

Customized Install Media



Introduction

There were presentations on Ignite-UX highlighting its capabilities and features at the April 1997 Interworks conference. This paper will concentrate on one specific area of Ignite-UX that has been added since that time. That is how to build your own customized install media. This paper will assume that people reading this have some basic knowledge of Ignite-UX, and have had some exposure to the product. There was a paper presented at the April 1997 Interworks conference ("Ignite-UX Cold Installations") that was an introduction to Ignite-UX that is an excellent reference. There is also a class on Ignite-UX given by HP Customer Education Services. The class number is H19785.

Customized install media has been a longstanding request of HP customers. With Ignite-UX, it is possible to create such media. At present the media can be either DDS tape or CD-ROM. If and when other formats like DVD and DLT are supported for booting HP-UX, support will appear in Ignite-UX shortly thereafter.

All Ignite-UX commands have the path `/opt/ignite/bin` unless otherwise noted. When Ignite-UX is loaded, it modifies `/etc/PATH` to include this path.

Rationale for Customized Install Media

There are some compelling reasons for building customized install media:

- You have a large number of systems that are basically identical and:
 - ◆ the systems do not have network boot capability; or
 - ◆ the networking will not allow easy or fast access to an Ignite-UX server; or
 - ◆ the systems are geographically widespread.
- You have Series 800 systems that lack network boot, and so a boot medium is required to contact the Ignite-UX server.

- You want to be able to hand media to a technician or operator, and have the entire install process automated without human intervention or interaction.
- You want a single media that contains all the desired parts of the operating system (HP-UX, applications, patches, diagnostics and local customizations).

Using customized install media also provides both system standardization and customization simultaneously.

The standardization comes from using “golden system images” which contain a base operating system, applications, patches, third-party software and local customizations, already packaged up in an archive. The entire system has been tested, verified and tuned before creating the image. This image can be the starting point for all installs, so they are guaranteed this much standardization.

The customization comes from using config files to load additional software, change kernel parameters, and run scripts. Software bundles can be:

- interactively chosen
- pre-selected
 - unconditionally
 - conditionally
- invisible

There are also parameters that control the environment that Ignite-UX operates in. The two most important of these are:

- `run_ui`
- `control_from_server`

If `run_ui` is false, then no interaction will occur and the load will proceed according to all the configuration information provided to Ignite-UX. If `control_from_server` is true, then an attempt will be made to contact the Ignite-UX server as defined in the configuration information.

The power is in your hands! You can chose how things work, what you will leave up to the end user, what will happen automatically, and so forth.

Customer Scenario

For the sake of simplicity, this presentation will deal only with HP-UX 10.20.

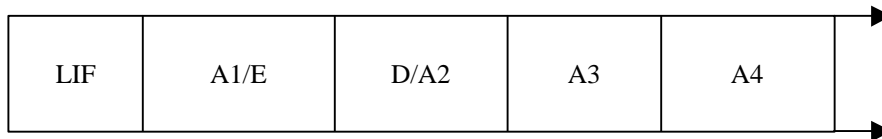
Assume that the intention is to create a customized install media that will be shipped worldwide so that systems can be installed with a “golden image”. There is both a Series 700 and a Series 800 “golden image” built with **make_sys_image**. In addition assume there is a set of applications that are to be chosen interactively by the end user. All software will come from the media, and there will be no contact with nor need for an Ignite-UX server during installation. However you do need an Ignite-UX server to create the media and execute Ignite-UX commands.

The Golden Image

The “golden system images” were created on Series 700 and Series 800 systems running 10.20 using `/opt/ignite/data/scripts/make_sys_image`. The archives are in tar format and are gzipped. The files are called `archive_700.gz` and `archive_800.gz` respectively and are located in `/var/tmp`. The **archive_impact** command (`/opt/ignite/sbin/archive_impact`) has been run on these archives to get disk space usage information (impacts) so that configuration information can be supplied.

DDS tape Layout

A DDS install tape is constructed logically like this:



At the beginning of a bootable DDS tape is a LIF (Logical Interchange Format) volume (LIF). This LIF volume contains all the components required to boot off the tape. In addition, it includes the Ignite-UX toolset and configuration information that controls how Ignite-UX will operate. It includes config file information about the depot on the tape (should there be one) and all archives on the media.

The next portion (A1/E) is either the first OS Archive (A1) or is Empty (E) if the installation is solely from the software depot.

The next portion (D/A2) is either a serial depot (D) or another archive (A2). There can only be one depot on a DDS tape, and it must be the third file on the tape due to a SD restriction.

From there on, there may be other archives (A3, A4,...) limited only by the capacity of the tape. If more archives are needed, they can be put on a second medium.

LIF Volume Content

The **make_medialif** command is used to create the LIF volume. The **lifs -l** output looks like this:

```
volume ISL10 data size 175771 directory size 2 98/02/10 14:06:38
filename  type  start  size  implement  created
-----
ISL      -12800 16     240   0          98/02/10 14:06:38
AUTO     -12289 256    1     0          98/02/10 14:06:38
INDEX    BIN    264    1     0          98/02/10 14:06:38
CONFIG   BIN    272    58    0          98/02/10 14:06:38
HPUX     -12928 336    800   0          98/02/10 14:06:38
INSTALL  -12290 1136   57503 0          98/02/10 14:06:45
INSTALLFS -12290 58640 31774 0          98/02/10 14:06:48
INSTCMDS BIN    90384 9873  0          98/02/10 14:06:51
SYSCMDS  BIN    100264 45901 0          98/02/10 14:07:02
SCRIPTS  BIN    146168 30    0          98/02/10 14:07:02
```

ISL is the Initial System Loader. If it is run interactively, it issues a prompt and waits for user interaction. Otherwise it looks for the AUTO file. It is extracted by **make_medialif** from the default boot file `/opt/ignite/boot/boot_lif`.

AUTO is the autoexecute file. It defines the default (possibly automatic) boot behavior.

INDEX is the default INDEX file (it has the same function as `/var/opt/ignite/INDEX` does on an Ignite-UX server). The file CONFIG is referenced in this file.

CONFIG is the file that contains all the software configuration information. It is recommended that it should begin with the default config file for that release (which can be found in `/opt/ignite/data/Rel_<release>/config`). Additional config files can be added via the `-f` option of `make_medialif`. Information in this file will allow complete access to all the archives and depots on the media.

HPUX is the HP-UX bootstrap utility. It is also extracted from the default boot file.

INSTALL is the 32-bit kernel booted by 32-bit install clients. There is a `-o 64` option to **make_medialif** to create a LIF volume with a 64-bit kernel, but that is needed only for V-class systems. In that case, the file is called VINSTALL.

INSTALLFS is the RAM file system used by install clients. There is configuration information stored in the first 8Kb of this file that is accessible using the **instl_adm** command. If this is a 64-bit LIF volume, the file is called VINSTALLFS.

INSTCMD5 is the gzip archive of the commands needed for disk layout. These commands run on the install kernel and inside the INSTALLFS.

SYSCMDS is the gzip archive of commands used to load the software onto the system. There are different archives for each release.

SCRIPTS is the gzip archive of all post_load and post_config scripts that are required. By default when loading a core archive (load_order is zero, which means it gets loaded first), the two scripts in `/opt/ignite/data/scripts` called `os_arch_post_l` and `os_arch_post_c` are executed. Scripts like these are discussed in depth in *instl_adm(4)*.

For more information on what happens during system boot and what files do what, see Appendix A.

Config Files

It is presumed that readers have some knowledge of config files. For this paper, there are two important concepts to consider. They apply equally to both archives and bundles that are to be loaded onto a target system.

The first concept is a `sw_source`. It specifies the access method to either an archive or a depot. The second concept is a `sw_sel`. It specifies the path of an archive or a bundle in a depot. For details on these objects, see *instl_adm(4)*.

It is always advisable to pass all user-generated config files through **instl_adm -T -f <cfg_file>** to check for syntax errors.

Accessing an Archive on a DDS Tape

The best place to start when creating a config file for an archive is the template file supplied by Ignite-UX in `/opt/ignite/data/examples/core.cfg`. A current copy of this file is included as Appendix B. This file can be copied elsewhere (say `/var/tmp/archive.cfg`) and then edited to suit your situation.

Assume that our DDS tape will be used more for Series 700 systems than Series 800 systems. Hence we will put the Series 700 archive on the tape first. In the diagram above, it will be A1. Since there will be a serial depot on the tape, the Series 800 archive will be located at A3.

To modify the config file to access the Series 700 archive, the following attributes need to be changed in `/var/tmp/archive.cfg` in the `sw_source` "core archive" stanza:

Attribute	Old Value	New Value
<code>source_type</code>	<code>"NET"</code>	<code>"MT"</code>
<code>change_media</code>	<code># change_media=FALSE</code>	<code>change_media=FALSE</code>
<code>nfs_source</code>	<code>nfs_source = "<IP>:<depot>"</code>	<code># nfs_source = "<IP>:<depot>"</code>

These changes will modify the source type from network ("NET") access (which is either NFS, ftp or remsh) to magnetic tape ("MT"). Since the archive is going to reside on the same media, `change_media` is set to false by uncommenting that attribute. To avoid trying to NFS mount that directory, the `nfs_source` is commented out.

Since the template file already has conditional logic that provides for different Series 700 and Series 800 archives, we will use that to our advantage.

Inside the stanza enabled by `HARDWARE_MODEL ~ "9000/7.*"` the following fields must be changed:

Attribute	Old Value	New Value
<code>archive_path</code>	<code>"B.10.20_700_CDE.gz"</code>	<code>"1"</code>
<code>impacts</code>	<code>"/" 27Kb</code>	<code><as reported by archive_impact></code>

The change in `archive_path` indicates that there is one EOF mark to skip on the tape and the archive will begin right after that mark. The archive will be the second file on the tape after the LIF volume. The `impacts` lines must be replaced with whatever was reported by **archive_impact** for the Series 700 archive.

Inside the same stanza the `sw_sel` and `description` strings can be changed to something more descriptive and applicable to your situation. The text inside the double quotes can be changed to whatever you like. They will be visible on the Ignite-UX UI. Note that `archive_type` must match what was done by **make_sys_image**. See `instl_adm(4)` for more on `archive_type`.

Since we have only one Series 700 archive the entire stanza called “golden image2” can be deleted. It was included in case you had two different types of archives, for example one for VUE and one for CDE. If more than one archive per architecture is on the media, it is advisable to use an `exrequisite` attribute between them so only one archive can be selected at one time.

In the stanza for the Series 800 (it is an else clause later in the file) the same sort of changes must be made. However remember that the Series 800 archive is in a different location on the tape. So these attributes need to change:

Attribute	Old Value	New Value
<code>archive_path</code>	<code>"B.10.20_700_CDE.gz"</code>	<code>"3"</code>
<code>impacts</code>	<code>"/" 27Kb</code>	<code><as reported by archive_impact></code>

The change in `archive_path` indicates that there are three EOF marks to skip on the tape (LIF volume, Series 700 archive, and the serial depot). The Series 800 archive is the fourth file on the tape. The `impacts` lines must be replaced with whatever was reported by **archive_impact** for the Series 800 archive.

It is important not to change anything else in the file, unless you are very sure of what you are doing. In particular, it is potentially dangerous to change the `sw_category` and other `sw_source` and `sw_sel` attributes not mentioned above.

Accessing the Serial Depot on a DDS Tape

Assume there is a depot (`/var/tmp/depot`) that contains all the applications you wish to install on top of the archive. It can be a mixture of Series 700-only applications, Series 800-only applications, and applications that can be loaded on both architectures. The `make_config` command is used to create config file information for this depot, and the config file is modified to reflect the ultimate destination of the depot. To create config files, execute the following commands:

```
make_config -s /var/tmp/depot -a 700 -c /var/tmp/depot_700_cfg
make_config -s /var/tmp/depot -a 800 -c /var/tmp/depot_800_cfg
```

On a DDS tape, the depot must be the third file, so there is no need to specify a path to the depot. Change both config files by removing these lines:

Attribute
<code>sd_server = "<IP>"</code>
<code>sd_depot_dir = /var/tmp/depot</code>

Change the following attribute in both files:

Attribute	Old Value	New Value
source_type	"NET"	"MT"

The deleted information is not needed when accessing a serial depot on a tape. The change to `source_type` indicates that the depot is located on a tape instead of over the network.

Creating the Serial Depot

The depot put on a DDS tape is known as a serial depot. It can exist as a regular file. However it cannot be registered which means that it cannot be accessed remotely.

To create a serial depot from `/var/tmp/depot` and store it in `/var/tmp/serialdepot`:

```
swpackage -s /var/tmp/depot -x target_type=tape @ /var/tmp/serialdepot
```

Assembling the DDS Tape

Now that all the components of the tape are done, we can create the LIF volume in `/var/tmp/lifvol` using **make_medialif**:

```
make_medialif -f /opt/ignite/data/Rel_B.10.20/config  
              -f /var/tmp/archive.cfg -f /var/tmp/depot_700_cfg  
              -f /var/tmp/depot_800_cfg -l /var/tmp/lifvol
```

This creates the LIF volume that includes all the configuration information. It includes the defaults Ignite-UX provides, as well as information on the archives and depot.

The configuration information stored in `INSTALLFS` must be changed. The two variables `run_ui` and `control_from_server` must be set using **instl_adm** to `TRUE` and `FALSE` respectively based on our scenario:

```
instl_adm -d -F /var/tmp/lifvol > /var/tmp/cfg  
vi /var/tmp/cfg      # Add/change the two variables  
instl_adm -F /var/tmp/lifvol -f /var/tmp/cfg  
instl_adm -d -F /var/tmp/lifvol # to make sure changes are there
```

Then using a DDS-1 density, no compression, no rewind device file (for example `/dev/rmt/c0t3d0DDS1n`) create the tape:

```
mt -t /dev/rmt/c0t3d0DDS1n rew  
dd if=/var/tmp/lifvol of=/dev/rmt/c0t3d0DDS1n obs=2k  
dd if=/var/tmp/archive_700.gz of=/dev/rmt/c0t3d0DDS1n obs=10k  
dd if=/var/tmp/serialdepot of=/dev/rmt/c0t3d0DDS1n obs=10k  
dd if=/var/tmp/archive_800.gz of=/dev/rmt/c0t3d0DDS1n obs=10k  
mt -t /dev/rmt/c0t3d0DDS1n rew
```

The tape is now ready for installations! Note that DDS-1 density is used so that the tape is more readily readable by all DDS tape devices, which are notorious for being finicky at times. To create a device with these characteristics:

```
ioscan -fC tape      # to get the hardware path
mksf -v -H <hardware path> -b DDS1 -n -a
```

SAM can also create this file for you. Bring up SAM and go to Peripheral Devices -> Tape Drives. Select (highlight) the tape drive you want to use. Go to Actions -> Create Device Files -> Create Custom Device File. Change DENSITY to DDS1, turn off Compressed Mode and Rewind on Close, and select OK.

CD-ROM layout

There are similarities between putting a CD-ROM together and putting a DDS tape together. One major difference however is in disk space usage. You have to create a logical volume (or provide a whole disk) large enough to hold the archives and the depots. This paper will assume that a logical volume is used. You need that much space again to copy the raw logical volume to a regular file. So you probably end up using around three times the disk space consumed by your archives and depots.

A bootable CD-ROM is not a serial device like a DDS tape. It has a file system on it, and it also has a LIF volume that contains the same information as above except for the config files which describe the archives and depots. Access to these objects is somewhat different.

The file system on the CD-ROM can be either HFS or CDFS. You can create an HFS file system using standard HP-UX commands. For a CDFS file system, there are various third party applications available. See Appendix C for a list of third-party software suppliers of CD writer software that runs on HP-UX. There is no implied or direct endorsement of any one package over another. Note that there is less capacity on a CD-ROM (650MB) than on a 90 meter DDS-1 tape (2GB).

Assume that the logical volume that will be used is `/dev/vg00/image`, and it is mounted at `/var/tmp/image`. Also assume that an HFS file system will be used. Using HFS and standard HP-UX commands, create the logical volume (assume everything fits in 500MB):

```
lvcreate -L 500 -n image vg00
newfs -F hfs -f 2048 /dev/vg00/rimage
mkdir -p /var/tmp/image
mount /dev/vg00/image /var/tmp/image
```

For CDFS file systems there are similar commands. Check the software supplier documentation.

Accessing an Archive on a CD-ROM

The file system in the logical volume will contain both the archives and the depots. The archives are placed in the CD-ROM image by copying them into the file system just created:

```
cp /var/tmp/archive_700.gz /var/tmp/image
cp /var/tmp/archive_800.gz /var/tmp/image
```

Creating config file information for the archives is similar to what was done for the DDS tape. Start with a new copy of `/opt/ignite/data/examples/core.cfg` in `/var/tmp/archive.cfg`.

To modify the config file to access the Series 700 archive, the following attributes need to be changed in `/var/tmp/archive.cfg` in the `sw_source` “core archive” stanza:

Attribute	Old Value	New Value
<code>source_type</code>	<code>"NET"</code>	<code>"DSK"</code>
<code>change_media</code>	<code># change_media=FALSE</code>	<code>change_media=FALSE</code>
<code>nfs_source</code>	<code>nfs_source = "<IP>:<depot>"</code>	<code># nfs_source = "<IP>:<depot>"</code>

These changes will modify the source type from a network (“NET”) access to CD-ROM (“DSK”). The other changes are as described before.

Inside the stanza enabled by `HARDWARE_MODEL ~ "9000/7.*"` the following fields must be changed:

Attribute	Old Value	New Value
<code>archive_path</code>	<code>"B.10.20_700_CDE.gz"</code>	<code>"archive_700.gz"</code>
<code>impacts</code>	<code>"/" 27Kb</code>	<code><as reported by archive_impact></code>

The change in `archive_path` indicates that the archive will be found in the pseudo-root of the CD-ROM in a file called `archive_700.gz`. Ignite-UX will prepend the mount point it uses to access the archive. As before, the correct set of `impacts` lines need to be included.

Again you can change the `sw_sel` and `description` strings to something more descriptive and applicable to your situation. The text inside the double quotes can be changed to whatever you like. Note that `archive_type` must match what was done by **make_sys_image**.

Delete the entire stanza called “golden image2” again.

In the stanza for the Series 800 (it is an else clause later in the file) the same sort of changes must be made. These attributes need to change:

Attribute	Old Value	New Value
<code>archive_path</code>	<code>"B.10.20_700_CDE.gz"</code>	<code>"archive_800.gz"</code>
<code>impacts</code>	<code>"/" 27Kb</code>	<code><as reported by archive_impact></code>

The change in `archive_path` indicates that the archive will be found in the pseudo-root of the CD-ROM in a file called `archive_800.gz`. Ignite-UX prepends the mount point it uses to access the archive. As before, the correct set of `impacts` lines need to be included.

It cannot be emphasized enough not to change anything else in the file.

Creating and Accessing the Depot on a CD-ROM

On a DDS tape, there is a restriction to a single depot. That restriction does not apply to a CD-ROM. However we will use a single depot for simplicity sake.

The depot is created using **swcopy** to a target in the logical volume:

```
swcopy -s /var/tmp/depot \* @ /var/tmp/image/depot
```

Once again **make_config** can create the start of the config files for the depot:

```
make_config -s /var/tmp/depot -a 700 -c /var/tmp/depot_700_cfg  
make_config -s /var/tmp/depot -a 800 -c /var/tmp/depot_800_cfg
```

Since the SD server is the system that is being installed, we can remove this attribute. Change both config files by removing this line:

Attribute
sd_server = "<IP>"

Change the following attribute in both files:

Attribute	Old Value	New Value
source_type	"NET"	"DSK"
sd_depot_dir	"var/tmp/depot"	"depot"

The change to `source_type` indicates that the depot is located on a CD-ROM instead of over the network. The change in `sd_depot_dir` indicates that the depot will be found in the pseudo-root of the CD-ROM in a depot called `depot`. Ignite-UX prepends the mount point it uses to access the depot.

Assembling the CD-ROM

The raw file system just created must be copied into a regular file so it can be written to the CD-ROM:

```
umount /var/tmp/image  
dd if=/dev/vg00/rimage of=/var/tmp/fs_image bs=1024k
```

Now that most of the components of the CD-ROM are complete, we can use **make_medialif** to create the LIF volume:

```
make_medialif -f /opt/ignite/data/Rel_B.10.20/config  
-f /var/tmp/archive.cfg -f /var/tmp/depot_700_cfg  
-f /var/tmp/depot_800_cfg -l /var/tmp/lifvol
```

This creates the LIF volume that includes all the configuration information. It includes the defaults Ignite-UX provides and provides the access to the archives and the depot.

The configuration information stored in the `INSTALLFS` needs to be changed as before. The sequence described before to set `run_ui` and `control_from_server` must be repeated here using **`instl_adm`**.

These two objects (the raw file system and the LIF volume) must be combined using **`instl_combine`**. The result is a single file with the LIF volume “wrapped around” the file system, which can then be written to the CD:

```
/opt/ignite/sbin/instl_combine -F /var/tmp/lifvol -C /var/tmp/fs_image
```

Then using the CD-ROM writer software you have, copy `/var/tmp/fs_image` to the CD.

The CD-ROM is now ready for installations! Note that you can test the image out before burning a CD if it is copied to an unused raw disk and the system is rebooted off that disk.

Media from `make_recovery`

There are many similarities between a DDS tape produced by **`make_recovery`** and one constructed by the method described herein. In fact, **`make_recovery`** performs many of the same steps. But there are some important differences as well. The primary purpose of a recovery tape is to restore enough of a system to get it going following some catastrophe, so the rest of the system can be recovered from backups. It is a fact that customers are using **`make_recovery`** for other purposes like system cloning. It does not have the flexibility provided in creating your own media, nor does it involve the possibility of using an Ignite-UX server.

Additional Archives and Depots Media

If there is insufficient space on either a DDS tape or a CD-ROM to hold all the archives and depots, it is possible to put them onto separate media. In this case, the config file describing these archives or depots would have `change_media` set to true. Ignite-UX would prompt the user for the new medium. If this is a CD-ROM, the **`instl_combine`** step is unnecessary.

Appendix A

This is the sequence of events that happen when a system boots up:

1. The firmware determines from which device to boot via either user input or primary path.
2. The firmware looks for a LIF header on that device, and if it finds it, it looks in the LIF header for where the ISL starts.
3. The firmware loads the ISL into memory from the boot device and executes it. It passes a flag to it that indicates whether to run interactively or to autoboot.
4. If the ISL is interactive then it gives the “ISL> ” prompt and waits for user input before proceeding.
5. If the ISL is not interactive, then it looks for the AUTO file on the boot device to determine what to run next.
6. Normally the AUTO file or user input supplies the command “hpux <args>”. This tells ISL to load the program HPUX from the LIF header on the boot device and to run it with the given arguments.
7. The “hpux” program (also known as the “secondary loader”) then figures out what HP-UX kernel to load, and what arguments to pass to it (like init state).
8. Then “hpux” loads the kernel and starts running it.
9. For the INSTALL kernel, the kernel looks at its name and realizes that it fits the pattern “*INSTALL” and turns around loads the matching “*INSTALLFS” file from the boot device.

In all these cases the firmware (Processor Dependent Code, or PDC) API services are used when accessing the boot device.

Appendix B

```
#####
##
## Filename:    core.cfg
##
## Description:
##   If the OS source is not an "SD" depot, the
##   add_release/make_config commands are not applicable. This file
##   gives an example of a config file that makes a core archive
##   available for installing using Ignite-UX.
##
##   To configure Ignite-UX to use an archive such as that created
##   by the make_sys_image(1M) script, you should do the following:
##
##   - Make an archive image of the desired system using the
##     script /opt/ignite/data/scripts/make_sys_image (See
##     make_sys_image(1M) for details).
##
##   - Determine if you want to use ftp, NFS, or remsh to access
##     the archive image, and locate the image on a server
##     as appropriate.
##
##   - Copy this config file to /var/opt/ignite/data/Rel_B.10.XX
##     (depending on the release you are working on).
##
##   - Edit the copy of this file and modify:
##     - The IP addresses of the server.
##     - The paths to archives.
##     - sw_sel descriptions.
##     - remove any extra sw_sel clauses.
##     - run /opt/ignite/lbin/archive_impact on each
##       archive and replace the "impacts" statements with
##       those given by the archive_impact tool.
##
##   - Edit the /var/opt/ignite/INDEX file. If you already
##     have a configuration in the INDEX file that refers to an SD
##     depot, then you will probably want to make a new "cfg"
##     clause that does not have references to the core SD config
##     files, and insert the path to this config file.
##     The end result should look something like this:
##
##         cfg "HP-UX B.10.20 via archive" {
##             description "10.20 system using archives."
##             "/opt/ignite/data/Rel_B.10.20/config"
##             "/opt/ignite/data/Rel_B.10.20/core.cfg"
##             "/var/opt/ignite/config.local"
##         }
##
##   - If you have additional archives or applications in a SD
##     depot that you would like loaded along with this archive,
##     you may create a config file for them and add it to the
##     INDEX file entry as well.
##
##     See make_config(1M) for creating config files for an SD
##     depot. See the example file "noncore.cfg" for setting up
##     application archives.
##
##   - After making changes, always run "/opt/ignite/bin/instl_adm -T"
##     to check the syntax.
##
#####
```

```

#####
## Software Sources
#####

sw_source "core archive" {
    description = "HP-UX Core Operating System Archives"
    load_order = 0
    source_format = archive
    source_type="NET"

    # When using this config file with a tape or CD-ROM where
    # the archive is on the same piece of media, then un-comment
    # the next line to prevent a prompt to change media during
    # install.
    # change_media=FALSE

    # The following 2 scripts should be reviewed to see if the
    # defined defaults are applicable.  If you need to modify
    # them, copy them to /var/opt/ignite/scripts first and change
    # the paths here.
    post_load_script = "/opt/ignite/data/scripts/os_arch_post_l"
    post_config_script = "/opt/ignite/data/scripts/os_arch_post_c"

    # If nfs_source is used, be sure to export the source.
    nfs_source = "15.1.54.123:/var/opt/ignite/archives"

    # ftp_source and remsh_source are alternate ways to copy
    # the archive.  An example of the ftp syntax:
    # ftp_source = "anonymous@15.1.54.123:iux"
    # remsh_source = "user@15.1.54.123"
}

#####
## Software Categories
#####

sw_category "Languages" {
    description = "Languages"
}

sw_category "HPUXEnvironments" {
    description = "HP-UX Operating Environments"
}

#####
## Operating Environments
##
#####

HARDWARE_MODEL ~ "9000/7.*" {
#####
## S700 OS archives
#####
init sw_sel "golden imagel" {
    description = "imagel: English HP-UX 10.20 CDE"
    sw_source = "core archive"
    sw_category = "HPUXEnvironments"
    archive_type = gzip tar

    # For NFS, the path to the archive is relative to the mount point
    # specified in the sw_source:
    archive_path = "B.10.20_700_CDE.gz"

    # ftp and remsh sources can use a full path:
    # archive_path = "/pub/IUXarchives/B.10.20_700_CDE.gz"
}

```

```

# The data for the "impacts" statements are found
# by running /opt/ignite/lbin/archive_impact
impacts = "/" 27Kb
impacts = "/.dt" 35Kb
impacts = "/etc" 1864Kb
impacts = "/export" 1Kb
impacts = "/home" 1Kb
impacts = "/opt" 74096Kb
impacts = "/sbin" 13449Kb
impacts = "/stand" 1Kb
impacts = "/tmp" 1Kb
impacts = "/users" 40Kb
impacts = "/usr" 225951Kb
impacts = "/var" 5705Kb
} = TRUE

init sw_sel "golden image2" {
description = "image2: English HP-UX 10.20 VUE"
sw_source = "core archive"
sw_category = "HPUXEnvironments"
archive_type = gzip cpio
archive_path = "B.10.20_700_VUE.gz"
impacts = "/" 421Kb
impacts = "/sbin" 30086Kb
impacts = "/opt" 78654Kb
impacts = "/usr" 276420Kb
impacts = "/var" 10059Kb
}

} else { # For S800's
#####
## S800 OS archives
#####
init sw_sel "golden image1" {
description = "image1: English HP-UX 10.20 CDE"
sw_source = "core archive"
sw_category = "HPUXEnvironments"
archive_type = gzip tar
archive_path = "B.10.20_700_CDE.gz"
# Example ftp or remsh path:
# archive_path = "/pub/IUXarchives/B.10.20_800_CDE.gz"
impacts = "/" 27Kb
impacts = "/.dt" 35Kb
impacts = "/etc" 1864Kb
impacts = "/export" 1Kb
impacts = "/home" 1Kb
impacts = "/opt" 74096Kb
impacts = "/sbin" 13449Kb
impacts = "/stand" 1Kb
impacts = "/tmp" 1Kb
impacts = "/users" 40Kb
impacts = "/usr" 225951Kb
impacts = "/var" 5705Kb
} = TRUE
}
}

```

```

#####
## Languages
## Specified so that the selector in the UI will
## not be empty. Cannot use the "core archive" sw_source
## because there is not archives associated with it.
##
## The "no select" sw_source is a no-op source since
## has is of "cmd" format, and does not have any scripts
## specified. Every sw_sel needs a sw_source.
##
#####
sw_source "no select" {
    source_format = cmd
}

init sw_sel "English" {
    description = "English Language Environment"
    sw_source = "no select"
    sw_category = "Languages"
    locale = { "SET_NULL_LOCALE:English", "C:English" }
} = TRUE

#####
## Keyboards
#####

_hp_keyboard = {
    "Not_Applicable",
    "PS2_DIN_US_English"
}
init _hp_keyboard = "PS2_DIN_US_English"

```


Appendix C

Table of some HP-UX CD writing software:

Company	Product Name
Elektroson	GEAR
Young Minds	CD Studio
TAMS	CD Recorder
Dataware Technology	CD Record
Kodak	Publishing Software
Creative DigitalResearch	HyCD

This information was gathered via a web search. There may be other products out there. No implied or expressed warranty and endorsement is assumed.