Ignite-LUX: Management and Integration of Ignite-UX Software on a Server Running Linux

Table of Contents

Introduction

The Ignite-LUX package enables network installation of HP-UX using a server that runs Linux.

The Ignite-LUX package includes portions of the Ignite-UX product that support installations initiated from a client's console. While this package does not include `ignite`'s functionality for server-controlled installations via GUI, `ignite` functionality for Central Management Server (CMS) integration may be used to perform client installations controlled by a Linux server. From the client's perspective, there are virtually no functional differences between an Ignite-UX server and an Ignite-LUX server.

This document details how to set up and manage Ignite-LUX content on a Linux server. Normally, the Ignite-LUX software will be installed and configured as part of the CMS software installation process.

Assumptions

It is assumed the reader is familiar with managing Ignite-UX. General Ignite-UX documentation should be consulted for creating config files and other details of Ignite-UX server management that are not covered in this document.

The Linux server must support the NFS and TFTP network protocols so HP-UX clients are able to access Ignite install environment and config file content.

The Ignite-LUX Package

A few Ignite-UX server commands are included in the Ignite-LUX package. These commands do not include all the same options and functionality provided by their counterparts in the HP-UX server Ignite-UX product. Generally, not all options are supported. In some cases, these commands support additional functionality and options not supported by the HP-UX Ignite-UX server commands. Command usage messages provide details regarding the options and features supported.

The Ignite-LUX package is released on the same schedule as the Ignite-UX product. Version numbers for both packages match, so Ignite-UX config file content may be shared or moved between servers. This also allows you to boot from one Ignite server and then switch to another Ignite server for installation. When CMS software includes Ignite-LUX, the most recent release is normally included. Consult CMS software documentation and support information about the processes for updating Ignite-LUX software.

Ignite-LUX is supported as part of your CMS software. Support requests should be sent through CMS support processes. Support processes normally used for HP-UX Ignite-UX server support should not be used. Any issue with HP-UX client software thatcan be reproduced on an HP-UX Ignite-UX server canbe processed under your HP-UX Ignite-UX support contract.

The Ignite-LUX package provides support for installation of Integrity systems; boot or installation of PA-RISC systems is not supported.

The Ignite-LUX package supports HP-UX installation but does not support installation of Linux, Solaris, Windows®, or any other operating system. Your CMS software might support deployment of these operating systems apart from the Ignite-LUX software.

The Ignite-LUX package may be installed on an HP-UX server to test its operation. However, this package is not supported on HP-UX servers for production use; you should use the Ignite-UX product on HP-UX servers.

The CMS

Use of the package is officially supported only when used with HP-provided Central Management Server (CMS) software, for example, HP Server Automation. CMS software might support servers running a specific Linux release. Also, CMS software might provide more limited functionality than presented in this document. You should consult your CMS documentation for additional integration and use details.

Archives and Depots

The Ignite-LUX server may be used to supply golden image archives and associated config files so the Ignite-LUX server supports complete HP-UX installation without the need for any other servers. Since the Software Distributor (SD) `swagent` daemon only runs on HP-UX systems, the Linux server cannot act as an HP-UX SD depot server. If desired, config file content on the Ignite-LUX server may be set up to use a separate HP-UX server that provides SD depots for installation.

An Ignite-LUX server may be used as a recovery server. The Ignite-LUX package includes content required to create a recovery image for a client system that has NFS write access to the Ignite-LUX server. CMS software might not support Ignite-LUX recovery functionality.

Linux Server Network Services Setup

The Ignite-UX config files and config file content placed on an Ignite-LUX server are the same as on an Ignite-UX server.

The Ignite-UX `instl_adm`, `manage_index,` and `ignite` commands are included in the Ignite-LUX package to aid config content creation and editing. However, these commands do not have all the same features as their counterparts in the Ignite-UX product. For example, the Ignite-LUX `instl_adm` command does not support the `–T` config file test option. Command usage messages include details regarding the options supported and functional differences.

If you are relatively new to Ignite-UX, you will want to consult Ignite-UX product documentation. Information provided in the *Ignite-UX Quick Start Guide* might be helpful. However, keep in mind that an Ignite-LUX server is unable to support HP-UX depot installation. For initial tests of the Ignite-LUX package you might want to set up a separate HP-UX SD server and use the `sd_server` config keyword to specify an alternate server for depot content. Refer to the `instl_adm(4)` man page and the *Ignite-UX Administration Guide* for further details.

Enabling Ignite-UX Network Boot

Network boot for HP-UX installation relies on DHCP, PXE, and `bootp` protocols. Configuration details for these protocol services are specific to the Linux server being used. If Ignite-LUX is included in your CMS software, Ignite-LUX configuration should be automatically done as part of the CMS software install.

DHCP `bootp` should be configured to provide appropriate UEFI PXE responses. The bootloader file name provided in the PXE response needs to be one of the following:

```
/opt/ignite/boot/nbp.efi
/opt/ignite/boot/Rel_B.11.23/nbp.efi
/opt/ignite/boot/Rel_B.11.31/nbp.efi
```

The file name will determine the `AUTO` file used to provide the boot menu and control default boot behavior.

Important: The `/opt/ignite/boot/AUTO` boot loader menu file is intended to be customized to provide desired client boot alternatives. Other release-specific `AUTO` files are used to automate boot for installation and should not be modified. Modifications to the release-specific AUTO files might interfere with CMS software operation and will be lost during package update.

The following directories should be set up for read-only access via TFTP:

```
/opt/ignite
/var/opt/ignite
```

By default, the Ignite-LUX package is configured to use NFS instead of TFTP whenever possible during the installation process. However, UEFI PXE network boot relies on the use of TFTP. If client system HP UEFI directed boot functionality is used, or you boot from a DVD and switch to the Ignite-UX server for installation, `bootp` and TFTP are not required. You should consult HP UEFI documentation and the *Ignite-UX Administration Guide* for further details.

Enabling HP-UX Installation

You must configure the NFS server to allow access to required Ignite-UX directories. Depending on how you intend to use the Ignite-UX server and how you handle per-client access, you will need to customize the set of directories exported.

```
/opt/ignite                                  (read-only)
/var/opt/ignite/data                         (read-only)
/var/opt/ignite/config                       (read-only)
/var/opt/ignite/archives                     (read-only)
/var/opt/ignite/scripts                      (read-only)
/var/opt/ignite/clients                      (read/write)
/var/opt/ignite/recovery/archives/sys1       (read/write)
```

Important: Some Linux NFS servers limit maximum file size to 2GB or 4GB. Ignite-UX archive files typically will be larger than 2GB and might be larger than 4GB. The Linux NFS server must be configured to support large files. This might require a newer version of NFS; for example, version 3 or later.

Creating and Using Golden Images

The Ignite-LUX server may be used to support HP-UX installation of golden images and recovery archives without requiring an HP-UX server.

The *Ignite-UX Administration Guide* provides details regarding creation and use of golden images. However, there are some special considerations for an Ignite-LUX server.

The Ignite-LUX package includes the same `make_sys_image` command as included in the Ignite-UX product. However, the `make_sys_image` command `–s` option relies on HP-UX-specific server functionality.

You may use NFS to mount an Ignite-LUX server directory so the client's local file system canbe used to write the client's operating system image directly to the server. For example, the following commands could be performed on the client system with 10.3.1.2 being the Ignite-LUX server:

```
# mkdir /tmp/archives
# mount 10.3.1.2:/var/opt/ignite/archives /tmp/archives

# mkdir /tmp/scripts
# mount 10.3.1.2:/opt/ignite/data/scripts /tmp/scripts

# /tmp/scripts/make_sys_image -s local -d /tmp/archives -n B.11.31_archive_IA.gz
```

Hewlett-Packard recommends using `/tmp` paths to avoid having this content included in the archive that is created.

Once archive creation has been completed, you need to determine archive disk space impacts and include them in the golden image config file. The `archive_impacts` command should be used to compute archive impacts.

Note: The `archive_impacts` command is not included in the Ignite-LUX package at this time. You will need to use the following alternative approach to determine archive impacts.

Use an Ignite-UX server to create the golden image and determine the impacts. Once the archive and config content are correct, they may be copied to the Ignite-LUX server.

Install Ignite-UX on the client system after the operating system image has been created, and then run the `archive_impacts` command on the client system. Since the archive is already created, you do not need to include Ignite-UX in the client golden image.

While not recommended, you may elect to estimate archive impacts based on experience. This approach might make sense when minor changes are made to archive content.

Like the HP-UX Ignite-UX product, the Ignite-LUX package includes example config files:

```
/opt/ignite/data/examples/B.11.23.archives.cfg
/opt/ignite/data/examples/B.11.23.golden_image.cfg
/opt/ignite/data/examples/B.11.31.archives.cfg
/opt/ignite/data/examples/B.11.31.golden_image.cfg
```

Remember to add archive disk space impacts to your golden image config file.

```
# /opt/ignite/lbin/archive_impact -tg B.11.23_archive_IA.gz
```

Special HP-UX Client Considerations

This document does not include information about detailed HP-UX client configuration. You should consult other HP-UX documentation regarding how to set up your client in your data center environment. The information presented here describes how to use your configurations with an Ignite-LUX server.

Client Console Configuration

Integrity EFI supports administrator interaction using any console when multiple console interfaces are enabled. Many OS's also support this capability. The HP-UX OS only supports the one console configured as the primary interface. If you are using another console to boot the client system, it might look like the HP-UX kernel has hung during boot. The HP-UX boot loader writes a message to all consoles prior to the HP-UX OS boot regarding the issue. If the message stays on the screen, switch to the correct console and consider if another interface should be configured to act as the primary console.

Client Network Configuration

Networking requires special consideration. Some HP-UX OS software depends on name service use for the client system to lookup its own IP address. You will need to configure a DNS server or change the HP-UX name services configuration to favor files on the client system. If the client system is unable to lookup its own IP address, you will experience significant delays during system boot.

To configure the client to favor files for name services you will need to modify `/etc/nsswitch.conf` and `/etc/hosts`.

```
# /etc/nsswitch.conf:
passwd:       compat
group:        compat
hosts:        files dns [NOTFOUND=return] nis [NOTFOUND=return]
ipnodes:      files dns [NOTFOUND=return]
networks:     files nis [NOTFOUND=return]
protocols:    files nis [NOTFOUND=return]
rpc:          files nis [NOTFOUND=return]
publickey:    files nis [NOTFOUND=return]
netgroup:     files nis [NOTFOUND=return]
automount:    files nis
aliases:      files nis
services:     files nis [NOTFOUND=return]
```

This example is meant to highlight the general issue. Consult networking documentation for details.

If you use dynamically assigned IP addresses via DHCP, you should be aware that the `auto_parms` command will edit `/etc/hosts` during system boot to add or update the client system's entry.

Client-Specific Server Content

Ignite-UX saves client-specific content in the directory

```
/var/opt/ignite/clients/<MAC_address>
```

To aid usability, Ignite-UX creates a symbolic link for the client hostname at:

```
/var/opt/ignite/clients/<hostname>
```

Warning: Features in recent Integrity systems might result in duplicate MAC address for different client systems. Use of HP Virtual Connect (VC) functionality is one example of how this issue might occur. Clearly, you would not expect duplicate MAC addresses on the same subnet. However, the same VC MAC address for clients on different subnets can be used successfully for typical network operations.

Ignite-UX uses MAC addresses as the method for differentiating client systems. Clients with duplicate MAC addresses will see each other's client-specific content on the Ignite-UX server. In addition, the hostname link will generally point to the same client-specific directory.

You should manage your network configuration so an Ignite-UX server does not have multiple clients with the same MAC address.

Server-Based Recovery

An Ignite-LUX server may be used as an HP-UX network recovery server. The Ignite-UX product must be installed on the client system. The versions of Ignite-UX installed on the server and the client must match. Ignite-UX uses a consistent version numbering scheme for the Ignite-UX product and the Ignite-LUX package.

Important: Your CMS might not include support for managing Ignite-UX recovery. There are various technical approaches for client system recovery, including: reinstallation, mirroring, Dynamic Root Disk (DRD), and storage. Consult your CMS documentation for specific solution information. Alternatively, you canelect to manage Ignite-UX recovery outside CMS control.

You may use the Ignite-UX `make_net_recovery` command to save a client system recovery archive on the Ignite-LUX server. The command runs on the HP-UX server and should operate as if the Ignite-LUX server were an HP-UX system.

The Ignite-LUX server must provide NFS access for the following client-specific directory:

```
   /var/opt/ignite/recovery/archives/<hostname>        (read/write)
```

The recommended best practice is to restrict NFS access to the specific client.

We recommend creating the client-specific recovery archive directory as follows:

```
   # mkdir -p /var/opt/ignite/recovery/archives/<hostname>
   # chown bin:bin /var/opt/ignite/recovery/archives/<hostname>
```

If the `make_net_recovery` command cannot mount the client directory, an error message will be given. The details in this error message describe how to resolve the issue on an HP-UX server. You will need to adapt details for changing the NFS configuration for your Linux server.

Review Ignite-UX documentation for the use of `make_net_recovery`. The following is a simple command line example to save the entire root volume group in a recovery archive:

```
# /opt/ignite/bin/make_net_recovery -s <server> -A
```

Recovery of the client system is done by reinstalling the client system recovery archive. There is a client-specific `CINDEX` file for client-specific recovery archive configuration clauses, so clients do have access to the recovery archives of other clients.

Saving multiple recovery archives for several systems will require considerable file system space on the server. A rough estimate for the size of recovery archives is 4GB to 8GB for HP-UX OS content. An Ignite-UX recovery archive is intended to include only HP-UX OS content and the application programs required to restore backup content saved using something other than Ignite-UX. Archives might be very large depending on what is included.

The Ignite-LUX Package Installation and Configuration

The Ignite-LUX package is supplied in the form of a compressed `tar` archive. This archive includes parts of the Ignite-UX product typically installed on an HP-UX Ignite-UX server. Most portions of Ignite-UX held on the HP-UX Ignite-UX server but run on a client system are included in the package and are identical to their counterparts in the Ignite-LUX product. Almost none of the Ignite-UX product software that normally runs on an HP-UX Ignite-UX server is included in this package. Those programs would not run correctly on a Linux system.

Note: This document does not include information about how to obtain the Ignite-LUX package. You should receive the package as part of the CMS product.

The procedure for installation and configuration of the Ignite-LUX software is significantly different from the procedure used for an HP-UX server.

Important: If your CMS software installation process performed the Ignite-LUX package installation and configuration, the steps presented in this section should not be performed. Your CMS software might rely on a special configuration that differs from the details presented here.

If you must install the Ignite-LUX package yourself, perform the steps presented here to ensure correct Ignite-UX server operation.

This procedure may be used for initial package installation or to update to a newer version of the package. When the package is reinstalled to perform an update to a newer version, `INDEX`, `IINSTALLFS` config content, and other normal Ignite-UX server customizations will be preserved.

You must be a privileged user on the Linux server to install the package. The package is provided as a `gzip` compressed `tar` archive. You should consult your OS documentation for details of `tar` and `gzip` command use to unpack a compressed `tar` archive file.

The archive uses relative file paths. Change your working directory (`cd`) to the correct directory before unpacking the archive. The archive is normally unpacked in the root (/) directory. Relative file paths allow installation to an alternate location if needed for a `chroot` or other special setup. If the package must be installed in an alternate location, it is likely symbolic links will need to be provided so commands run on the Linux server operate correctly. For example:

```
# ln –s /opt/opsware/boot/tftpboot/opt/ignite/boot /opt/ignite/boot
```

When possible these links should be created before package installation. If links are set up before installation you should unpack the package in the root directory:

For example:

```
# cd /
# tar xzf Ignite-UX-C-7-8.201.tar.gz
```

Or:

```
# cd /
# gunzip < Ignite-UX-C-7-8.201.tar.gz | tar xf -
```

When the Ignite-LUX product is installed using the SD `swinstall` command on an HP-UX Ignite-UX system, package configuration is automatically done. The Ignite-LUX package includes a program that performs necessary package configuration. This program must be manually run to complete package installation or update:

```
# cd /
# /opt/ignite/bin/setup/configure
```

The first time the Ignite-LUX package is installed, the configure program will attempt to examine the network configuration of your server and generate appropriate `IINSTALLFS` config content. After the configuration is completed, you should validate these settings and modify them if necessary using the `instl_adm` command:

```
# /opt/ignite/bin/instl_adm
```

When the package is reinstalled to update to a new version, `IINSTALLFS` config content is retained without modification.

The package installs several files named `*.new` that allow the configuration process to retain Ignite-UX server customizations. Some of these files are left on the server after configuration to act as a reference for default file contents.

Warning: If the configuration is incomplete, the Ignite-LUX server will not function correctly to support HP-UX install. Client systems might be unable to boot the Ignite-UX install environment, crash or hang during attempted installation, or fail to correctly install in some other manner.

CMS Integration with Ignite-LUX Software

The following sections are primarily intended to provide CMS developers with Ignite-LUX integration details. CMS solution users might find this information useful for understanding Ignite-UX interaction, maintaining their Ignite-LUX server, and assisting issue diagnosis.

HP-UX Ignite-UX and Linux Ignite-LUX servers include compatible CMS integration support with minor differences. It is possible to use the same implementation for control of both types of servers.

CMS software generally uses the same Ignite-UX interface as the Ignite-UX software that runs on the HP-UX server to control clients.

The *Ignite-UX Administration Guide* includes a section on Ignite-UX theory of operation that CMS developers might find especially useful. The guide also includes information on handling complex networks that is very likely to contain important, helpful details.

Caution: CMS developers and others performing automated control of Ignite-LUX functionality should only use the methods and details presented in this section. Ignite-UX provides a rich set of features. Experience has shown that exploring Ignite-UX capabilities and applying integration creativity can result in software that does not provide a good overall solution later. For example, in some cases developers have used automatically generated per-client config content placed in `IINSTALLFS` that worked well for limited testing but prevented successfully scaling to a full solution.

Important: The Ignite-UX team might make product changes in the future releases that would break CMS integration or other automated control that includes use of functionality not described here.

If functionality is required beyond what is presented here, you must request an integration enhancement so the required functionality is added to the package if necessary and becomes documented in this section.

Future: Where noted, some of the interface details presented in this section have not yet been implemented. This information about future plans reflects the Ignite-UX team’s perspective but is not a commitment.

We recommend that CMS software developers consider developing to use these future planned interfaces when present since when implemented they provide a better and more stable means of integration.

In addition to other future items in this section, the Ignite-UX team is investigating a Command Line Interface (CLI), which makes use of XML for data interchange as an improved interface for Ignite-UX control. Due to customer compatibility needs and other issues, the interface details included in this section will not be immediately replaced by an improved interface.

Ignite-LUX includes two approaches for CMS integration. The `ignite` command improves control by using XML for data interchange and is the preferred approach. . Legacy Ignite commands such as `manage_index` and `instl_adm` may also be used for CMS control. Both approaches may be used.

Terminology

HP-UX uses different release identifiers for customer marketing communication and technical product implementation. HP-UX 11i v3 is also known as HP-UX 11.31; HP-UX 11i v2 is also known as HP-UX 11.23.

Deployment software and CMS software seem to have fairly similar concepts. However, there are considerable differences in the terms used. For example, Ignite-UX uses the term "configuration clause" to refer to the combination of the set of software to be installed and the config file content used to control installation.

You may choose exactly one configuration clause for installation. In the case of an automated installation, that configuration clause controls client installation details. In the case of an interactive installation, that configuration clause provides defaults that may be modified prior to installation via the Ignite-UX user interface on the client system. The set of configuration clauses is stored in `/var/opt/ignite/data/INDEX`.

The *Ignite-UX Administration Guide* includes a glossary of terms.

Ignite-UX Concepts

Universal install kernels - The same Ignite-UX install kernels and install file systems are used for all Integrity systems: rack-mount, Blade, HPVM, nPar, and vPar.

Install kernel and installed software consistency - You must boot the install kernel that matches the HP-UX release to be installed. For example, to install HP-UX 11.31 you must boot the HP-UX 11.31 install kernel.

Config-based control - Ignite-UX performs an installation based on config file content, which reflects user interface settings, is able to specify considerable system configuration detail, supports formulas, and supports conditional statements. This is Ignite-UX's equivalent of what is often referred to as install scripts by other OS deployment.These config file features are helpful in constructing:

- golden images that may be used for a variety of systems
- cloned systems
- recovery images for disaster recovery

OS content for hardware support - You should consider the software included for installation that supports hardware and virtualization. The recommended approach is to include the full set of drivers needed for all systems. This approach allows you to create universal OS images.

Configuration adaptation to hardware - Ignite-UX will significantly adapt configuration details to the system being installed. Recovery will restore a system to a nearly original state. A recovery image may be used to recover a very dissimilar type of system. For example, Ignite-UX will find replacement disks if needed. Consult Ignite-UX config file content details in the instl_adm(4) manpage for information regarding how to control Ignite-UX behavior.

Client boot and deployment server control - You will need to consider how new client systems will be recognized and set up for CMS-controlled Ignite-UX installation support. CMS software can take a boot-first or configure-first approach, or support both. Client systems use NFS to mount a client-specific directory if networking is enabled and an Ignite-UX server is specified. When the Ignite-UX install environment starts up, one of the first things that happens is a client system inventory. That inventory data are written to the server via NFS.

Networking configuration - You must also decide if static or dynamic IP addresses will be used for your solution. Ignite-UX software supports both. Depending on the approach, you will want to include config file content in `IINSTALLFS` to correctly set up client networking when Ignite-UX starts. During initial installation configuration, the Ignite-LUX package places content in the `IINSTALLFS` file to handle a dynamic IP address approach.

Note: Hewlett-Packard strongly recommends installing all HP-UX OS content on one LUN (with the obvious exception of mirroring). Existing HP-UX Dynamic Root Disk (DRD) capabilities rely on this type of system configuration. Additional future HP-UX OS image management features are likely to rely on the same type of single LUN OS content configuration.

Client-Specific Server Content

Ignite-UX uses several files to control clients and provide client information. When possible, all client control and monitoring should be done using Ignite-LUX commands rather than direct file access.

Ignite-UX saves client-specific content in the directory

```
   /var/opt/ignite/clients/<MAC_address>
```

To aid usability, Ignite-UX creates a symbolic link for the client hostname at

```
   /var/opt/ignite/clients/<hostname>
```

Ignite-LUX servers have the following client-specific files that may be used for CMS integration:

`CINDEX`	`(client-specific config clauses)`
`client_name`	`(client hostname)`
`client_status`	`(client installation status)`
`config`	`(config file used to control installation)`
`config.sys`	`(config file content used to perform installation)`
`env.vars`	`(environment variables set in config content)`
`host.info`	`(client system characteristics)`
`install.log`	`(detailed IINSTALL log)`
`io.info`	`(client system I/O inventory)`
`server.instr`	`(instructions for client system from server)`

Other files should not be used. For example, the directory might include an `hw.info` file, which you might find easier to parse than `io.info`. But, `hw.info` is being replaced with `io.info` and might not be created in a future Ignite-UX release.

Client-Specific File: CINDEX

This file specifies configuration clauses that are available only for the associated client. Typically, these are network recovery configuration clauses. Thus, the set of configuration clauses valid for a client system include those listed in the general INDEX config file and the client-specific CINDEX config file.

Client-Specific File: client_name

This file indicates the hostname of the client system.

For example:

```
# cat client_name
testsys
```

Client-Specific File: client_status

This file indicates the status of the current or most recent installation.

This file contains lines that have a status "phase" followed by "Active" or "Complete". Note that some phases might not have an "Active" status line. The file may have consecutive "Complete" status lines. The "DONE" line indicates a complete installation; no additional status will be reported.

For example:

```
# cat client_status
Boot_Client/Discover_System     Complete
Waiting_To_Install      Complete
Prepare_Config_File     Complete
Configure_Disks Active
Configure_Disks Complete
Download_mini-system    Active
Download_mini-system    Complete
Loading_software        Active
Loading_software        Complete
Build_Kernel    Active
Build_Kernel    Complete
Boot_From_Client_Disk   Active
Boot_From_Client_Disk   Complete
Run_SD_Configure_Scripts        Active
Run_SD_Configure_Scripts        Complete
Run_Postconfigure_Scripts       Active
Run_Postconfigure_Scripts       Complete
DONE
```

The Ignite-UX client I/O inventory phase is done independently from the server, since an inventory must be done in order to perform network configuration. The client requires network configuration to NFS mount the client-specific directory on the server and start reporting status.

For server-controlled installation, the client will be in the "Waiting_To_Install Active" phase. For example:

```
# cat client_status
Boot_Client/Discover_System     Complete
Waiting_To_Install      Active
```

Warning: If a client installation or recovery fails, Ignite-UX might return to the `Waiting_To_Install` phase to get instructions from the server on how to proceed.

The Ignite-UX `ignite` command provides a user interface that uses more understandable phase names than the ones listed above. However, we recommend you do not attempt to provide a similar mapping with CMS software. Any improvement is minimal compared to the risk of a broken CMS integration because of future Ignite-UX changes.

Client-Specific File: `config`

The `config` file is used to specify the installation configuration settings supplied by CMS software that are client-specific. It might include the `INDEX` configuration clause to be used, the root disk, and any other Ignite-UX config content (see `instl_adm(4)`).

```
cfg "HP-UX B.11.31 Default"=TRUE
_ip_cfg_detail_levels="ivsp"
init _hp_root_disk="WWID='0x5000c50005e33057'"
```

Note: The Ignite-UX `_hp_cfg_detail_level` variable indicates the type of config content in the file. This variable allows Ignite-UX to optimize config content processing. You should consult `instl_adm(4)` for details regarding correct values.

Client-Specific File: `config.sys`

At the phase `Prepare_Config_File Complete`, this is the config file content that will be used for installation. Prior to that phase, the file may be used to hold partial and temporary config file content.

Client-Specific File: `env.vars`

At the phase `Prepare_Config_File Complete`, this file holds environment variable values. Prior to that phase, the file may be used to hold partial and temporary environment variable values.

```
# cat env.vars
SOURCE=10.3.1.2
SOURCE_TYPE=NET
NET_STATUS=UP
TERM=unknown
INST_CLIENT_DIR=/var/opt/ignite/clients/0x001083049436
INST_LOG_FILE=/var/opt/ignite/clients/0x001083049436/install.log
INST_SERVER_MOUNTED=1
TZ=MST7MDT
INST_ALLOW_WARNINGS=5
```

Client-Specific File: host.info

The host.info file is created as a result of the system inventory done during Ignite-UX client boot.

The inventory collects various details of the client system and the Ignite-UX install environment. The file is formatted as an Ignite-UX config file.

```
# cat host.info
MEMORY=4173204K
HARDWARE_MODEL="ia64 hp "
MODEL="ia64 hp server rx2660"
_hp_ikernel_os_release="B.11.31"
_hp_ikernel_os_release visible_if FALSE
can_run_32bit=FALSE
can_run_64bit=TRUE
is_numa=FALSE
is_ia64=TRUE
is_hppa=FALSE
is_ht_capable=FALSE
_hp_boot_dev_path="0/1/2/0"
_hp_boot_dev_path visible_if FALSE
init _hp_default_cur_lan_dev="lan0"
_hp_default_cur_lan_dev visible_if FALSE
_hp_default_cur_lan_dev="0/1/2/0"
server="10.3.1.2"
is_net_info_temporary=FALSE
system_name="hpuxsys1"
ip_addr[]="10.3.1.101"
netmask[]="0xffffff00"
dns_domain="fc.hp.com"
dns_nameserver[0]="15.1.48.11"
nis_domain="lab"
wait_for_nis_server=TRUE
ntpdate_server="10.3.1.2"
route_gateway[0]=" 10.3.1.1"
route_destination[0]="default"
SOURCE_TYPE="NET"
_hp_primary_path="0/1/1/0.0x5000c50005e33055.0x0"
_hp_primary_path visible_if false
```

Client-Specific File: hw.info

The hw.info file is being deprecated. No hw.info file content should be used. The io.info file should be used for all I/O inventory details.

Client-Specific File: install.log

The install.log file captures detailed log messages during installation or recovery. The file might include a mixture of Ignite-UX messages and output from various commands run during installation or recovery. Thus, this file may have content of any format.

CMS software should generally only look for the Ignite-UX messages: ERROR, WARNING, CAUTION, NOTE, PHASE, and *. Ignite-UX messages indicated by the * character are also known as "OTHER" messages.

The OTHER type messages indicate installation or recovery processing.

If present, PHASE messages will match client_status values. For example:

```
PHASE: Build_Kernel Complete
```

CMS software may do processing based on the type and number of messages. However, CMS software should not implement any functionality that recognizes specific messages.

Longer Ignite-UX messages might be continued on subsequent install.log file lines.

Example install.log file content:

```
       * Configuring RAM filesystems...
NOTE:    Ignite-UX will use NFS for loadfile.
       * Number of SAS devices swapped to be in physical location order: 1
       * Scanning system for IO devices...
       * Boot device is: 0/1/2/0
       * Bringing up Network (lan0)
add net default: gateway 15.1.48.1
       * NFS mounting source /opt/ignite/data directory.
       * NFS mounting source /var/opt/ignite/data directory.
       * Reading configuration information from server...
       * NFS mounting clients directory.
       * Using client directory: /var/opt/ignite/clients/0x0017A4AB4449
NOTE:    Ignite-UX is ready to install the client.  Start the installation from
         the "ignite" user interface using the menu: "Action" -> "Install
         client" -> "New Install"
NOTE:    Starting install without a client specific config file, all defaults
         will be used.
       * Checking configuration for consistency...
ERROR:   The version of HP-UX you have chosen to install on the system
         (B.11.23) is not supported by the version of the Ignite-UX install
         kernel that the system booted (B.11.31).  You will need to reboot the
         target system from an install kernel matching the desired release from
         the menu at the console. If using the bootsys command, use the '-R'
         option to specify the install kernel version.
WARNING: The disk at: 0/1/1/0:SAS:ENC1:BAY07 (HP_DG072ABAB3) appears to contain
         a file system and boot area.  Continuing the installation will destroy
         any existing data on this disk.
WARNING: The root password has not been set.  If you wish to set the root
         password now, return to the System tab and do so.  Not setting the
         root password is considered a security risk.  Since the hostname for
         the system has already been set, set_parms will not run during the
         initial boot process.  If you do not set the root password now, you
         will need to wait until the system completes its boot process and
         either set the root password directly or with sam(1M).
```

Warning: It is common for certain Ignite-UX WARNING messages to be issued during a normal installation, such as for disks already having recognized file system content.

Client-Specific File: io.info

The io.info file is created as a result of the system inventory done during Ignite-UX client boot.

The io.info file should be used to get any system I/O inventory details. For example, it may be used to determine the identifiers and sizes of disk devices to be presented in a CMS UI for user selection. After selection in the CMS UI, the identifier values for selected disks may be used in Ignite-UX config file content to control installation.

This file includes records of type: file version, comments, field names, and inventory data. Field name records should be used to parse inventory data so CMS software will be able to cope with possible future format changes. The file version number will change when there is a format change.

Comment lines start with a '#' character. Other record lines have a specific record type label at the start of the line prior to the ':' character. Record fields are separated by a space character. Special characters in field values are escaped using the '\' character. The character sequence '[\]' is used to indicate there is no value for the field (a null value).

The file version record (io.info) includes the specific file format version. In this example it is 2.1:

```
io.info: 2.1 B.11.31 C.7.9.140
```

If the Ignite-UX server content has been updated, it is possible that io.info content from a prior installation uses a format that is not recognized by CMS software. If parsing has any issues with the io.info version, the best approach is to remove all *.info files. These inventory files will be recreated during Ignite-UX install environment boot on the client system.

Each io.info file will include *_fields records that describe all the various inventory data records that are supported by the io.info file format. These records indicate the order and meaning of field values in associated inventory data records. CMS software doing io.info file parsing must use *_fields records to correctly process inventory data to provide a robust method for data processing.

```
# cat io.info
io.info: 2.1 B.11.31 C.7.9.140
# Created: Sat Apr 4 15:59:56 2009
OO_fields: hw_path instance driver model state_flags
diskid_fields: unique_name wwid phys_loc c_major b_major minor c_device b_device
state_flags instance driver model capacity format usage alias device_id performance
device_type lun_hw_path
diskp_fields: unique_name wwid c_major b_major minor c_device b_device state_flags
instance driver hw_path protocol locality priority leg_driver leg_hw_path
leg_instance
escsi_ctlr_fields: hw_path instance driver model state_flags
ext_bus_fields: hw_path instance driver model state_flags
fc_fields: hw_path instance driver model state_flags
graphics_fields: hw_path instance driver c_device model crt_x crt_y crt_planes
crt_id
iscsi_hl_fields: hw_path instance driver model state_flags
```

```
lan_fields: hw_path instance driver c_device model lla ppa
processor_fields: hw_path instance driver model
ps2_fields: hw_path instance driver c_device model
tapeid_fields: unique_name wwid phys_loc c_major b_major minor c_device b_device
state_flags instance driver model capacity format usage alias device_id performance
device_type lun_hw_path
tapep_fields: unique_name wwid c_major b_major minor c_device b_device state_flags
instance driver hw_path protocol locality priority leg_driver leg_hw_path
leg_instance
usb_fields: hw_path instance driver model state_flags
usb: 0/0/2/0 0 hcd NEC_OHCI_Controller 0
usb: 0/0/2/1 1 hcd NEC_OHCI_Controller 0
usb: 0/0/2/2 2 ehci NEC_EHCI_Controller 0
graphics: 0/0/3/0 0 gvid_core /dev/crt0 PCI_Display_(1002515e) 0 0 0 0
escsi_ctlr: 0/1/1/0 1 sasd HP_PCI/PCI-X_SAS_MPT_Adapter 1
diskid: 0x01.0x00.0x03.0x5000c50005e2b8d3 0x5000c50005e2b8d3 SAS:ENC1:BAY08 13 1
0x2 /dev/rdisk/disk2 /dev/disk/disk2 3 2 esdisk HP_DG072ABAB3 71687369 lvm [\] [\]
[] [\] disk 64000/0xfa00/0x2
diskp: 0x01.0x00.0x03.0x5000c50005e2b8d3 0x5000c50005e2b8d3 188 31 0x11000
/dev/rdsk/c1t1d0 /dev/dsk/c1t1d0 1 2 eslpt 0/1/1/0.0x5000c50005e2b8d1.0x0 sas 5 [\]
sdisk 0/1/1/0.0.0.1.0 1
diskid: 0x01.0x00.0x03.0x5000c50005e33057 0x5000c50005e33057 SAS:ENC1:BAY07 13 1
0x3 /dev/rdisk/disk3 /dev/disk/disk3 3 3 esdisk HP_DG072ABAB3 71687369 lvm [\] [\]
^F, [\] disk 64000/0xfa00/0x3
diskp: 0x01.0x00.0x03.0x5000c50005e33057 0x5000c50005e33057 188 31 0x10000
/dev/rdsk/c1t0d0 /dev/dsk/c1t0d0 1 3 eslpt 0/1/1/0.0x5000c50005e33055.0x0 sas 5 [\]
sdisk 0/1/1/0.0.0.0.0 0
lan: 0/1/2/0 0 igelan /dev/lan0 HP_PCI-X_1000Base-T_Dual-port_Built-in 0017A4AB4449
0
lan: 0/1/2/1 1 igelan /dev/lan1 HP_PCI-X_1000Base-T_Dual-port_Built-in 0017A4AB4448
1
fc: 0/3/1/0 0 fcd HP_AB379-60101_4Gb_Dual_Port_PCI/PCI-
X_Fibre_Channel_Adapter_(FC_Port_1) 1
tapeid: 0x01.0x00.0x03.0x50014380018b6174 0x50014380018b6174 [\] 14 [\] 0x1
/dev/rtape/tape1_BEST [\] 1 1 estape HP_Ultrium_3-SCSI [\] [\] [\] [\] [\] [\] tape
64000/0xfa00/0x1
tapep: 0x01.0x00.0x03.0x50014380018b6174 0x50014380018b6174 205 [\] 0x2000
/dev/rmt/c0t2d0BEST [\] 1 1 eslpt 0/3/1/0.0x50014380018b6175.0x0 fibre_channel 5
[\] stape 0/3/1/0.8.0.255.0.2.0 0
fc: 0/3/1/1 1 fcd HP_AB379-60101_4Gb_Dual_Port_PCI/PCI-
X_Fibre_Channel_Adapter_(FC_Port_2) 1
processor: 120 0 processor Processor
escsi_ctlr: 64000/0x0/0x0 0 usb_ms_scsi USB_Mass_Storage_Virt_Ctlr 1
diskid: iux-0000-0001 [] [\] 13 1 0x4 /dev/rdisk/disk6 /dev/disk/disk6 1 6 esdisk
TEAC_DV-28E-N 0 [\] [\] [\] [] [\] cdrom 64000/0xfa00/0x4
diskp: iux-0000-0001 [] 188 31 0x20000 /dev/rdsk/c2t0d0 /dev/dsk/c2t0d0 1 4 eslpt
64000/0x0/0x0.0x0.0x0 usb 4 [\] sdisk 255/1/0.0.0 4
diskid: iux-0000-0002 [] [\] 13 1 0x5 /dev/rdisk/disk7 /dev/disk/disk7 1 7 esdisk
TOSHIBA_TransMemory 0 [] [\] [\] [] [\] baddisk 64000/0xfa00/0x5
diskp: iux-0000-0002 [] 188 31 0x21000 /dev/rdsk/c2t1d0 /dev/dsk/c2t1d0 1 5 eslpt
64000/0x0/0x0.0x1.0x0 usb 4 [\] sdisk 255/1/0.1.0 5
```

Client-Specific File: `manifest`

The `manifest` file is intended to be a human-readable report of how the client system was installed. CMS software should not parse this file since Ignite-UX software might make changes in the future.

Client-Specific File: `server.instr`

The `server.instr` file is used to control client operation when a client is booted into Ignite-UX control from the server. CMS software may write strings into this file. The Ignite-UX client periodically polls this file to get server instructions.

The following control strings may be used:

```
start_install    (initiate install or recovery)
reboot_client    (interrupt install and reboot the system)
recovery_shell   (interrupt install or recovery and launch diagnostic shell)
halt_client      (interrupt install or recovery and shutdown system)
```

Client Installation and Recovery Control Operations

There are a number of operations that are typically necessary for a CMS to control HP-UX installation. This section provides some best practices for implementing integration for these operations.

One of the high-level design considerations is the exact sequence of client and server operations controlled by the CMS software. There are two ways to approach the client/server boot and control synchronization problem:

1. Live client system inventory data are used with server controlled installation. This may be done via the following sequence of operations:
    a. The client system is booted manually or by CMS control, and the Ignite-UX install environment is started. A client system inventory is done and reported to the server. The client waits for server-controlled installation.
    b. CMS software uses the client inventory data to create an Ignite-UX config file that will perform needed selections and provide other configuration details.
    c. CMS software allows the client system to proceed with installation using the supplied config.
2. Previously collected inventory data are used to control non-interactive installation. This may be done as follows:
    a. CMS software already has any needed client system inventory data via some process. That process may be unrelated to Ignite-UX or may be the result of a previous use of Ignite-UX.
    b. CMS software uses the client inventory data to create an Ignite-UX config file that will perform needed selections and provide other configuration details.
    c. The client system is booted manually or by CMS control and the Ignite-UX install environment is started. Upon boot, the client immediately initiates installation using the config content previously created by the CMS software.

Either approach may be used; they may even be intermixed on the same Ignite-UX server. Care is required if independently collected client inventory data are used so values match Ignite-UX config values.

You will have to consider the configuration complexity of HP-UX client systems and what that means to CMS software. An important question is how the end user will control config settings. For example, HP-UX systems often have complex configurations that include many disks. Ignite-UX is able to install by choosing a default disk from the set of available disks. However, if the system has hundreds or thousands of disks, you might be extremely unhappy with this approach. It could be vital for you to be able to control things like root disk selection.

Depending on the end user policies, it might be possible to set up default config rules that manage these issues to a certain extent. For example, if the end user policy is to place OS content on local storage and use FibreChannel for application storage, Ignite-UX inventory blocking of FibreChannel could be used to ensure that FC LUNs are not selected for OS installation.

You should consider how to could create custom config content for clients that would fit with the CMS software approach for deployment control. This is important so you can get correct client-specific Ignite-UX configuration details not supported in your CMS user interface.

The CMS solution should consider software maintenance needs as supported by Ignite-UX. An Ignite-LUX server is unable to support HP-UX depot installation since the Software Deport (SD) agent (`swagent`) can only run on HP-UX servers. However, installations often rely on depots of patches and additional HP-UX software. Ignite-UX does have the ability for an installation to use an archive from a Linux server and software from depots provided from HP-UX servers.

Note: Support for the installation of Software Distributor (SD) depots via NFS from a Linux Ignite-LUX server might be supported in a future release. That approach does not require the `swagent` agent to run on the Linux server.

The focus of this section is installation and recovery using config files and archives that have previously been set up on the Ignite-UX server. CMS integrators should consider how that set up will be done. Manual set up of the Ignite-UX server might be acceptable. However, there might be considerable value in providing a means for controlling golden image creation and client recovery image creation.

The Ignite-LUX `ignite` CLI uses best practices for client management and provides machine-readable output in XML format. The `ignite, ignite-version, ignite-client,` and `ignite-config` manpages should be consulted. Using the Ignite-LUX CLI will simplify compatibility with future Ignite-LUX releases since the data format is extensible.

The Ignite-LUX `ignite` command has limitations and differences from the HP-UX Ignite-UX `ignite` command. The Ignite-LUX `ignite` command does not manage `bootp` configuration. Normally, CMS software manages client boot from a variety of operating systems and Ignite-LUX boot configuration would cause conflicts. Also, the Ignite-LUX `ignite` command does not include config file content parsing, so there is no check for syntax errors. Due to the approach used to implement Ignite-LUX, parsing is not likely to be developed anytime soon.

Care should be taken to avoid sensitivity to the format and details of file content accessed outside recommended commands. For example, it might make sense for a CMS to provide the ability to show `install.log` details, but it would be a bad idea to parse these log file details rather than use `ignite client status` XML data.

Listing Configuration Clauses

A CMS will need to list the OS configurations available for installation. For Ignite-UX, these are represented in config file content. Ignite-UX can only install configuration clauses for the same HP-UX release as the install kernel booted on the client. We assume these config clauses have been previously set up by an administrator.

The following Ignite-LUX command will list all available client configurations. The XML output shows available config clauses plus information about each clause, including the HP-UX release.

```
# ignite config list –m xml
```

Alternatively, the Ignite-LUX package includes the Ignite-UX `manage_index` command, which can be used to list the set of INDEX and client-specific CINDEX configuration clauses available for HP-UX installation. The `manage_index –r` option may be used to constrain the list to those appropriate for the install kernel that will be booted for installation or recovery. For example, to list the configuration clauses in the INDEX file for HP-UX 11i v3, use the following command.

```
# manage_index –lr B.11.31
```

Client-specific configuration clauses are stored in a CINDEX file in the client directory. Listing these entries is not currently supported by the Ignite-LUX `ignite` command. The `manage_index` command may be used to list configuration clauses in a CINDEX file. For example:

```
# manage_index –lr B.11.31 –i /var/opt/ignite/clients/hpuxsys1/CINDEX
```

Appropriate clauses in either the INDEX or CINDEX file may be used for installation.

Listing Clients Known to an Ignite-UX Server

The Ignite-UX server maintains knowledge of client systems that have been previously installed, are currently being installed, or are waiting for server controlled installation.

The CMS software will need to keep track of what client systems are being managed by the Ignite-UX server for HP-UX deployment. Use the following command to list clients. The `summary` and `detail` levels can be used to determine the current status of clients.

```
# ignite client list –m xml –l summary
```

Client status information might also be available in the client files within client-specific directories at `/var/opt/ignite/clients`. A directory for each uniquely identified client system is created on the Ignite-UX server via NFS. Symbolic links are created in the `/var/opt/ignite/clients` directory using other client system identifiers. For example:

```
# ls -l /var/opt/ignite/clients/ | grep 000E7FED52C5

drwxr-xr-x 3 bin bi 8192 Mar 19 02:13 0x000E7FED52C5

lrwxrwxrwx 1 bin bin 14 Mar 19 01:50 hpuxsys1 -> 0x000E7FED52C5
```

Network Boot for Automated Installation or Recovery

You might want to use the CMS software to automate booting. This is sometimes called “automated redeployment” and “cold dead iron provisioning.” You can also elect to have the initial system boot for install occur outside CMS control (for example, via manual operation).

The Ignite-LUX package does not include the Ignite-UX `bootsys` command. That Ignite-UX server command requires the client system to have HP-UX already installed and running. Often it is not reasonable to assume the client system is already running HP-UX. Also, many CMS solutions provide a separate method for client system power and boot control, so a consistent method is used regardless of the status of the client system. The *Ignite-UX Administration Guide* provides considerable detail regarding alternatives for client boot and Ignite-UX startup.

Regardless of how the system is to be booted, it is necessary to boot the Ignite-UX install environment to accomplish installation or recovery. This might include management of DHCP PXE `bootp` from the Linux server – this is intentionally not handled by Ignite-LUX.

Per-release `AUTO` files in the Ignite-LUX package are set up to automatically initiate server controlled installation. They also have entries for automated installation for the specific HP-UX release that do not wait for the server, and entries for manual installation for all supported HP-UX releases. These manual installation menu entries help you avoid issues such as when an administrator needs to install a specific HP-UX release but does not have administrative privileges on the Ignite-UX server to modify the `AUTO` file. In this situation, the administator can interrupt the boot process and select the correct HP-UX release for boot.

Booting the correct install kernel requires the use of the correct DHCP PXE boot file path. In this example an HP UEFI `dbprofile` is used to show the set of expected DHCP PXE response values:

```
Shell> dbprofile
Profile Name: iuxserv31
     Network Type: IPv4
     Client IP address: 15.1.48.199
     Gateway IP address: 15.1.48.1
     Subnet Mask: 255.255.248.0
     Server IP address: 15.1.53.172
     Boot File: /opt/ignite/boot/Rel_B.11.31/nbp.efi
     Optional Data:
```

An HP UEFI directed boot using a `dbprofile` may be used for testing to see if the Ignite-UX server is correctly set up.

Boot paths for a specific release are set up for server controlled installation. Upon boot, the client system will start the install environment and will wait for the server to instruct the client to proceed. You can use this to discover clients by having the CMS software use Ignite-UX client status and inventory data.

General CMS `config` Content

It might be useful for CMS software to provide an Ignite-UX `config` file that includes CMS-specific content appropriate for all client systems. The `config` file could be included in all `INDEX` file `cfg` clauses. For example:

```
cfg "HP-UX B.11.31 Custom Config" {
        description "User-Supplied 11.31 Config"
        "/opt/ignite/data/Rel_B.11.31/config"
        "/opt/ignite/data/Rel_B.11.31/hw_patches_cfg"
        "/var/opt/ignite/data/CMS_cfg"
        "/var/opt/ignite/data/config.local"
}
```

In this case, the CMS-specific config file `/var/opt/ignite/data/CMS_cfg` will be included in the config file list.

Hewlett-Packard does not recommend using CMS software to automatically “fix” user-created `INDEX` file configuration clauses to include the CMS-specific config file. Instead, it might be more appropriate to limit the list of available configuration clauses presented in CMS user interfaces to those that include the required config file.

Client-Specific config Content

Typically, CMS software will provide a method for controlling deployment content and configuration for a specific client system. These CMS user interface or policy choices must be converted to Ignite-UX config file content to control installation or recovery.

The `config` file in the client-specific directory is used to control client-specific installation or recovery details. Ignite-UX defaults will be used for client-specific `config` content that is not specified. The following `config` content shows some of the most commonly used settings:

```
# cat /var/opt/ignite/clients/0x001083049436/config
cfg "HP-UX B.11.31 Default"=TRUE
_ip_cfg_detail_levels="ivsp"
init _hp_root_disk="WWID='0x5000c50005e33057'"
```

The `cfg` Ignite-UX keyword specifies the `INDEX` file configuration clause to be installed. Valid values can be obtained by listing configuration clauses using the `ignite` or `manage_index commands`.

The `_ip_cfg_detail_levels` variable must be set to indicate what config content is specified in the client-specific config file. Ignite-UX software will use this value to help decide what configuration information comes from standard config files and what comes from client-specific content. The Ignite-UX `instl_adm(4)` man page should be consulted for further details.

The `_hp_root_disk` variable indicates the disk to use as the HP-UX installation target device. If not specified, Ignite-UX will choose a disk using a default selection method. Typically, users prefer to explicitly indicate which disk to use. The Ignite-UX `instl_adm(4)` man page details the parameters that may be used for selection. The `io.info` inventory data file lists the client's disks.

Initiating Client Installation or Recovery

Once all Ignite-UX server, CMS-general, and client-specific config files are set up, the Ignite-UX server is ready to support client installation.

If the non-interactive installation approach is used, installation may be initiated by network-booting the client system to use the Ignite-UX server.

The following command may be used to initiate server-controlled installation.

```
# ignite client control –x action=start_install 0x001083049436
```

In this example, the client system is identified using its MAC Address.

Warning: It is possible to interrupt a non-interactive or server controlled installation to make client system configuration modifications via the Ignite-UX user interface. CMS software should be designed to appropriately handle this interruption if possible.

Client Installation Progress

It is common for CMS software to report progress on operations that take a long time, such as HP-UX installation and recovery. The following command may be used to monitor install progress.

```
# ignite client list -m xml -l status <mac address>
```

The pctComplete and state values indicate progress. Step names, which indicate install phases, are intended to be fairly human-readable. CMS software should not parse or be coded to recognize these state values, but might display the last one as part of status.

While not recommended, you could directly access client files.

The client_status file indicates progress via listing the installation phases completed and the phase currently in progress. The phase names are meaningful and can be reported along with percentage progress.

Client Installation Status

The following command may be used to monitor installation status by using the state value to indicate high-level status. The client-specific log file install.log should offer more details.

```
   # ignite client list -m xml -l status <mac address>
```

Unfortunately, things sometimes go wrong. The install.log will report Ignite-UX NOTE, CAUTION, WARNING, and ERROR messages. ERROR messages indicate installation failure.

CMS software could instruct the user to look in the log file, or could provide a way to browse or search the file, but should not parse the log file content.

Certain Ignite-UX WARNING message can be expected (see Client-Specific File: install.log for details). Other WARNING messages indicate a complete installation with issues that should be communicated. Other message types are intended to be informative and do not indicate serious installation issues.

The install.log might include error message output from HP-UX programs run by Ignite-UX to perform installation. It might be useful to perform a case-insensitive search of the install.log file.

Client Installation Timeout

You might want your CMS software to monitor installation progress for installations taking longer than some period of time (timeout). These installations would be considered failures and the client system should be reset. This requires deciding on an appropriate timeout.

The time required to complete an installation or recovery depends on many factors. For example, the content to be installed, the performance of the client system, and the client system memory and I/O configuration are some of the important characteristics that might influence installation or recovery time.

Successful installations can take as long as 20 hours in some very extreme situations with unsupported configurations. Typical network installations take 20 to 60 minutes, assuming relatively fast networks, but can easily require 2 hours.

It might be a best practice for CMS software to allow configurable timeouts.

Installing CMS Agent Software

If your CMS system management solution requires installation of client agent software to control HP-UX systems after installation is completed, you will want to consider how the agent software should be installed.

If a golden image is used for install content, the agent software could be installed on the original system and thus included in the golden archive. It might be necessary to have an iux_postload control script to customize agent settings for the specific client system at the end of the installation process.

Ignite-UX command or script hooks may be used to perform special operations during the installation process. Files required by the program or script may be included in archive content installed on the system.

Files may also be located on the Ignite-UX server and accessed via NFS so that they do not need to be included in installed content. For example:

```
post_config_script += "/usr/contrib/bin/gunzip \
  /var/opt/ignite/data/CMS_agent.tar.gz | tar xvf -"
```

Note that Ignite-UX includes several of these hooks at different phases in the installation process. In the earlier stages, only very limited HP-UX commands are available. When possible, this type of customization should be done in the late stages, as in the example above. See `instl_adm`(4) for more information.

It is also possible to have the `INDEX` configuration clause include content from a depot on an HP-UX server. In this case the configuration clause may include a golden image and a depot. The depot must come from an HP-UX Software Distributor server.

Installing Additional HP-UX OS Software and Patches

It might be desirable to install additional HP-UX software as part of the initial system installation or after installation has completed. HP-UX OS software is packaged in Software Distributor depot format. During initial network installation, support of installation of this software will require an Ignite-UX server that is distinct from the Ignite-LUX server.

Another alternative is to use CMS agent control or HP-UX scripts during boot-after-install to install additional software. HP-UX Software Depot commands may be used to convert depots to a serial format consisting of a single file. A serial depot may be located on the Ignite-LUX server and accessed via NFS.

Warning: Do not use Ignite-UX config file program or command hooks to attempt `swinstall`. This will result in overlapping installations and unpredictable results. In some cases installation might work or might appear to work, but you cannot be confident in the results. Ignite-UX installation should be allowed to fully complete before additional software is installed.

Additional Information

Websites

- *http://www.hp.com/go/ignite-ux*
- *http://www.hp.com/go/ignite-ux-docs*
- *http://www.twitter.com/HP_UX_Docs*

Product Documentation

- *Ignite-UX Quick Start Guide*
- *Ignite-UX Administration Guide*
- *Ignite-UX Reference (Ignite-UX manpages, including instl_adm(4))*

Contacting HP

HP welcomes your feedback. To make comments and suggestions about product documentation, send a message to docsfeedback@hp.com. Include the document title and manufacturing part number. All submissions become the property of HP.

The information contained herein is subject to change without notice. The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

MPN 5992-5164, August 2010