This endnote explains in one paragraph all the botanical and maritime analogies used to reference filesystem namespace components within this entire document. The *noun* **tree** in this document is an inode in a filesystem that has or can have trees of its own and includes all the referenced tree's (*noun*) **branches** (just *subtrees*) and (*noun*) **leaves** (the ends of the lines, never directories or anchors, but possibly reflinks). The *noun* **trunk** connotes the foundational inode of a tree, which must be a directory inode type, and resides within the parent directory of the tree. The trunk is not the directory of the tree. Rather, the *noun* **trunk** connects to the (*adjective*) **trunk** (*noun*) **directory**, possibly empty but having the potential to sprout branches. A symlink to a trunk directory is not the trunk itself and is referenced using the *noun* **anchor** when referring to the trunk, otherwise it is referred to as a tree. However, the *transitive verb* **anchor** implies a direct object that may be a trunk or an anchor. Note that a tree may not necessarily refer to a trunk or an anchor—it may be a file inode in the logical filesystem but its name and any content can identify a tree the logical filesystem should contain somewhere. This works somewhat like a `C++` reference so we'll call this a (*noun*) **reflink** when needed. One last definition: the *phrase* **logical filesystem** refers to the effective root filesystem including all other filesystems mounted within the root filesystem's namespace at the time of reference; i.e., all the inodes that are currently accessible via an absolute path.

4 The *noun* **cell** refers to a `prtpkg` software building zone. These can be bootable or chrootable; either way, they have unique instances of `/etc/ports`, `/usr/ports`, `/var`, etc.. A system is a `prtpkg` cell iff it is defined as such in a `prtpkg` commonwealth. The possibility that the cell is also a system is rarely relevant to the discussion.

5 The *noun* **porter** refers to a cell that is authorized to perform ***`portdb`*** and ***`portsu`*** processing within a `prtpkg` commonwealth (even if the commonwealth is comprised of a single `prtpkg` platform).

6 The *noun* **builder** refers to a cell that is authorized to perform ***`pkgmk`*** processing within a `prtpkg` commonwealth (even if the commonwealth is comprised of a single `prtpkg` platform). Builders may be restricted to specific distribution releases and build definitions.

7 The *noun* **deployer** refers to a cell that is authorized to perform ***`pkgadd`***, ***`pkgrm`***, and ***`rejmerge`*** processing on a particular cell (even if the commonwealth is comprised of a single `prtpkg` platform). Normally deployers are only authorized to add and remove packages on their own `BOOTOS ROOTFS` tree and may be restricted to packages created for particular distribution releases and build definitions.