The Creative Application of Ignite-UX to Meet Unique Requirements

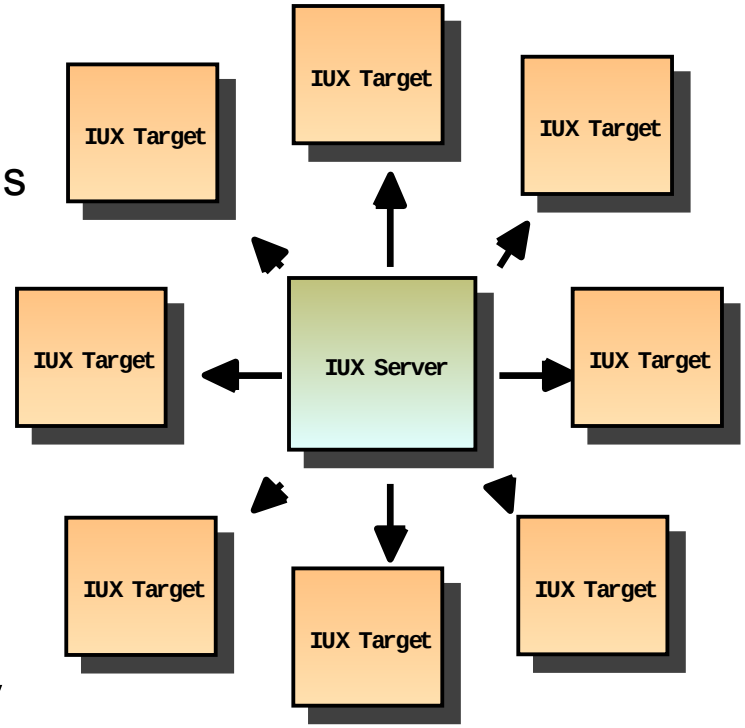
Tim McHenry
Senior Solutions Architect
Hewlett Packard Co.
4680 Parkway Dr.
Cincinnati, OH 45040
tim_mchenry@hp.com
513-754-4315
Interworks 2002 #009

Agenda:

- Review Ignite-UX Basics
- Creating System Images (Depots vs. Archives)
- Internals
- Configuration File Logic and Fine Points
- Pre & Post Commands & Scripts
- Best Practices
- Case Studies
- Procedures to Create Bootable Ignite CDs
- Where to go for more information

What Is Ignite-UX?

- process for initial system deployment or redeployment
- client/server model:
  - can install multiple target machines simultaneously
  - allows target customization and status monitoring
- ability to build and reuse standard configurations
- ability to do site-specific and system specific customization
- ability to automate installation process
- extensive system manifest capability
- ability to install software from multiple sources in a single session

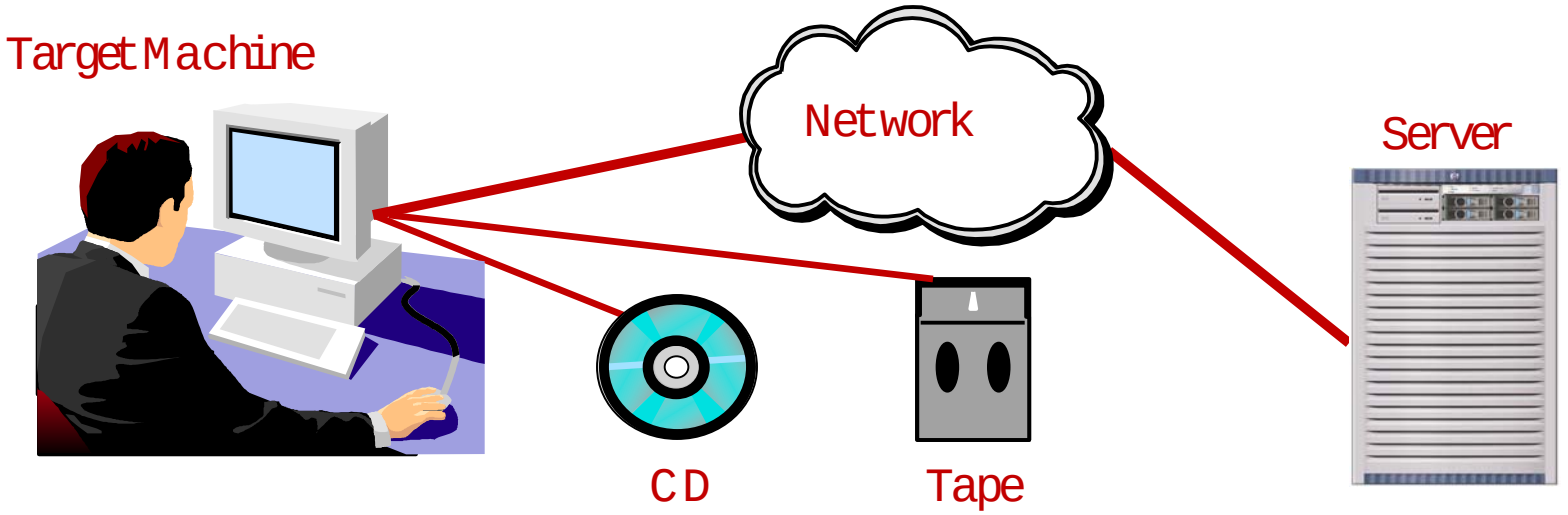


Ignite-UX versus SD-UX

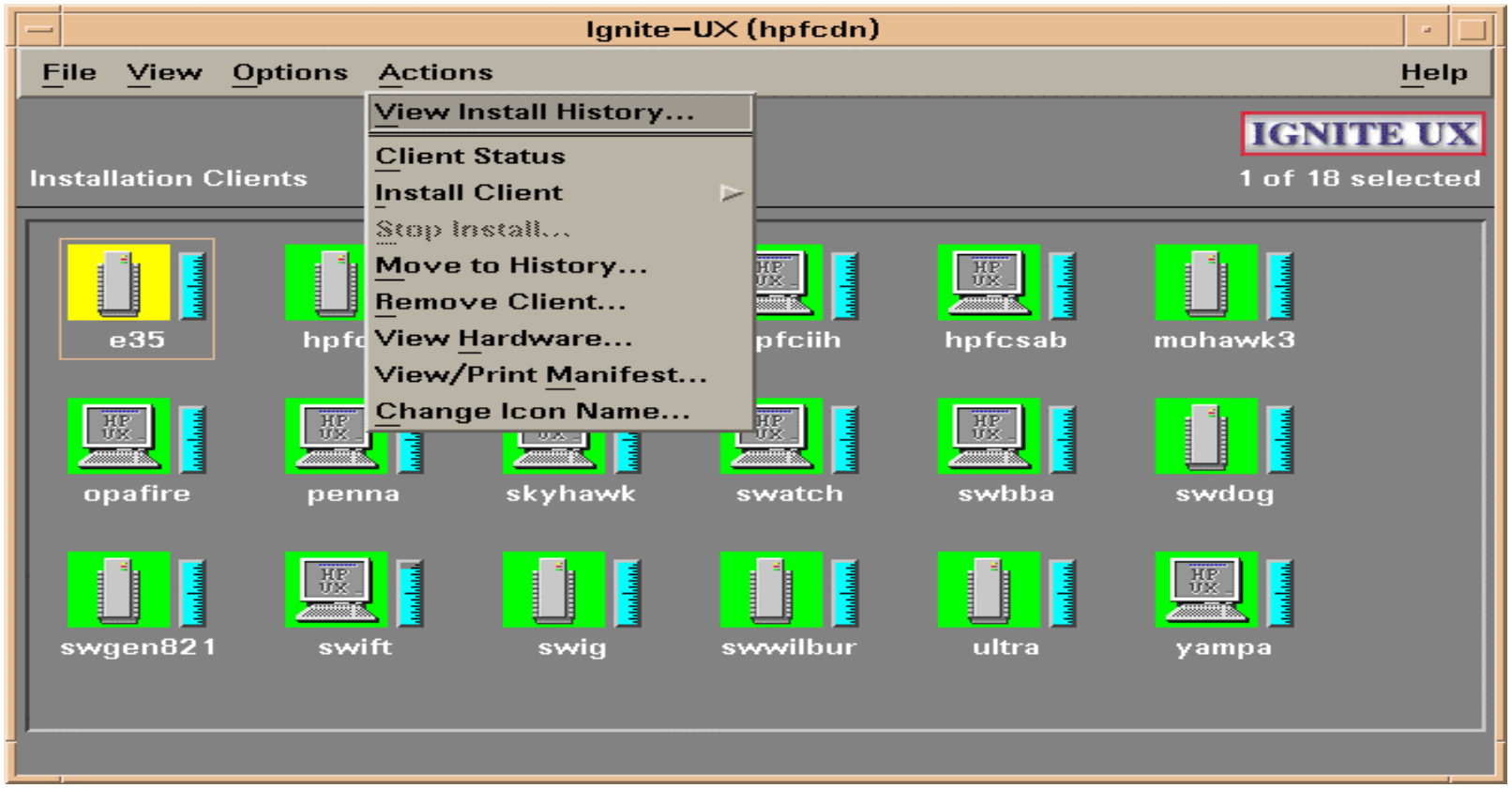
	Ignite-UX	Software Distributor
Purpose	Complete installs of system software.	Manages software on an existing system.
Disk space considerations	Performs disk and file system layout based on software selected.	Cannot modify file system if there is insufficient space.
Objects handled	Handles SD depots and/or golden system archives in **tar** or **cpio** format.	Understands standard bundles, products and file sets (OS, Applications, patches etc.)

Fundamental Models for Ignite-UX

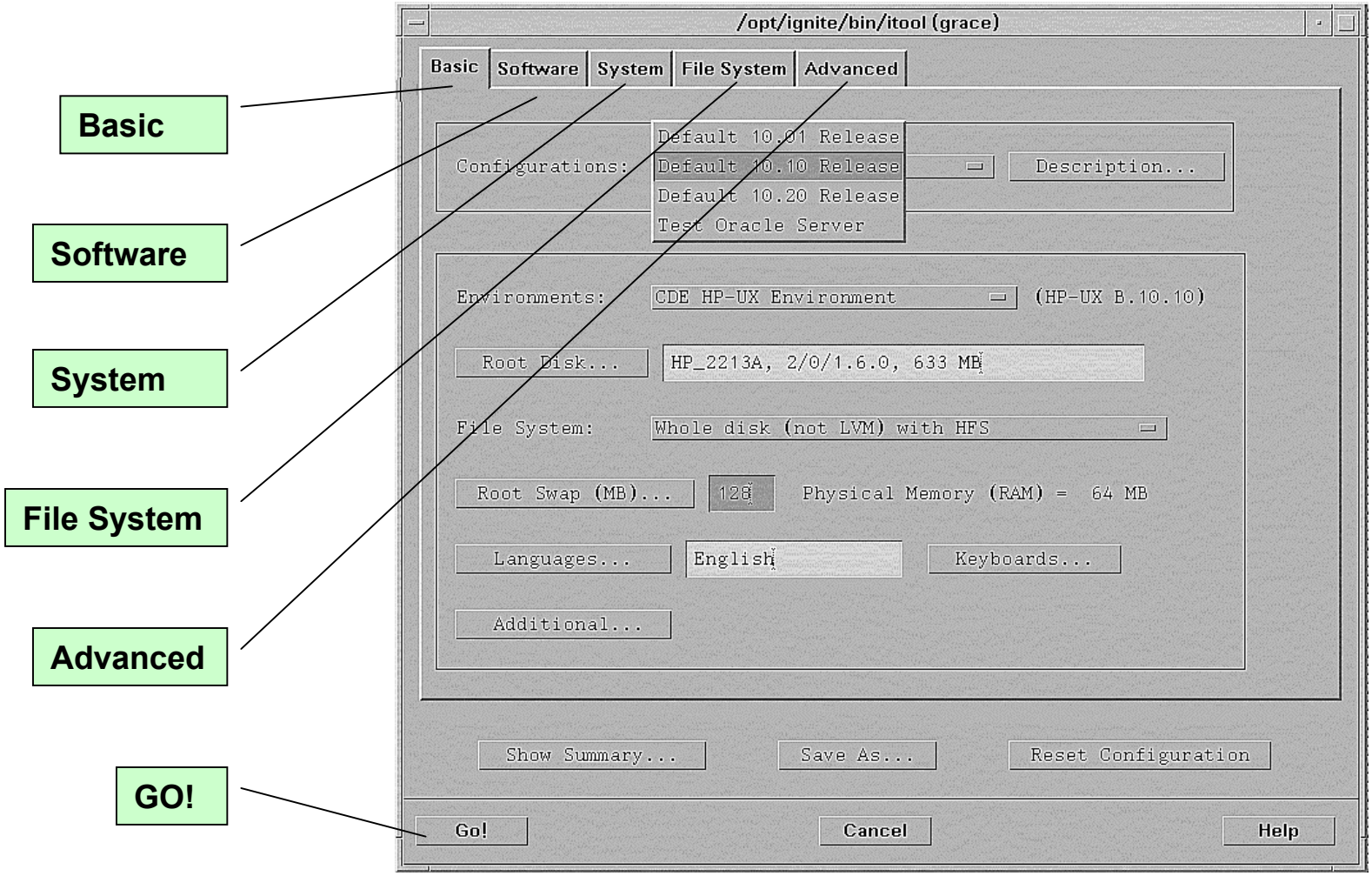
- Initial Installation from Media (Tape or CD)
  - New, redeployed, repaired systems
  - Examples: Instant Ignition and Easy Setup CDs
- Installation over the Network from Target (Push)
- Installation over the Network from Server (Pull)
- Redeploy over the Network from Target or Server



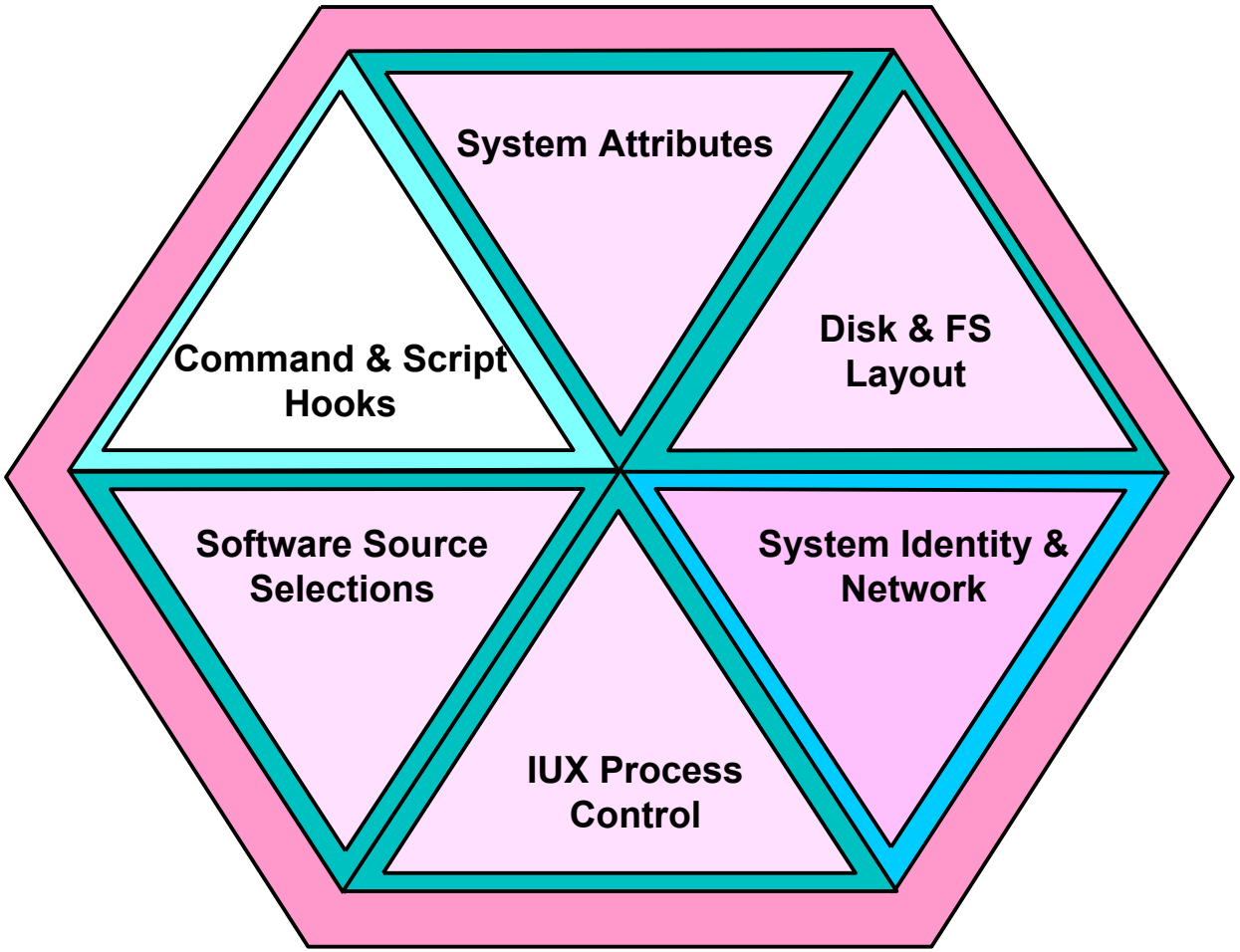
Graphical User Interface (or not)



Ignite-UX Client Configuration



Ignite-UX Configuration Files



Creating System Images

Depots vs. Archives

	Depots	Archives
Original Setup	Straightforward	Time Consuming
Additions	Simple to add patches or applications	Must add to master and harder to undo
Modifications	Easy to make changes	Requires changing the master
Image Size	Larger	Smaller
Load Time	Slower	Faster
Best Use	Ongoing image management	Quick cloning and compact media image

Creating a Depot Based Image

Step 1: Create a large directory space(/IGNITE/depots)

Step 2: Making the depots from HP CD media

Core **Applications** **Support Plus**

```
# make_depots  -v  -d  /IGNITE/depots  -s  /dev/dsk/c0t0d0
```

Step 3: Use swremove to remove any unwanted software from the depot:

```
# swremove @  /IGNITE/depots
```

Step 4: Create the configuration file for your depot:

```
make_config  -s /IGNITE/depots  -c  /IGNITE/config_file
```

Creating an Archive (*Golden*) Image

Step 1: Load the master system (*ignite_master*) software:

- Operating System
- Patches
- HP & 3rd Party Applications

Step 2: Configure the system for proper operation:

- Networking, name services, etc.
- Local and remote file systems
- Kernel tuning, etc

Step 3: Exhaustively test system for proper operation!

- **Don't shortcut this step!!!**

Step 4: Create a directory (/IGNITE) with enough disk space to hold at least 50% of the current disk space used by the master system (bdf).

Creating an Archive (*Golden*) Image(cont)

Step 5: Determine which directories you wish to exclude (/IGNITE) from the archive (use +NO_ARCHIVE heading) and place them in a file (/IGNITE/files_to_exclude).

Step 6: Determine which system configuration files you wish to prevent from being reset:

- Those in /etc/rc.config.d/ for example
- See manpage for make_sys_image for details

and place these in the file created in step 5 (use +NO_RESET heading).

Step 7: Create the final archive image:

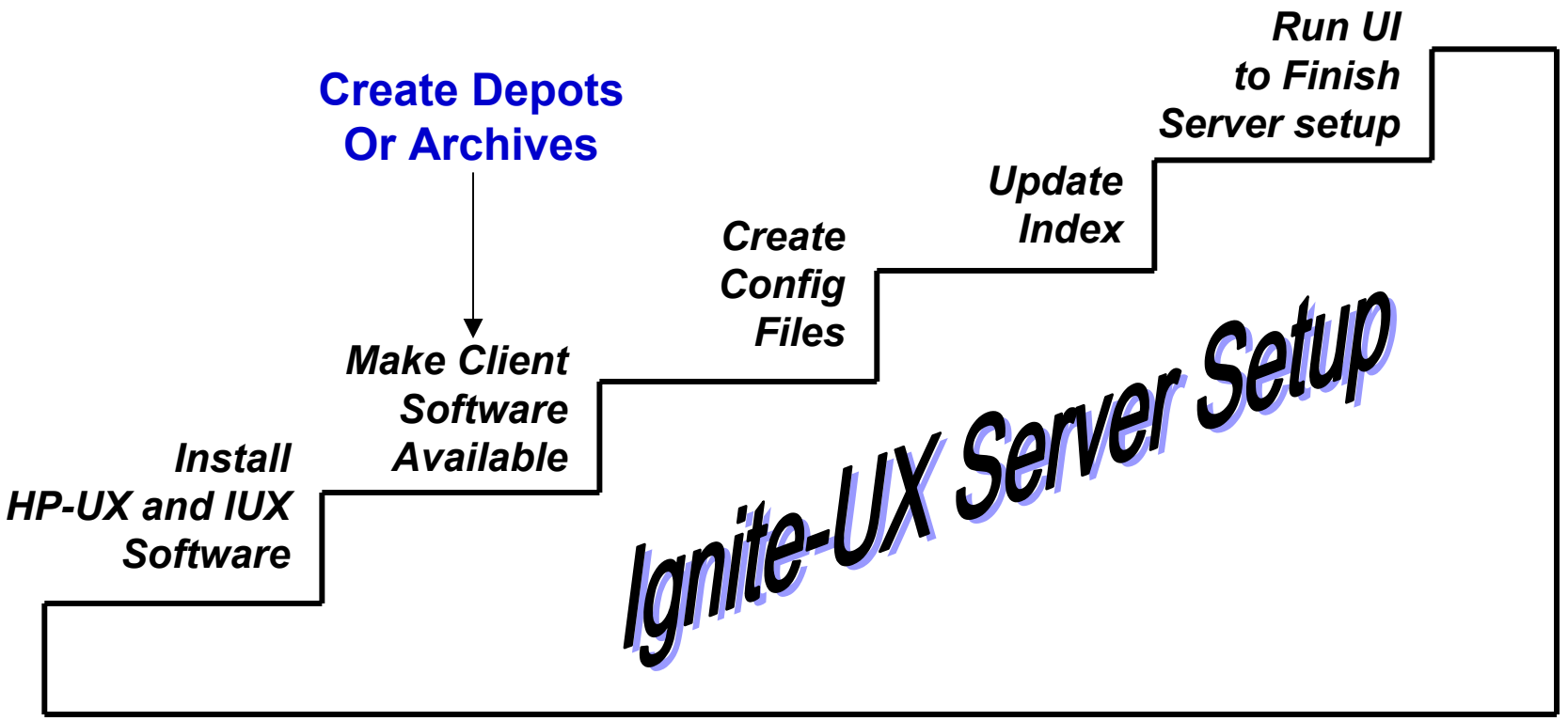
 # make_sys_image –v –d /IGNITE –s local –n image.gz –f /IGNITE/files_to_exclude

Step 8: Create configuration files beginning with:

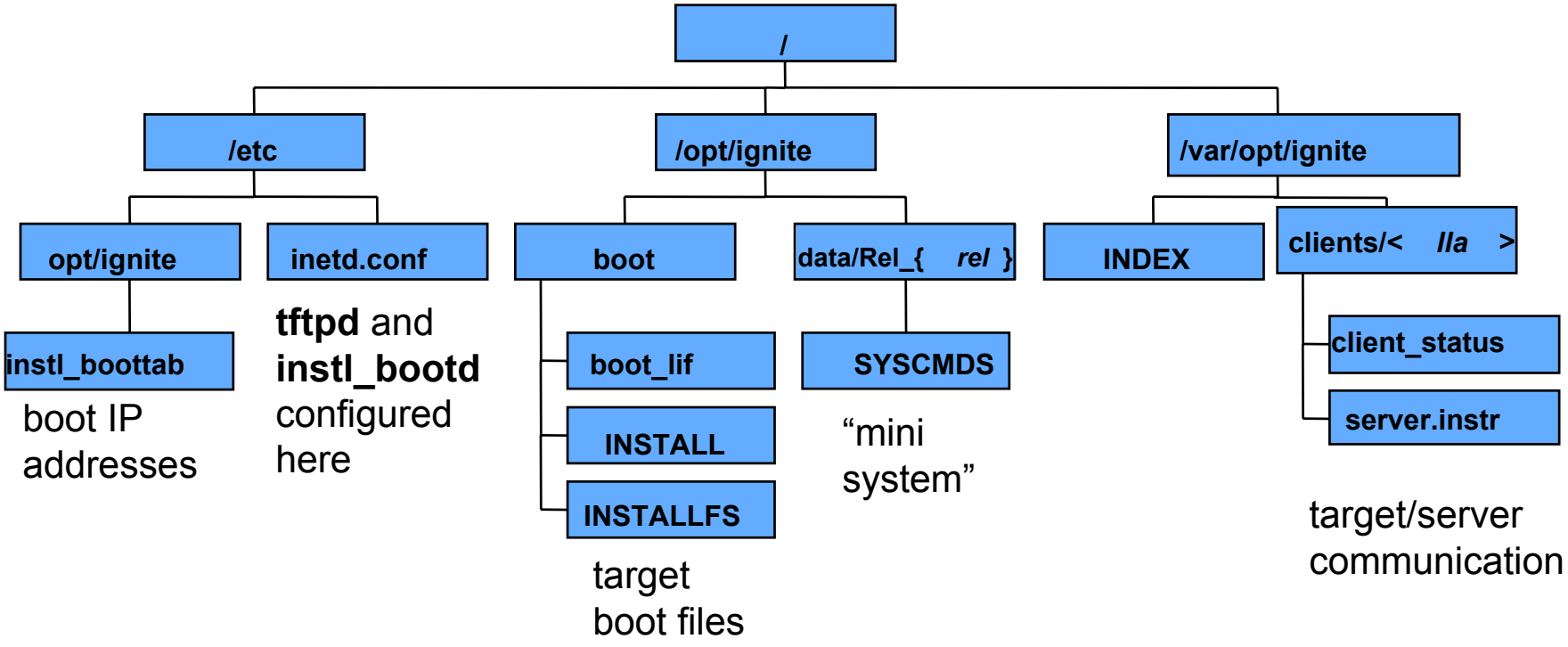
 # /opt/ignite/lbin/archive_impact -t -g image.gz >archive_impact.cfg

Ignite-UX Internals

Ignite-UX Server Setup Overview



Ignite-UX File Set



Key Ignite-UX Files

1. /opt/ignite/boot/ :

   INSTALL – Memory-based 32-bit Ignite Kernel
   INSTALLFS – Memory-based Ignite File System (gzipped tar file)
   WINSTALL – Memory-based 64-bit Ignite Kernel
   VINSTALL – V-Class Memory-based 64-bit Ignite Kernel
   boot_lif - bootable LIF file to initiate Ignite boot (ISL, AUTO, HPUX)
   fs_cfd.def – sample 8K LIF area at the beginning of INSTALLFS

2. /opt/ignite/data/Rel_{ref #} :

   SYSCMDS – additional system commands available by internal calls(gzipped tar file)

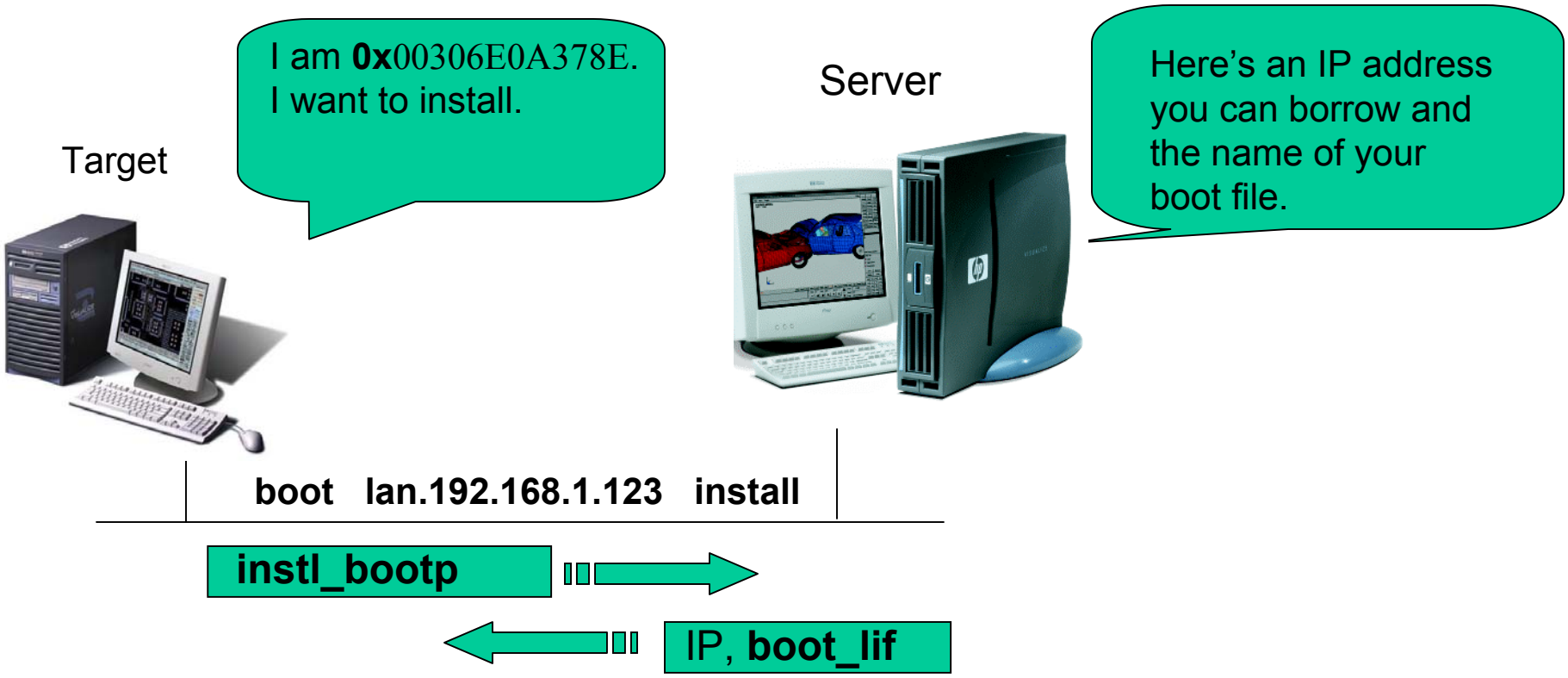
3. /var/opt/ignite :

   INDEX – Default location of the primary Ignite index which links all configuration file to a specific system configuration
   clients/local – all local results and log file for a target node
   clients/{-lla} – found on Ignite server where there is Ignite installation directory (Link Level Access) for each ignited target

Network Source Installation

(Running HP-UX OS (9.05/9.07, 10.x, 11.x)

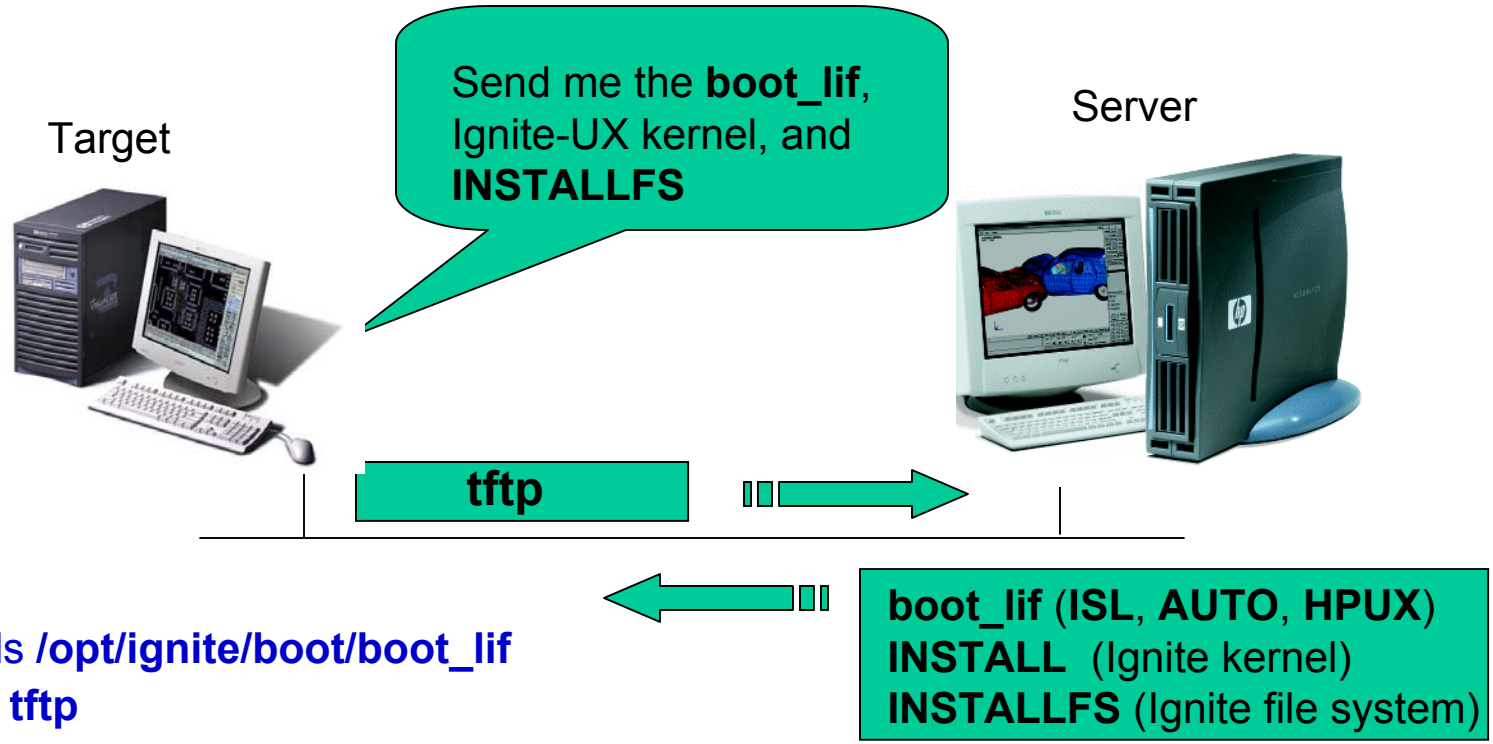
Booting the Target — Pull Install



- Sends an **instl_bootp** request to server
- Pulls **boot_lif** via **tftp**

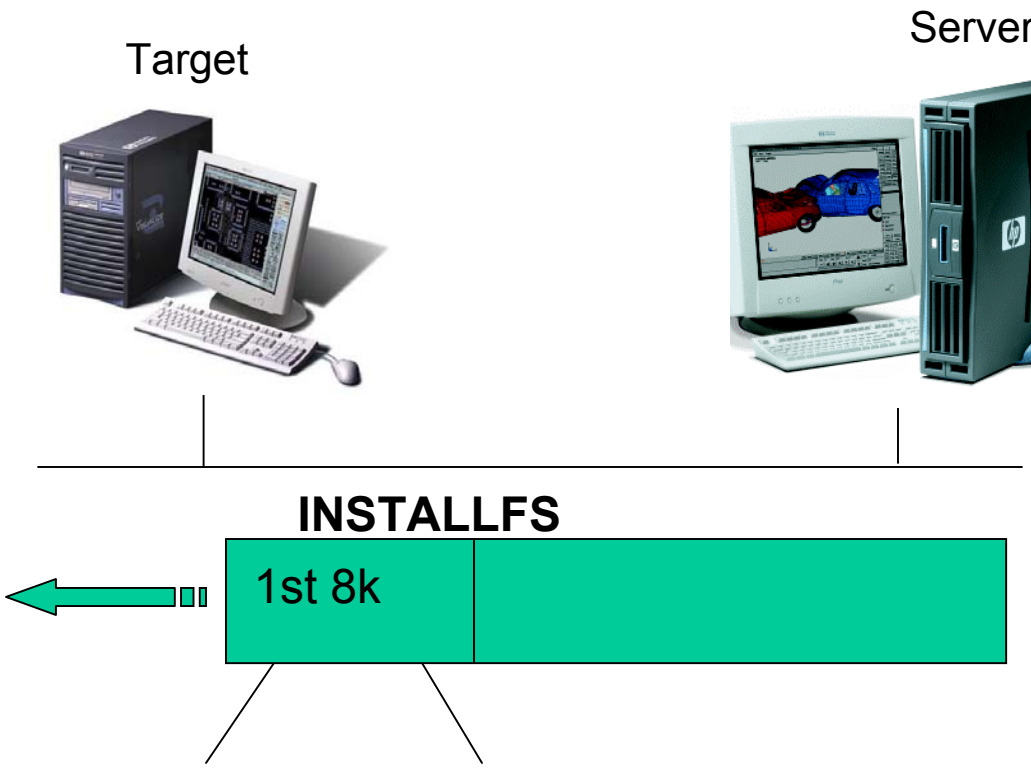
- **inetd** starts **instl_bootd**
- **instl_bootd** gets a temporary IP address from **/etc/opt/ignite/instl_boottab**
- sends **udp** packet back with IP address and path to **/opt/ignite/boot/boot_lif**

Booting the Target — Pull Install (Continued)



- Pulls **/opt/ignite/boot/boot_lif** via **tftp**
- Runs **ISL**, gets **AUTO** file from **boot_lif**
- **AUTO** file pulls **HPUX** from **boot_lif**
- **HPUX** pulls **IUX** kernel, **/opt/ignite/boot//INSTALL** via **tftp**
- System starts executing the kernel
- Kernel looks for and pulls **INSTALLFS** via **tftp**

INSTALLFS



Customize with:

- GUI/TUI
- **setup_server**
- **instl_adm(1m)**

- Where is IUX server?
- Use DHCP for IP for remaining install?
- Gateway, subnet,…
- Should UI run? On server or target?

Install Boot Process (Continued)

Target

IUX Server 192.168.1.123

- Pulls **/opt/ignite/data/INSTCMDS** (IUX commands) via **tftp**.
- NFS mounts server **/var/opt/ignite/clients/** directory.
- Creates **clients/<lla>/** and **client_status** file on server.
- Waits for **server.instr** file to be created in **clients/<lla>** directory with the command **start_install.** →
- ← Gets client config files
- Based on the OS release to be installed, pulls **/opt/ignite/data/Rel_{rel}/SYSCMDS** (mini-system) via **tftp.**
- Client tracks **server.instr** file. Server could change request to stop install and reboot!

Server sees new **/var/opt/ignite/clients/<lla>** directory, and flags Ignite UI of new client.

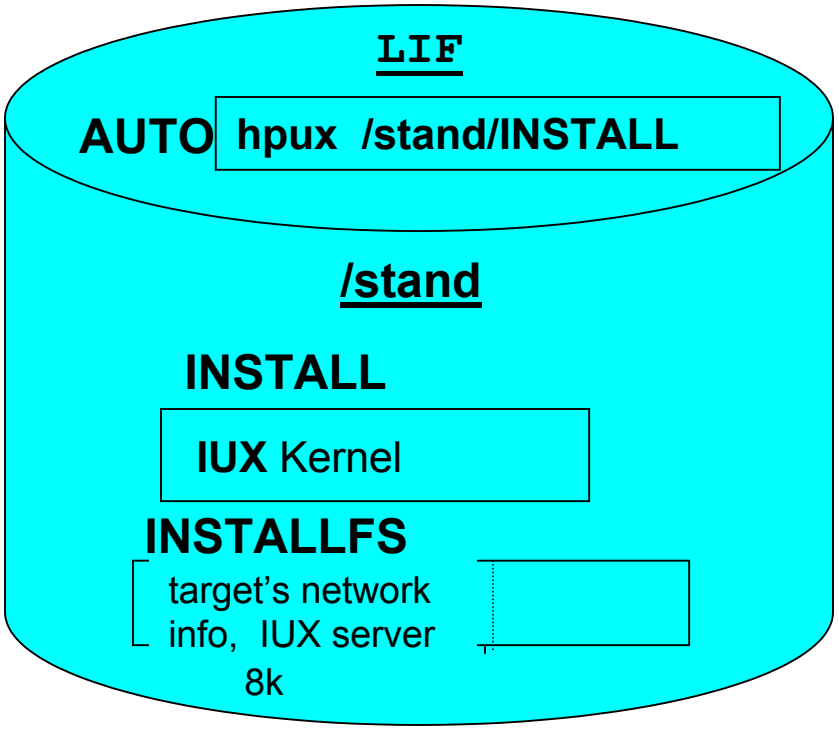
Once config files are generated, creates a **server.instr** file containing the command **start_install**

Push Install — `bootsys`

bootsys is an IUX server command, which reboots a target system and begins an IUX install.

Server:

- Copy **INSTALL** and **INSTALLFS** to target.
- Customize **INSTALLFS** on target with target's current network information.
- Modify target's **AUTO** file in **LIF** area.
- Reboot target (from its local disk).



Using **bootsys**

- Interactive install:

  # bootsys -vw <*target*>

  – machine will boot and wait for instructions

  – run ui to continue install

- Noninteractive install:

  # bootsys -a <*target*>

  # bootsys -a -i <*configuration*> -f

  – Install non-interactively using previous, default, or specified configuration

- Requires **.rhosts** on target machine

Cautions about **bootsys**

- **bootsys** modifies the **AUTO** file and the primary boot device on the target system. Despite being a powerful tool it has some sensitive areas:(Repair **AUTO** with **mkboot** command if necessary)
  - HP_UX 10.x & 11.X have **setboot** command to change the primary boot device, but 9.0x systems do not have such a command. Fixing 9.0x system's primary boot device requires interrupting the boot process and fixing it at the ISL prompt.

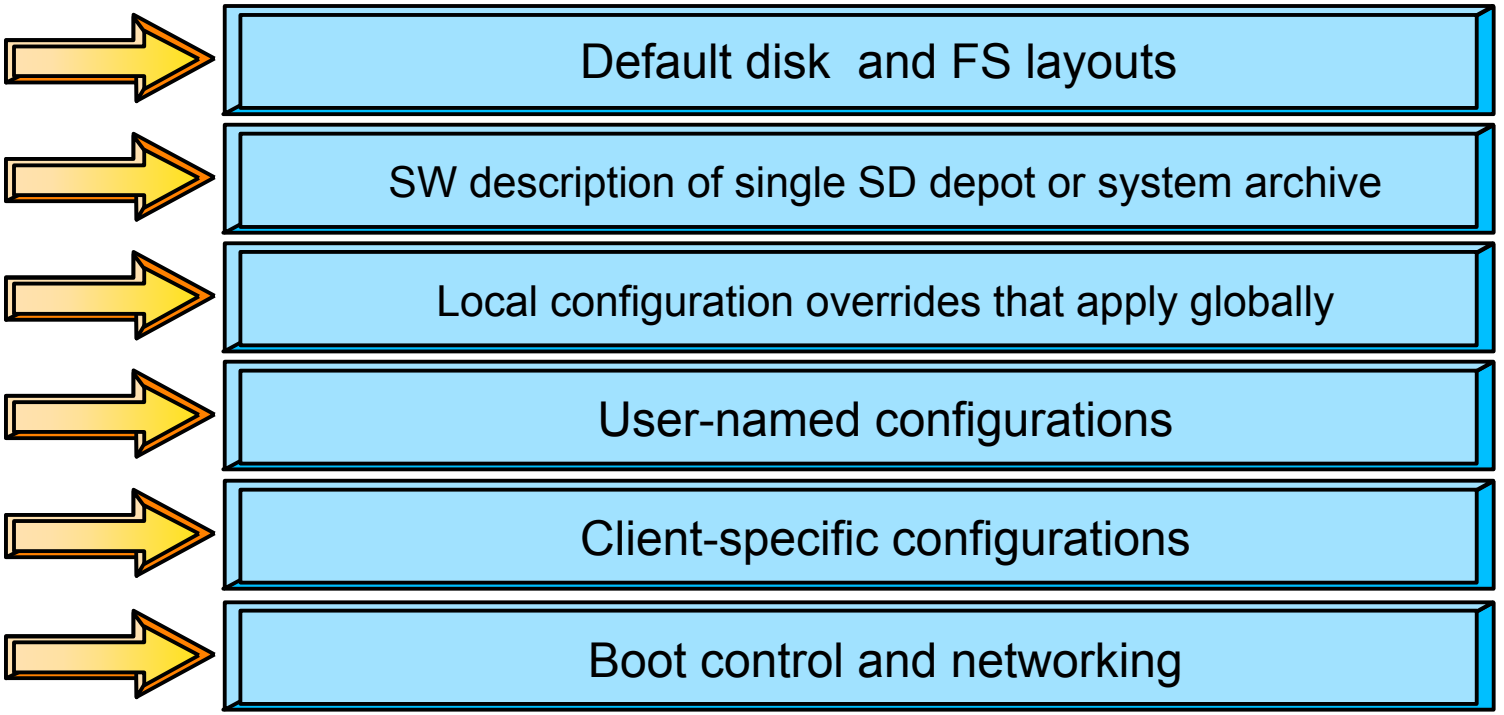
Configuration File Logic and Fine Points

Where Do Configuration Files Come?

- Shipped with Ignite-UX
  - defaults: **/opt/ignite/data/Rel_{*release*}/config**
  - examples: **/opt/ignite/data/examples**
- IUX commands
  - **make_config**: describes software in a depot
  - **save_config**: creates config file to match current hardware configuration
- IUX GUI/TUI
  - save as...
- Created when a client is installed
  - **/var/opt/ignite/clients/0x{*lla*}/config**
- Manually from scratch using your favorite editor

The Ignite-UX **INDEX** File

Defines a grouping or set of configuration files



INDEX File Default Location is /var/opt/ignite

INDEX File Example

```
cfg "HP-UX B.11.00 Default" {
        description “HP default system configuration for the B.11.00 release.“
        "/opt/ignite/data/Rel_B.11.00/config"
        "/opt/ignite/data/Rel_B.11.00/hw_patches_cfg"
        "/var/opt/ignite/config.local“
}
cfg "HP-UX B.11.11 Default" {
        description “HP default system configuration for the B.11.11 release."
        "/opt/ignite/data/Rel_B.11.11/config"
        "/opt/ignite/data/Rel_B.11.11/hw_patches_cfg"
        "/var/opt/ignite/config.local"
}
cfg “Interex 2000" {
        description “Example for these slides."
        “/IGNITE/CONFIG/dsk.cfg“
        “/IGNITE/CONFIG/kernel.cfg”
        "/opt/ignite/data/Rel_B.11.11/hw_patches_cfg"
        "/var/opt/ignite/config.local"
}
```

Disk Configuration File Snippet (dsk.cfg)

```
disk[_hp_root_disk].size >= 8000Mb & disk[_hp_root_disk].size<= 10000Mb {
   volume_group "vg00"
   {
    max_physical_extents = 2500
      logical_volume "lvol3"{
         mount_point = "/"
         size = 200Mb
         contiguous_allocation = true
         bad_block_relocate    = false
      }
      logical_volume "lvol2" {
         usage = SWAP_DUMP
         mount_point = "primary"
         contiguous_allocation = true
         bad_block_relocate    = false
         size = memory * 2
      }
           ...
      }
```

Other useful config entries:

size = 500Mb | remaining | 20% free

```
volume_group “apps”{
    physical volume disk[10/0/15/0.4.0]{}
    physical volume disk[10/0/15/0.5.0]{}
      logical_volume “apps1” {
            mount_point=open_source
            usage=VxFS
            size=36000Mb
}
```

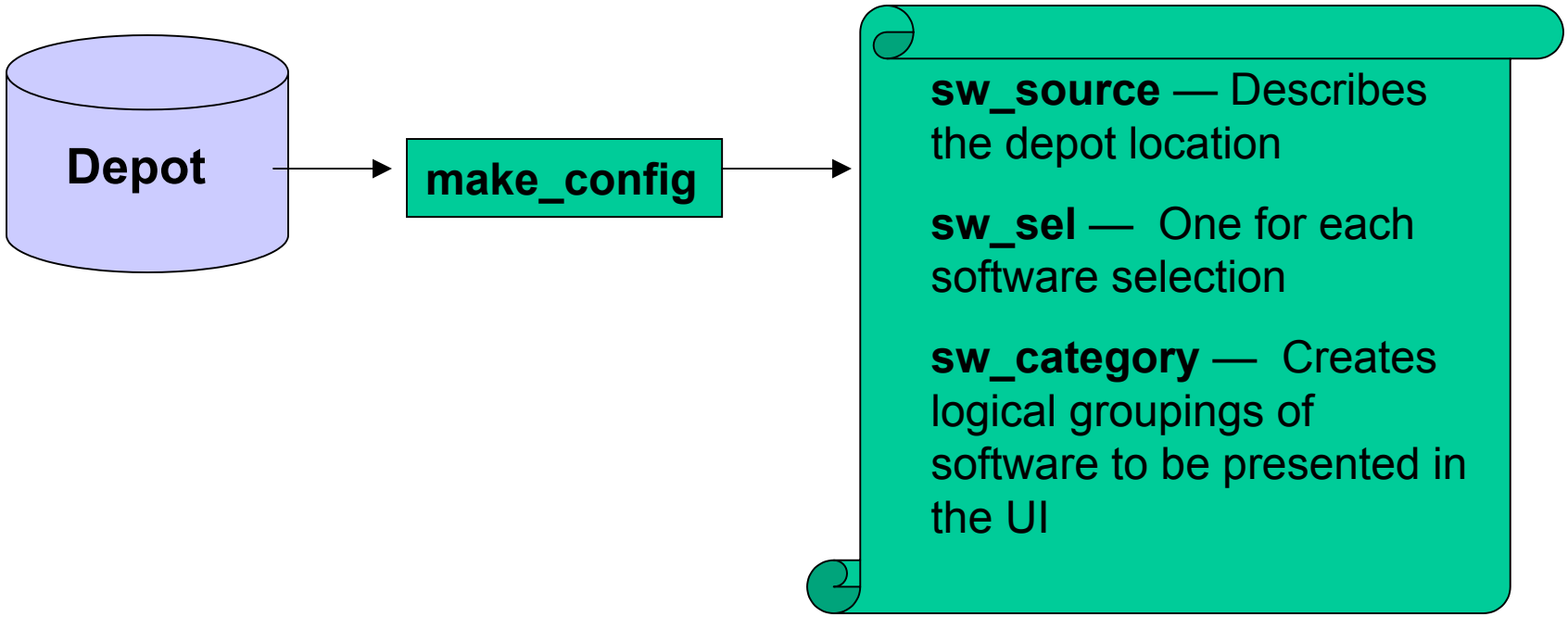
Kernel Configuration File Snippet (kernel.cfg)

```
(memory >= 384MB & memory < 512MB)
        {
                mod_kernel +="maxusers              128"
                mod_kernel +="nproc                1784"
                mod_kernel +="maxuprc              1764"
                mod_kernel +="shmmni                810"
                mod_kernel +="semmnu                426"
                mod_kernel +="nstrtel               264"
                mod_kernel +="maxswapchunks        1480"
                mod_kernel +="ninode               6144"
                mod_kernel +="nfile                7776"
                mod_kernel +="dbc_min_pct             5"
                mod_kernel +="dbc_max_pct            15"
        }
(memory >= 512MB & memory < 768MB)
        {
                mod_kernel +="maxusers              128"
                mod_kernel +="nproc                2264"
                mod_kernel +="maxuprc              2244"
                mod_kernel +="shmmni                ....
```

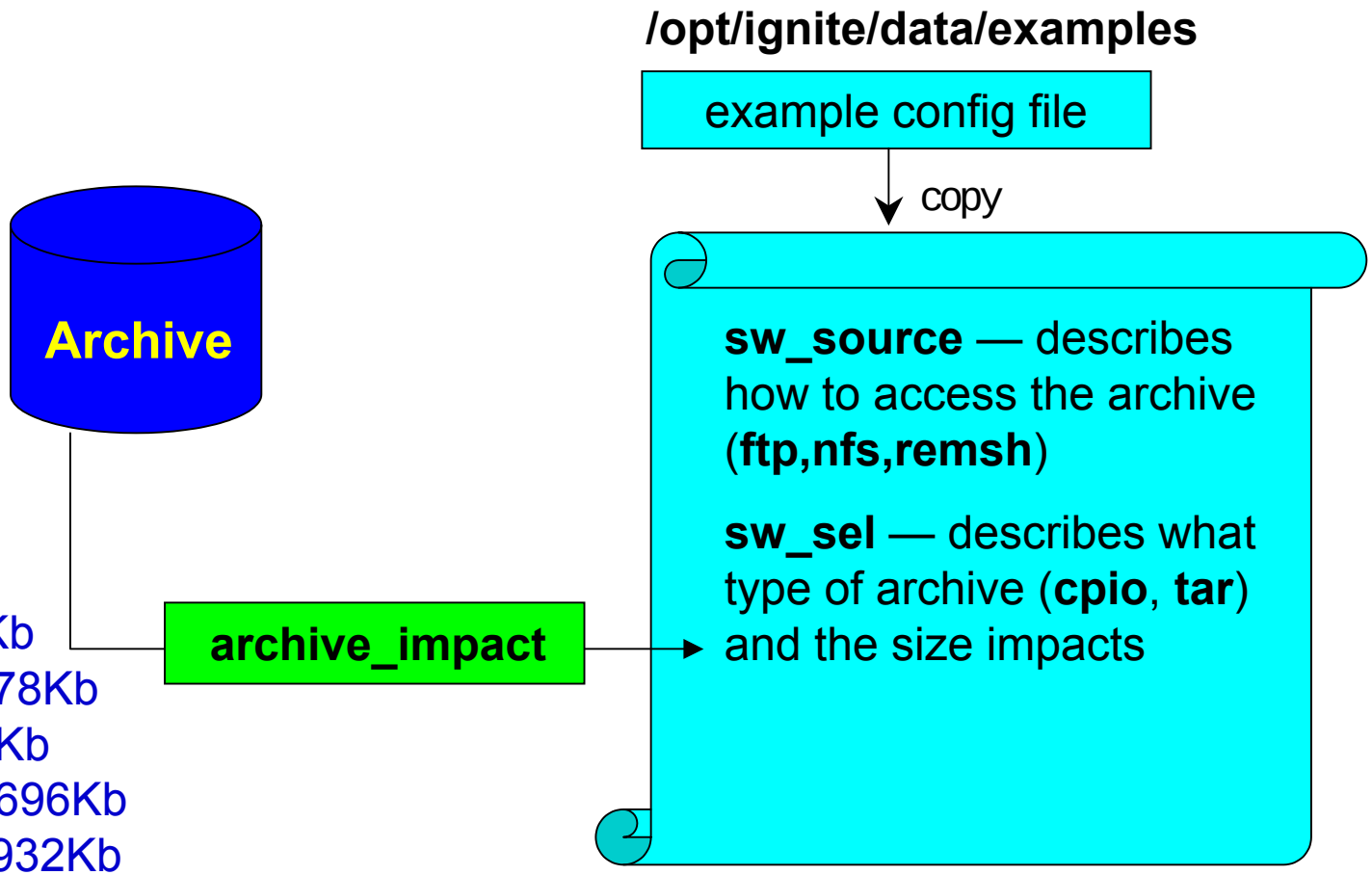
Configuration File for a Software Depot

make_config -s *source_depot* -c *config_file*
-a 700|800|both

/var/opt/ignite/data/Rel_{*release*}/*name*

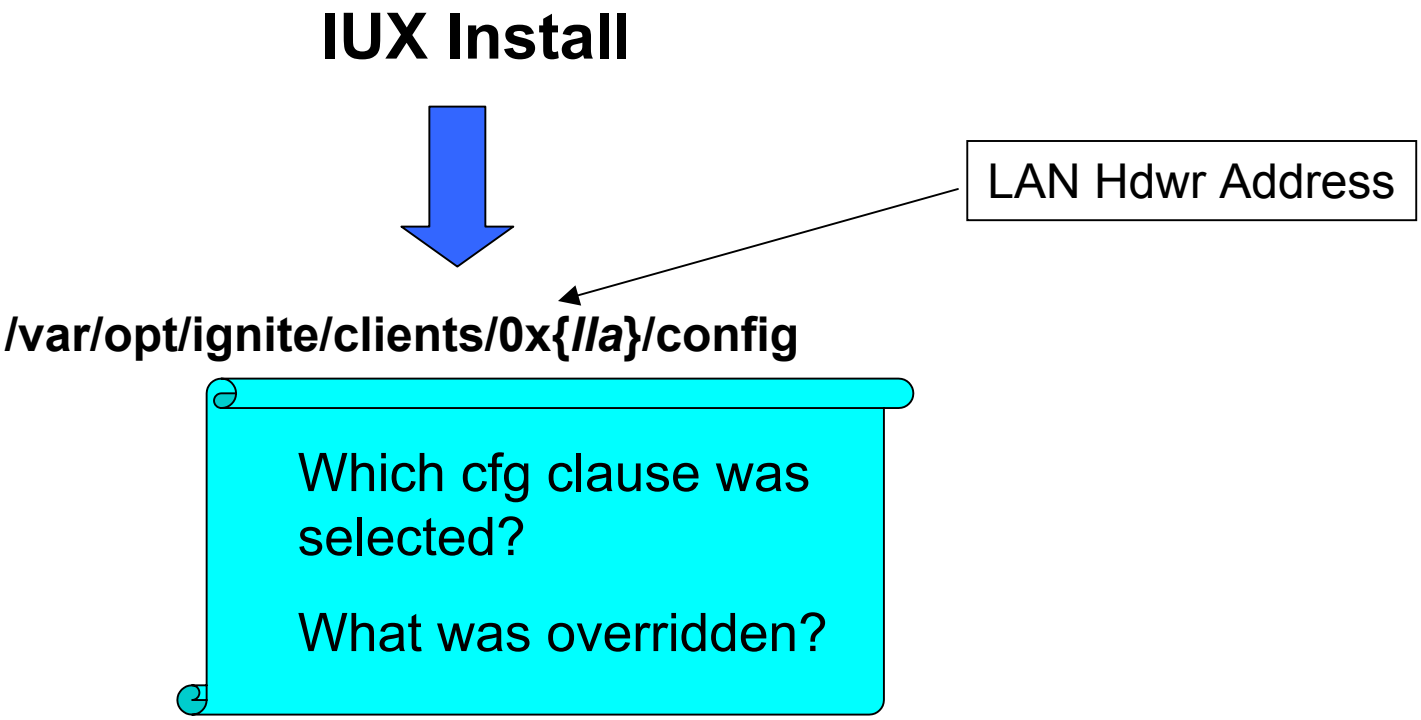


Configuration File for an Archive



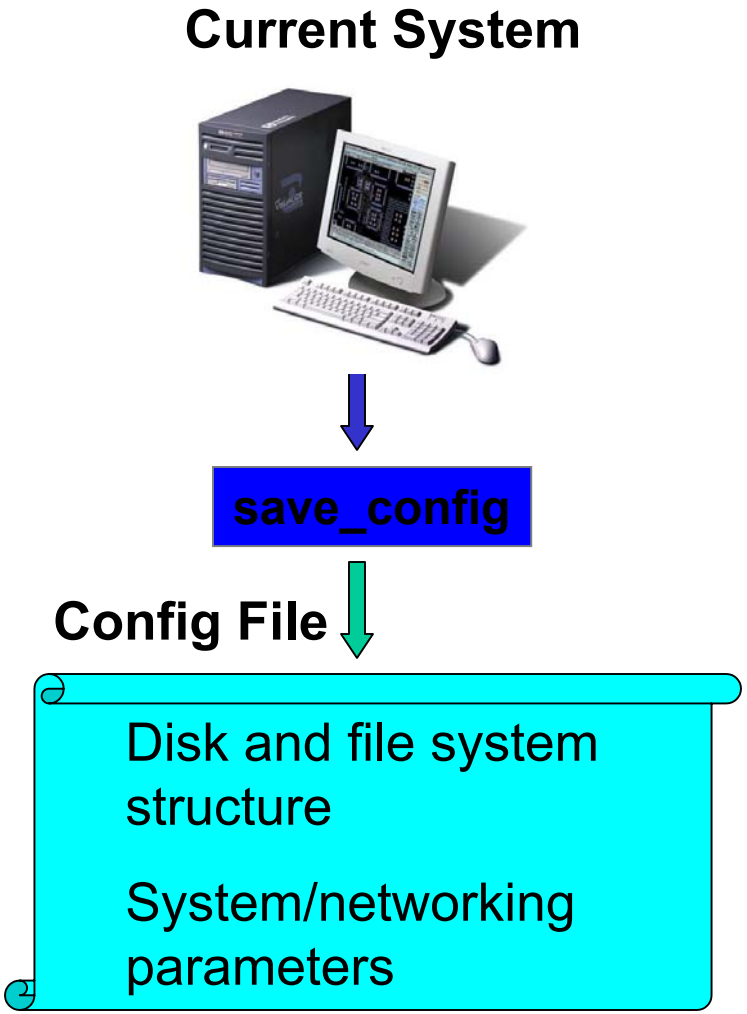
Example:
impacts = "/" 14Kb
impacts = "/dev" 13Kb
impacts = "/etc" 21578Kb
impacts = "/home" 1Kb
impacts = "/opt" 355696Kb
impacts = "/sbin" 34932Kb
impacts = "/stand" 1048Kb
impacts = "/usr" 655560Kb
impacts = "/var" 219530Kb

Client-Specific Configuration File

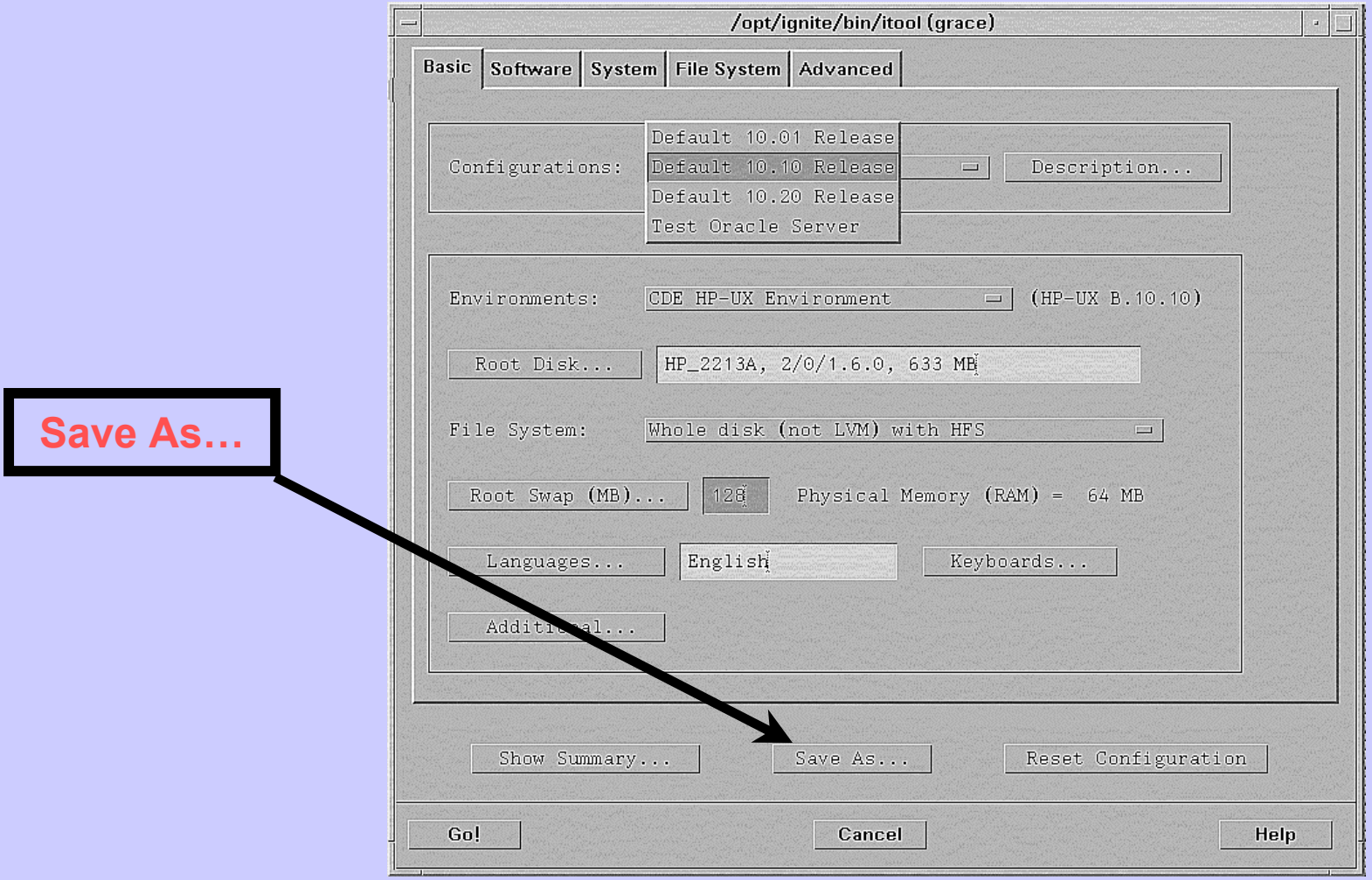


- **Used for performing a reinstall**

Using the “save_config” Command

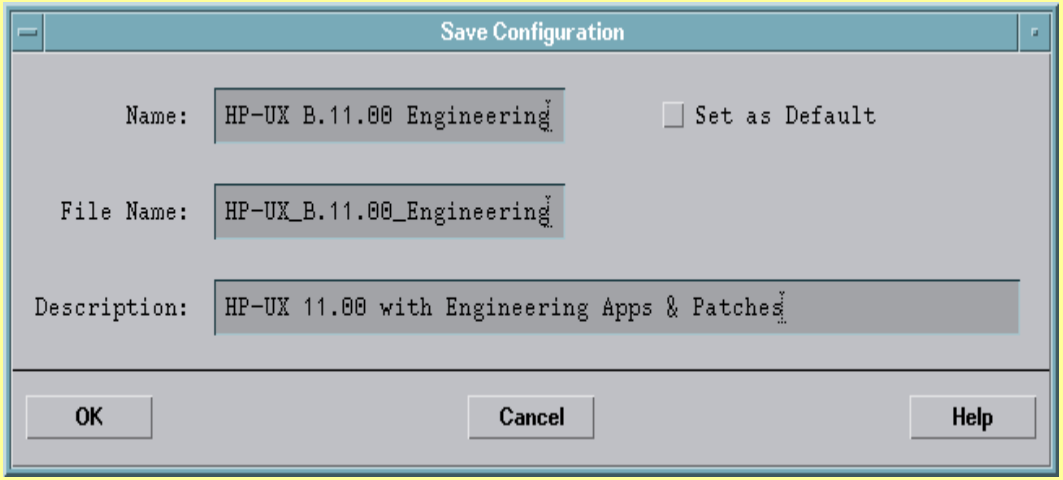


Saving an Ignite-UX Configuration



Creating a Custom Configuration with the User Interface

1. Select a configuration in User Interface:
   - such as, HP-UX B.11.00 Default
2. Make modifications.
   - How you want to lay out disks?
   - What software you want installed?
   - Other changes?
3. Choose **Save As** to write your customized configuration file.
4. Optionally, hand-edit the configuration to make it hardware independent.



Don’t Forget the INSTALLFS Configuration (Contained in the first 8K Block)

This File Contains Boot control parameters and networking Information such as:

- Whether or not to invoke the UI:
  run_ui=FALSE
- Whether control will be from Server or Client:
  control_from_server=TRUE
- Set key Ignite environment variables to prevent unnecessary user interaction:
  env_vars = "INST_BATCH_MODE_TIMEOUT=0"
  env_vars += "INST_ALLOW_WARNINGS=1"
- Add informative console messages:
  sysadm_message=“OS Installation Underway“

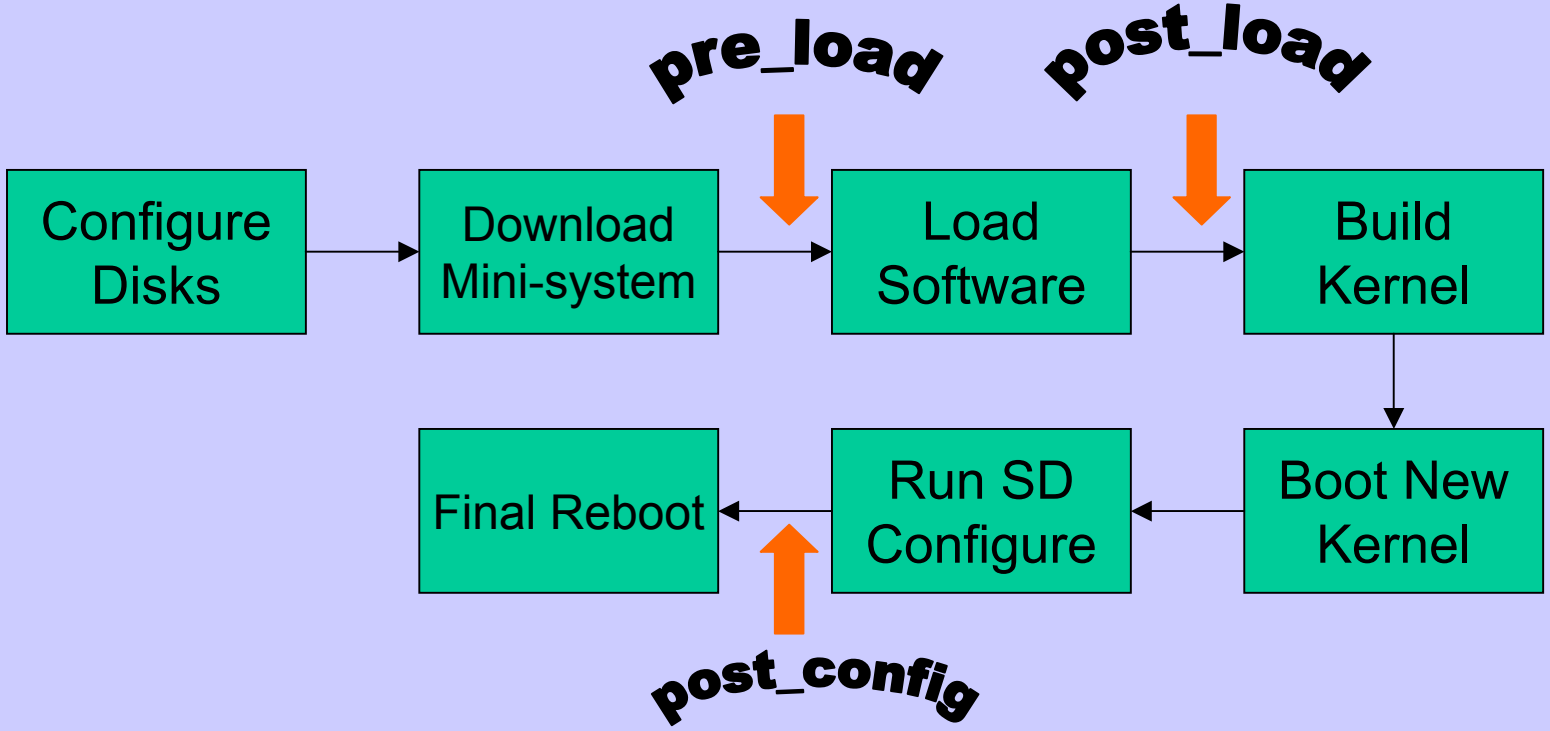
To view: # instl_adm –F /opt/ignite/boot/INSTALLFS
To modify: # instl_adm –F /opt/ignite/boot/INSTALLFS –f <*new_cfg*>

Tips For Creating Trouble Free Config. Files

1. **Use templates included within the Ignite installation file**
2. **Keep the "man 4 instl_adm" output handy**
3. **Read the Ignite-UX Admin. Manual and other documents available on the public Web page.**
4. **Use the GUI Interface to create custom configuration files**
5. **Solicit help from the Ignite Email list**
6. **Study the work of others (I.e. Easy Setup CDs, etc.)**
7. **Make extensive use of the "instl_adm –T –f <*your_config_file*>" to check syntax and instl_dbg to debug them.**

Pre & Post Commands & Scripts

Opportunities for Inserting Commands and Scripts



Note: Until "Load Software " step all Ignite-UX files are memory based

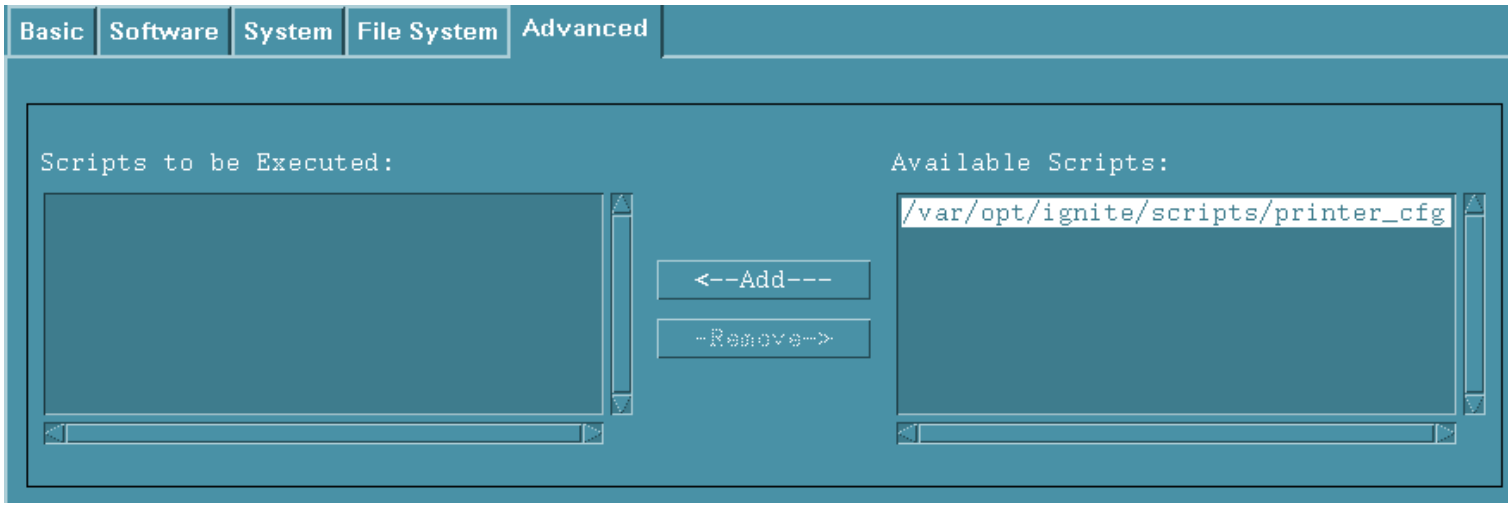
Customizations Provided

- Any number of tasks may be performed on the target using execution commands and scripts.
  - **NFS, DNS, NIS configuration etc.**
  - **Mounting additional disks**
  - **Configuring spoolers**
  - **Loading additional application software not stored in an SD depot**
  - **HW customizations not supplied in an archive**
  - **Copying user data to a target**
  - **Adding RC scripts**
  - **Testing hardware**

Valid Locations for Commands and Scripts

- Software related
  - **`sw_source`**
  - **`sw_sel`**
- Conditional execution
  - In any config file
- Sitewide customization
  - **`/var/opt/ignite/config.local`**
- Selectable in the UI
  - **`/var/opt/ignite/INDEX`**
- SCRIPTS section of LIF area on media
- Standard post config and post load scripts:
  - **/opt/ignite/data/scripts/os_arch_post_[c,l]**

Scripts Defined in the **INDEX** File



- A list of optional scripts may be defined in **/var/opt/ignite/INDEX**:

```
scripts {
    "script-file-path-name-1"
    "script-file-path-name-2"
      }
```

Tips on Creating System Admin. Scripts

1. Run SAM.
2. Perform your task (for example, add a printer).
3. Select **`Options->View Sam Log`**.
4. Change **`Message Level`** to **`Commands Only`**.
5. Save logfile and exit SAM.
6. Cut commands from the saved file and paste them into your script.

Using Commands Instead of Scripts

- Use Config File Reserved Words:
  - pre_config_cmd
  - post_config_cmd

Examples:

```
pre_config_cmd = “loadfile –q –l SYSCMDS /sbin/sh;
                  loadfile –q  -l MYSCRIPTS ./menu.sh;
                  /sbin/sh ./menu.sh;
                 /sbin/rm ./menu.sh /sbin/sh;”

post_load_cmd =“ tar xvf  /tarfiles/masterconfig.tar
                rm –rf  /tarfiles”
```

Case Studies

Case Study #1: Auto Dealer Management System Provider

Customer Profile: Provide complete automobile dealership management system to 100+ customers per month in US and Canada

Requirements

- Single HP-UX based image to configure systems to support 10 – 3000 users
- B-Class to L-Class(rp5400) systems
- 1-4 CPUs
- 256MB – 8GB Memory
- 9GB – 100GB Disk Capacity (JBOD, DS2100, Model 12H AutoRAID & VA74XX)
- CDROM for loading and DAT Drive for backup
- Remote support primarily through modems
- Easy to install by Dealer personnel (only basic computer skills required)
- Simple terminal console interface for all system admin.
- Frequent application updates (including new OS features or patches)
- Automated system software version control

Case Study #1: Auto Dealer Management System Provider (cont.)

Ignite-UX Solution

- Create single HP-UX 11.0 image to include all relevant drivers and patches
  - maintain an archive image for size and speed
- Develop complex pre-load script providing simple menu based installation
  - add required commands to INSTALLFS *(UNSUPPORTED!)*
- Create custom configuration files –
  - making extensive use of conditional statement
  - using separate config. file for:
    - + disk configuration
    - + kernel tuning based on memory ranges
  - define 8K INSTALLFS for local terminal console UI only
- Utilize LIF area for Version Control *(UNSUPPORTED!)*
  - *lifcp <version file>*
  - *key to keep it small using a static format*
- Use “ideal” Ignite-UX system to create and test new images quickly
- Use cdrecord utility to burn master CD image
- Use make_recovery to maintain a backup of all archive image versions

Case Study #1: Auto Dealer Management System Provider (cont.)

Adding pre-load script to LIF file:

1. Develop and test script (/IGNITE/preconfig.sh)
2. Place script in tar file:

```
# cd /IGNITE
# tar cvf MYSCRIPTS  ./preconfig.sh
```

3. Compress and rename tar file:

```
# gzip  MYSCRPTS
# mv MYSCRIPTS.gz  MYSCRIPTS
```

4. Add it to existing LIF file (lif-file)

```
# lifcp –r  /IGNITE/MYSCRIPTS  /IGNITE/lif-file:MYSCRIPTS
```

5. Loading and executing preconfig.sh from Ignite Config file:

```
pre_config_cmd = " loadfile –q –l SYSCMDS  /sbin/sh;
                   loadfile –q –l MYSCRIPTS  ./preconfig.sh;
                   /sbin/sh  ./preconfig.sh;
                   /sbin/rm  ./preconfig.sh  /sbin/sh;"
```

Case Study #2: Automated Disk Formatting System

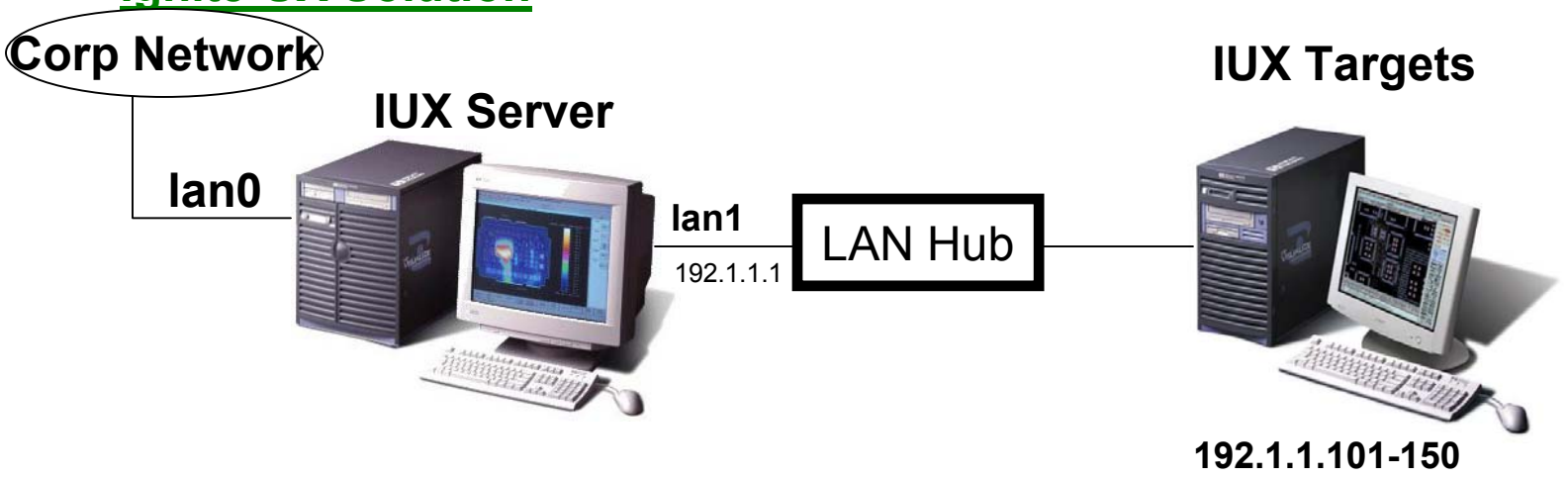
Customer Problem: Need simple system to completely format all hard disks on 600-800 workstations annually

Requirements

- Engineering design workstations data must be securely removed prior to return to leasing company as part of annual upgrade project
- LAN based solution
- Administered by personnel with limited Sys. Admin skills
- Records maintained in simple data base for auditing purposes
  - Indexed by company asset number
  - Record disk manufacturer & model number
  - Record Sys. Admin. Name
  - Record date and time
- Must be able to detect failed hard drives so they can be destroyed
- Email reports on weekly basis to appropriate personnel

Case Study #2: Automated Disk Formatting System(Cont.)

Ignite-UX Solution



- Make use of Ignite-UX memory based kernel and file system only!
- Make use of DHCP to automatically create temporary IP addresses for target systems to be formatted
- Develop pre-load script which:
  - provide simple operator menu to initiate format and capture name
  - utilize “mediainit” to determine good or bad disks and format good mechs.
  - send all pertinent data to IUX server for data base collection
- Develop scripts for auditing reports(for weekly email and on-demand)

Best Practices

Common Problems and Resolutions

1. **Networking services not configured correctly:**
   - Insure tftp and bootps are uncommented in /etc/inetd.conf
   - Verify NFS operations and be sure /var/opt/ignite/clients is included in the /etc/exports file
     - NFS server started in /etc/rc.config.d/nfsconf
   - Make sure hostname resolution is setup correctly:
     - /etc/nsswitch.conf (/etc/hosts, DNS, NIS)
     - Make sure server can find client by hostname
   - If using DHCP make sure temporary Ignite addresses do not conflict with DHCP address range
2. **Test all scripts thoroughly before adding them to Ignite config files**
3. **Config file syntax – use "instl_adm –T" to verify**

Other Useful Ignite-Utilities

make_recovery and make_net_recovery:
- provide easy means to create bootable tape backup for Ignite systems used for cloning or for paranoid users

print_manifest – a non-root utility for displaying a system's complete hardware and software configuration

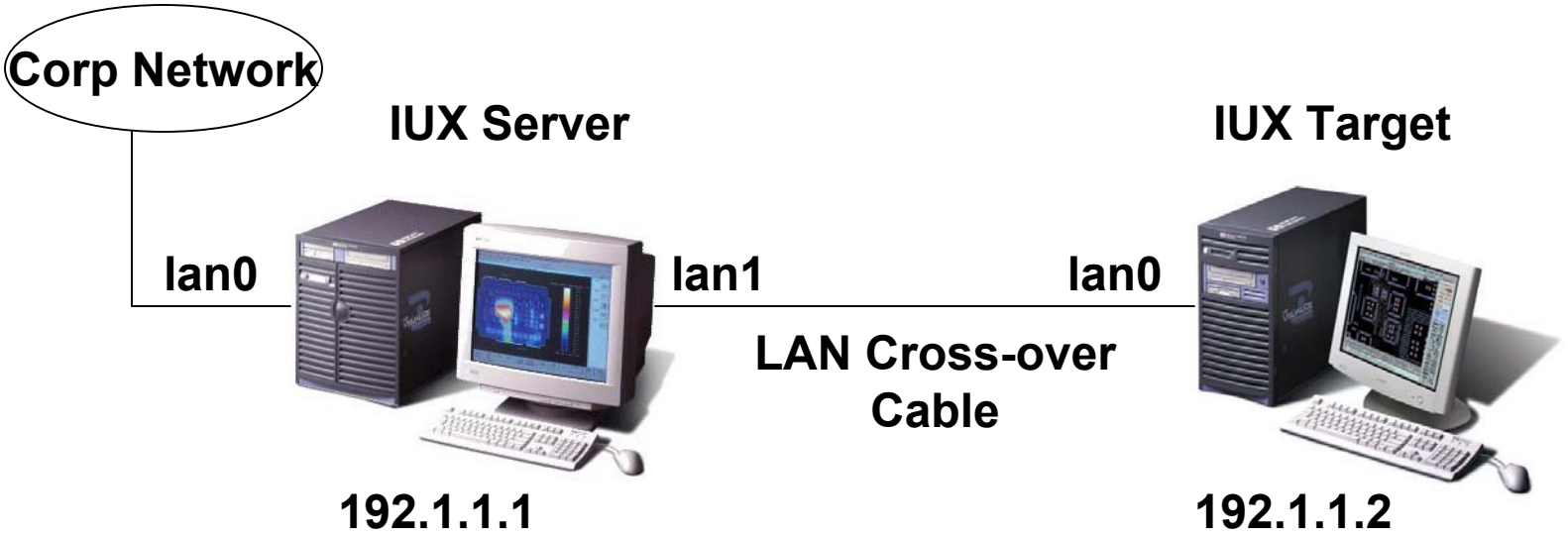
instl_adm and instl_dbg – great for manipulating and testing config files

make_medialif – creates complete LIF area for bootable Ignite media

instl_combine – combines the LIF area with the archive or depot image for bootable Ignite CD media.

lifls, lifcp, lifrm, lifinit, lifrename – utilities for managing LIF files

The Ideal Ignite-UX Self-training Lab



- 2 removable disks
- CD_RW Drive
- DDS Tape Drive
- Easy access to the Corp. Net.
- Easy access to Public Web
- access to printer

- 2 or more removable drives
- access to local printer

Procedures to Create Bootable Ignite CDs

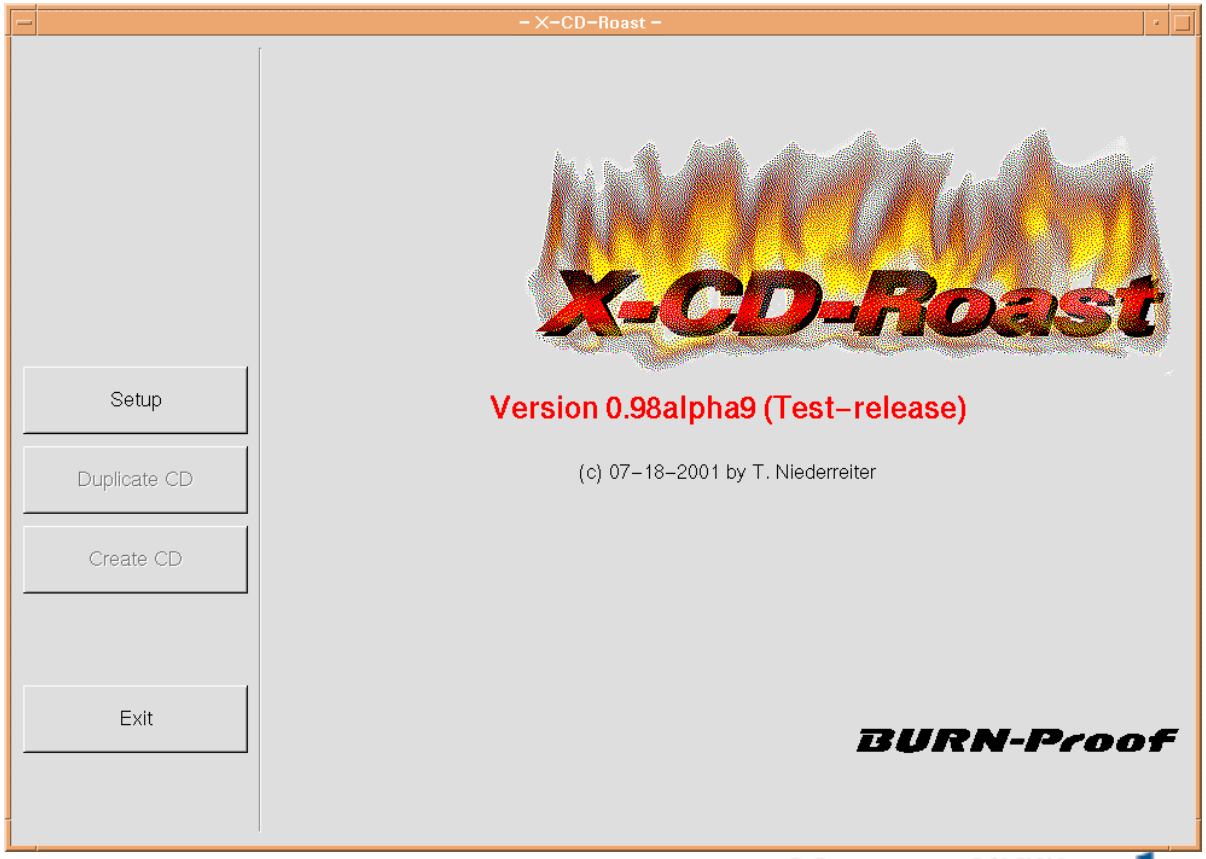
Step by Step Procedures

1. Create depot or archive (I'll use *archive.gz* in this example).
2. Determine size of archive and transform into raw format:
   ```
   # du –sk archive.gz
   # lvcreate –L <size of archive in KB>  -n image  vg01
   # newfs –F hfs –f 2048 –b 16384  /dev/vg01/rimage
   # mkdir /image
   # mount /dev/vg01/image  /image
   # cp –p archive.gz  /image/archive.gz
   # umount /image
   # dd if=/dev/vg01/rimage  of=/IGNITE/image  bs=1024k
   ```
3. Create LIF file from appropriate entry in INDEX:
   ```
   # make_media_lif –c "My INDEX Selection" –l  /IGNITE/LIF-file
   ```
4. Optional – Add custom scripts to LIF-file:
   ```
   # lifcp –r  /IGNITE/MYSCRIPTS  /IGNITE/LIF-file:MYSCRIPTS
   ```
5. Combine LIF file and raw archive image:
   ```
   # instl_combine –F /IGNITE/LIF-file –C /IGNITE/image
   ```
6. Burn CD:
   ```
   # cdrecord –v speed=12  dev=2,4,0  /IGNITE/image
   ```

HP-UX Supported CD-RW Utilities

The Mar. 2002 HP-UX Applications CDs include the CDRW depot:

- includes the public domain “cdrecord”
- includes the public domain GUI “xcdroast
- work with most IDE or SCSI CD-RW drives



Where to go for more information

Reference Material

- **www.software.hp.com/products/IUX/index.html**
  - Latest versions with Release Notes
  - Documentation
  - Technical Papers
  - FAQ
  - IUX mailing list information (email notification)

- **Documents and man pages shipped with the IUX product**
  /opt/ignite/share/doc
  /opt/ignite/share/man

- **Formal HP Training – Class # H1978S (3 Days)**

QUESTIONS?