Autodesk®
Burn® 2010

Installation and User Guide

Autodesk®

Trademarks

Disclaimer

Published by: Autodesk, Inc.
111 McInnis Parkway
San Rafael, CA 94903, USA

Title: Autodesk Burn 2010 Installation and User Guide
Document Version: 2
Date: March 18, 2009
Part Number: 2010-0021

Contents

Introduction 1

Topics in this chapter:

About Burn

Autodesk® Burn® is a Linux®-based network processing solution for Autodesk® Inferno®, Autodesk® Flame®, Autodesk® Flint®, Autodesk® Smoke®, and Autodesk® Backdraft® Conform. Burn allows you to render images in the background using low-cost Linux systems, and frees your workstation for more creative tasks. With Burn, facilities can leverage a larger pool of rendering power so that complex visual effects requiring intensive processing can be created more quickly.

As part of the Autodesk® Backburner™ Distributed Queueing System, a network of Burn render nodes can render multiple tasks in the background, dramatically increasing the efficiency of the visual effects and finishing processes.

Burn render nodes equipped with GPU-accelerated graphics boards are capable of rendering complex jobs, such as floating point jobs. See Hardware and Software Requirements on page 5 for the list of supported render node hardware.

By combining a fast network connection such as Infiniband® technology with a GPU-accelerated graphics board, Burn can deliver extremely fast background rendering of timelines, Batch setups, or any other job that supports background rendering.

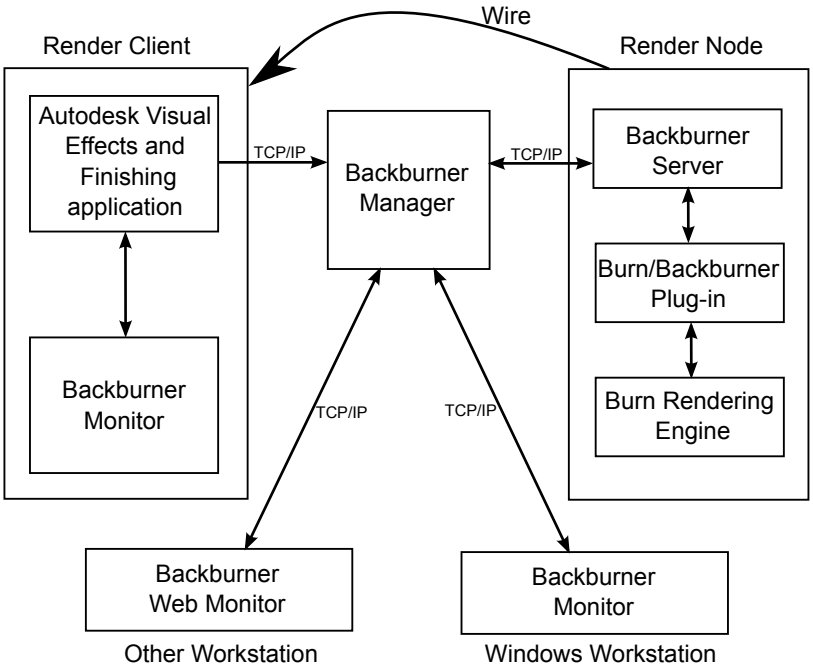
You can also benefit from the fast background rendering provided with Burn when using the Sparks® API to port your custom Autodesk Developer Network Sparks plug-ins to the Linux® environment.

Architectural Overview

The components of the Burn architecture include the following:

- An Autodesk Visual Effects and Finishing application (the render client) that sends jobs to the Distributed Queueing System
- At least one Linux computer that does the rendering (the render node)
- A workstation that distributes and manages the jobs running on the Distributed Queueing System (the Backburner Manager)
- The Burn and Backburner software components running on the render nodes on the Distributed Queueing System
- At least one workstation that monitors the jobs running on the Distributed Queueing System (the Backburner Monitor)

The following illustration shows the workflow between the components on a Distributed Queueing System.



Render Client This is the Autodesk application, such as Inferno, Flame, Flint, Smoke, and Backdraft Conform, running on a Linux workstation. From here, you create and submit rendering jobs to be processed by the Distributed Queueing System. Each setup or clip submitted for processing is called a *job*.

Backburner Manager This is the hub of the Distributed Queueing System, and it runs on a Windows® or Linux workstation. The render client submits jobs to the Backburner Manager, which distributes them to the render nodes available on the network, according to the job type.

Backburner Manager runs as either a service on Windows or a daemon on Linux. Backburner Manager starts automatically when the system is booted and runs continuously until either the workstation is shut down or the service/daemon is stopped. You can also start Backburner Manager manually as an application from the Windows Start menu.

Burn Render Node This is a Linux system on the Distributed Queueing System that hosts the Burn Rendering Engine. Jobs received from a render

client via Backburner Manager are assigned to the Rendering Engine on the render node via Backburner Server.

Render nodes use common network protocols like TCP/IP and/or Autodesk® Wire® to receive source frames and to return rendered frames back to the render client.

A render node's ability to render certain types of jobs depends on its hardware capabilities. render nodes without GPU-accelerated graphics boards cannot render jobs that require a GPU (such as floating point jobs). They can only render jobs in software mode, using the OSMesa API. Render nodes equipped with GPU-accelerated graphics boards can render both jobs that require a GPU and OSMesa jobs. To see the hardware capabilities of a Burn node in your rendering network, use the *verifyBurnServer* tool, or Autodesk Backburner Manager. See Testing Render Node Hardware for Burn on page 61, and the *Autodesk Backburner User Guide*.

Backburner Monitor This is the user interface for the Distributed Queueing System. Backburner Monitor runs as an application on a Windows system. Backburner Web Monitor runs in a Web browser from any workstation on the network. Either version allows you to view and control jobs currently being processed. Jobs in the Distributed Queueing System can be stopped, restarted, reordered, archived, or removed. You can also monitor the overall health of the Distributed Queueing System and identify any render nodes that are not working.

Backburner Server This is an application that runs on each render node in the Distributed Queueing System. Backburner Server accepts commands from Backburner Manager to start and stop the Rendering Engine for the assigned rendering tasks on the render node.

Burn/Backburner Plug-in This provides the communication link between the Backburner Server and the Burn Rendering Engine.

Each Autodesk application uses its own plug-in (such as the Burn/Backburner Plug-in) to communicate with its Rendering Engine via Backburner. This architecture allows multiple Autodesk applications to share the same render node for a variety of background rendering tasks, such as rendering 3D models or transcoding media between video formats.

There are separate Burn/Backburner plug-ins for GPU-enabled render nodes and for render nodes without GPUs. The Burn installation script detects the presence of a GPU in the render node, and installs the appropriate plug-in.

Rendering Engine This is the process that renders frames from jobs submitted from render clients. Specifically, this refers to the Burn application that renders

frames from jobs submitted from Inferno, Flame, Flint, Smoke, or Backdraft Conform.

The Rendering Engine is installed on each render node. By installing multiple rendering engines on a render node, the render node is able to render jobs from different render clients.

Wire This is the networking mechanism that is part of Autodesk® Stone® and Wire®. render nodes use Wire to transfer source frames from the render client, and to return the rendered frames back again. Wire is only required for applications that use Burn.

Hardware and Software Requirements

The minimum and recommended hardware requirements for each participating render node running Burn in a Distributed Queueing System are as follows.

Component	Minimum	Recommended
CPU	Two single-core, or one dual-core 64-bit processor, such as AMD™ Opteron™ or Intel® Xeon®	Two quad-core 64-bit processors, such as the Intel Xeon E5472 CPU at 3.0 GHz
Memory	8 GB or higher	Same amount of memory as your Visual Effects and Finishing workstation
Hard Disk	20 GB or higher SATA, Ultra-SCSI 320, SAS, or IDE drive. *Note:* The system disk must be a single, physical hard disk drive. It cannot be a logical volume from an array of disks.	
Network card	On-board Gigabit Ethernet card based on the BroadCom NetXtreme 5704 chipset	Infiniband card, if you want to run Burn and Autodesk® Incinerator® on the same render node

Component	Minimum	Recommended
GPU-accelerated Graphics Board	None. *Note:* Burn nodes without a GPU-accelerated graphics board cannot render jobs that require a GPU, such as floating point jobs.	NVIDIA® Quadro® FX 3700 *Note:* The NVIDIA Quadro FX 3700 is the only certified GPU-accelerated card for Burn nodes.
Operating System	CentOS 4.6 *Note:* If Fedora™ Core 4 is already installed on your Burn node, you do not need to replace it with CentOS. On new nodes, however, the minimum requirement is CentOS 4.6.	The custom Autodesk distribution of Red Hat® Enterprise Linux® Workstation 4 Update 3

NOTE See the application release notes for information on the version of the *Discreet Kernel Utilities* (DKU) required on Burn render nodes for this release.

Autodesk recommends the HP® ProLiant DL160 G5 with the NVIDIA Quadro FX 3700 graphics board as the optimal hardware configuration for a Burn node.

NOTE The HP ProLiant DL160 G5 system equipped with an Infiniband card, and running the custom Autodesk distribution of Red Hat Enterprise Linux Workstation 4 Update 3 is the only supported platform for running both Burn and Autodesk Incinerator on the same render node. See HP ProLiant DL160 G5 Render Node on page 77.

NOTE If you did not purchase your render nodes from Autodesk, check the Red Hat or CentOS Web sites to ensure that the hardware for your render nodes is supported by Red Hat Enterprise Linux Workstation 4 or CentOS 4.6.

The recommended Burn requirements for Sparks developers are available in the *Configuration Guide for Autodesk Developer Network Sparks Plug-in Developers*.

Compatibility

Each version of Burn is compatible with only one version of Autodesk Visual Effects and Finishing applications. Burn 2010 is compatible with Inferno 2010,

Flame 2010, Flint 2010, Smoke 2010, and Backdraft Conform 2010. Running earlier versions of Burn with these applications is not recommended.

If you run multiple versions of Autodesk applications on your Linux workstation, you can run different versions of Burn, although not at the same time. To do this, do one of the following:

- For Burn 2.0 and later, start the Autodesk Visual Effects and Finishing application(s).
- For versions of Burn earlier than 2.0, manually start the Burn client that is compatible with the version of the Autodesk application(s) you are using. See Synchronizing a Burn 1.x Client with the Autodesk Application on page 40.

About this Guide

Intended Audience

This guide assumes that you have knowledge of the following:

- Autodesk applications, specifically Inferno, Flame, Flint, Smoke, or Backdraft Conform
- Linux, UNIX, and Windows operating systems
- Computer hardware and networking

Do not attempt to carry out the procedures if you are unfamiliar with these. Contact Autodesk should you require further assistance. See Contacting Customer Support.

NOTE Most procedures described in this guide require root privileges.

Notation Conventions

A number of style conventions are used throughout your documentation. These conventions and examples of their use are shown as follows.

Convention	Example
Text that you enter in a command line or shell appears in Courier bold. Press the Enter key after each command.	**`install rpm -qa`**
Variable names appear in Courier, enclosed in angle brackets.	`<filename>`
Feedback from the command line or shell appears in Courier.	`limit coredumpsize`
Directory names, filenames, URLs, and command line utilities appear in italics.	*/usr/discreet*

Related Documentation

Documentation for this release is installed with the product as PDF files and as an HTML help system, and is also available on the Autodesk web site at *http://www.autodesk.com/me-documentation*. From this page you can access the complete documentation library.

You should also refer to the product release notes for all late-breaking release information.

Contacting Customer Support

For Autodesk® Media and Entertainment Customer Support, visit *http://www.autodesk.com/support*.

Customer support is also available through your Autodesk reseller. To find a reseller near you, consult the reseller look-up database at *http://www.autodesk.com/resellers*.

Installing and Configuring Burn

2

Topics in this chapter:

Installation Overview

Burn and Backburner components interact across Linux and Windows platforms over the network. As a result, Burn and Backburner components must be installed on each of these platforms. The following outlines the general workflow for installing Burn and Backburner.

To install Burn and Backburner:

1. Install the Backburner package on the Linux or Windows workstation that will act as the Backburner Manager for submitted Burn jobs. See Installing Backburner Manager on page 10.

2 Configure Burn-related keywords for the Autodesk Visual Effects and Finishing application, and configure the Linux workstation to allow access to its framestore from other Burn components. See Configuring Your Visual Effects and Finishing Application for Burn on page 11.

3 Install the Linux operating system, and Burn on each render node to be used in the rendering network, and optionally configure the Infiniband™ card. See Installing and Configuring Burn on the Render Nodes on page 16.

4 Activate each render node so it can receive and process jobs from Backburner Manager. See Connecting the Render Node to Backburner Manager on page 24.

5 License the systems on your rendering network. See Licensing Burn on page 26.

The required components and packages for each corresponding platform are outlined in the following table.

Component	Platform	Installation Package Content
Backburner	Intel or AMD system running Microsoft® Windows® XP Professional Service Pack 2, Microsoft Windows Vista®, Red Hat Enterprise Linux Workstation 4.0, or CentOS 4.6	The Windows Backburner installation package installs Backburner Manager and Backburner Monitor. The Linux Backburner Manager installation package installs Backburner Manager only.
Burn	Intel or AMD-based render node running 64-bit version of Red Hat Enterprise Linux Workstation 4.0 or CentOS 4.6	The Burn installation package installs Burn, Backburner Server, and Stone and Wire for Linux.

Installing Backburner Manager

Backburner Manager acts as the communications hub for a Distributed Queueing System. In Burn 2010, Backburner Manager works on either Windows or Linux workstations running the following operating systems:

- Windows XP Professional (32 bit and 64-bit) Service Pack 2, as well as Windows Vista (32-bit and 64-bit)

- Red Hat Enterprise Linux Workstation 4.0 (64-bit distribution)
- CentOS 4.6 (64-bit distribution)

Refer to the *Autodesk Backburner Installation Guide* for instructions on installing Backburner Manager. It contains information and procedures on how to install, configure, and use Backburner for a variety of Autodesk applications, including Burn.

Configuring Your Visual Effects and Finishing Application for Burn

Your Autodesk Visual Effects and Finishing application can use Burn by default; no additional software is required. To configure your application and workstation to use Burn, do the following:

- Configure keywords in the software initialization configuration file (*init.cfg*) of Inferno, Flame, Flint, Smoke, or Backdraft Conform. See Configuring Burn-related Keywords for the Application on page 11.
- Set up export permissions for your workstation to allow render nodes access to job information. See Enabling Export Permissions on page 14.
- Check the *sw_framestore_map* file to allow render nodes access to job media. See Viewing the sw_framestore_map File on page 15.

Configuring Burn-related Keywords for the Application

You must set up four keywords in the software initialization configuration file (*init.cfg*) of Inferno, Flame, Flint, Smoke, or Backdraft Conform to connect the application to the Backburner Manager and the rendering network. To set up these keywords, do the following procedure for each application installed on your Linux workstation.

NOTE Some of these keywords are already enabled if you are using Background Wire and Background Proxies in your Visual Effects and Finishing application.

NOTE Exit the application before changing these keywords, and restart the application after the keywords have been changed.

To set up Burn-related keywords for your application:

1. Log in to the account for your Autodesk Visual Effects and Finishing application and open a terminal.
2. Open the software initialization configuration file (*init.cfg*) for your application by typing:

   `dlcfg`

   The *init.cfg* file for your Autodesk application appears in a text editor.
3. Locate the following keyword:

   `#BackburnerManagerHostname <hostname>`

   This keyword identifies the workstation running the Backburner Manager for the rendering network.
4. Configure the *BackburnerManagerHostname* keyword to allow your application to locate the Backburner Manager for the rendering network by doing the following:
   - Remove the pound sign (#) from the beginning of the line.
   - Replace `<hostname>` with the name or IP address of the Windows or Linux workstation running Backburner Manager for the rendering network.
5. Locate the following keyword:

   `#BackburnerManagerPriority 50`

   This optional keyword sets the priority for all Burn jobs submitted by the application to the Backburner Manager. The Manager uses job priority to determine the order in which jobs are processed. By default, all submitted jobs are given the same priority, which allows Backburner Manager to manage jobs and network resources automatically. Use this keyword to override the Manager's default behavior when processing your Burn jobs.

   Changing the *BackburnerManagerPriority* keyword may compromise access to the Distributed Queueing System for others in your facility. Autodesk does not recommend changing its value, except if you are in a client session and/or facing a deadline.
6. Configure the *BackburnerManagerPriority* keyword to set the default priority for all jobs submitted to the Distributed Queueing System by doing the following:
   - Remove the pound sign (#) from the beginning of the line.

- Change the priority from 1 (most important) to 100 (least important).

7 Scroll down to the following line:

```
#BackburnerManagerGroup <groupname>
```

This optional keyword specifies a Server group (a preset group of render nodes) used to process all Burn jobs submitted by the application. By default, Backburner Manager assigns a job to all available render nodes capable of processing it. Use this keyword if you already have a dedicated group of render nodes for processing Burn jobs.

8 Configure the *BackburnerManagerGroup* keyword to set the default server group used to process all jobs submitted to the Distributed Queueing System. Replace `<groupname>` with the name of the render node group responsible for processing Burn jobs.

9 Scroll through the file and locate the following keyword:

```
BackburnerManagerGroupCapability
```

This keyword specifies whether the nodes in your rendering network are equipped with GPUs or not. Based on the value of this keyword, the Visual Effects and Finishing application enables or disables the submission of jobs that require a GPU (such as floating point jobs) to the rendering network.

10 Configure the `BackburnerManagerGroupCapability` keyword according to the GPU capabilities of the nodes in your rendering network:

- If none of the nodes in your rendering network are equipped with GPUs, set the value of the `BackburnerManagerGroupCapability` keyword to *software*. The application does not send jobs that require a GPU to the rendering network, but only jobs that can be rendered in software mode (using OSMesa) by the Burn render nodes.
- If ALL the nodes in your rendering network are GPU-enabled, set the value of the `BackburnerManagerGroupCapability` keyword to *gpu*. The application sends all jobs to the GPU-equipped Burn nodes in the rendering network, even if some jobs do not specifically require a GPU Burn node. The GPU-equipped render nodes will render jobs that require a GPU, as well as OSMesa jobs.

NOTE If your rendering network also contains Burn nodes without a GPU, but the `BackburnerManagerGroupCapability` keyword is wrongfully set to *gpu*, all jobs are sent only to GPU-equipped render nodes, and the nodes without a GPU are never used.

- If your rendering network contains a mix of nodes with GPUs and without GPUs, set the `BackburnerManagerGroupCapability` keyword to *hybrid*. The application sends all jobs to the rendering network, and Backburner Manager distributes each job to the appropriate type of Burn render node. Jobs that require a GPU are sent only to GPU-equipped Burn nodes, while jobs that do not require a GPU are sent to any available Burn render node (GPU or non-GPU), to be rendered in software mode.

NOTE Set the `BackburnerManagerGroupCapability` keyword to *gpu* or *hybrid* only if you are sure that at least one node in your rendering network is equipped with a GPU. Attempting to submit a job that requires a GPU to a rendering network with no GPU-enabled Burn nodes results in the job being stuck in the rendering queue indefinitely.

11 Save and close the *init.cfg* file.

Enabling Export Permissions

Edit the Exports file (*/etc/exports*) on the Linux workstation so that render nodes on the Distributed Queueing System can access the framestore of the jobs submitted to Burn. Enable access to the following directories.

Directory	Description
/usr/discreet/clip	Enables remote access to clip libraries for Wire.
/usr/discreet/project	Enables remote access to projects by Burn at Burn start-up.

In the Exports file, you can enable access for the rendering network to the directories specified in the preceding table.

To edit the Exports file:

1 Log in as root to the workstation and open a Linux terminal.

2 Open the Exports file in a text editor.

3 Enable access to the */usr/discreet* directory on your workstation by adding the following line to the end of the Exports file:

```
/usr/discreet *(rw,sync,no_root_squash)
```

NOTE The previous line allows any remote system access to the */usr/discreet* directory. You can restrict access by adding the hostnames for each render node to the line, separated by a colon (:). For example, to restrict access to the */usr/discreet* directory for a Linux workstation to nodes *burn01* to *burn03*, type: **`/usr/discreet burn01:burn02:burn03(rw,sync,no_root_squash)`**.

4 Save and close the file.

5 Enable the changes you make by typing:

```
/usr/sbin/exportfs -va
```

Viewing the sw_framestore_map File

Verify that the *sw_framestore_map* file contains only the local framestore. This procedure is necessary for the SelfDiscovery option in Stone and Wire to work correctly.

To view the sw_framestore_map file:

1 Log in as root to the workstation and open a terminal.

2 Type:

```
cd /usr/discreet/sw/cfg
```

3 Use a text editor to open *sw_framestore_map*.

4 Verify that only the local framestore appears in this file. If other framestores appear, delete them, unless server self-discovery is not enabled for Stone and Wire.

NOTE Framestores must be explicitly listed in the *sw_framestore_map* file if server self-discovery is not enabled for Stone and Wire, such as when you are using legacy disk arrays. Use *sw_framestore_dump* to see if storage devices for other workstations on the network can be viewed. See the *Filesystem and Networking Guide*.

5 Save and close the file.

Installing and Configuring Burn on the Render Nodes

Install Burn on each Linux render node on the Distributed Queueing System. The following procedure outlines the general workflow for installing Burn on a render node.

To install Burn on each Render Node:

1. Optional: Install Red Hat Enterprise Linux Workstation or CentOS. See Installing Linux on a Burn Render Node on page 67.
2. Optional: Configure the render node to use an Infiniband network if you plan to integrate the render node into an Infiniband-connected render network. See Configuring an Infiniband Card on page 16.
3. Configure the network settings so the render node can communicate with other Burn and Backburner components on the network. See Configuring the Linux Network Settings on page 17.
4. Install Burn on each render node from the application DVD. See Installing Burn on the Render Nodes on page 22.
5. Connect the render node to Backburner Manager. See Connecting the Render Node to Backburner Manager on page 24.
6. Optional: If the Burn render node has a slower connection to a shared storage device (such as a SAN) than the Visual Effects and Finishing workstation it is recommended to disable local Stone and Wire I/O on the render node. See Disabling Local Stone and Wire I/O on Burn Nodes on page 25

Configuring an Infiniband Card

If you are planning to use the render node in an Infiniband-connected rendering network, you must meet the following two conditions:

- Your rendering network must use an Infiniband network.
- The render node you are configuring must be equipped with an Infiniband network adapter.

If you meet both conditions, you can configure the render node to use the Infiniband network.

The precompiled QuickSilver InfiniServ 7000 HCA adapter drivers for the Red Hat Enterprise Linux kernel are included in the *dist/ib* subdirectory of the Burn installation package, for example *Burn_2010_LINUX64_RHEL4/dist/ib*.

If you are using Cent2010OS, you need to manually compile the Infiniband driver for your version of the Linux kernel. The source files for the driver are located in the *src/infiniband* subdirectory of the latest DKU installation package, for example *DKU_4.0.0/src/infinband*.

NOTE Refer to the README file located inside the driver tar file for instructions on how to install the driver and configure your Infiniband interface.

Configuring the Linux Network Settings

Configure the Linux network settings of the render node so that Burn can communicate with other Burn and Backburner components on Linux and Windows workstations. Do the following for each render node to configure its Linux network settings:

- Set up NFS access to the Linux workstations submitting rendering jobs. See Setting Up NFS Access to Workstations on page 17.
- Modify the *amd* automounter configuration file to redirect mount points. See Modify the amd Automounter Configuration File on page 18.
- Put a generic host name in the loopback settings. See Modifying the Loopback Host Name on page 18.
- Enable Multicasting across the NFS and TCP/IP networks. See Enabling Multicasting on page 19.
- Configure the *NFS* and *amd* services on the render node. See Configuring the NFS and amd Services to Start Automatically on page 21.

Setting Up NFS Access to Workstations

NFS access to the Linux workstations at */hosts/*`<hostname>` is required, where `<hostname>` is the name of each workstation submitting rendering jobs. You can use the *amd* automounter installed by the custom kickstart file, or create a static mount point in */etc/fstab*. See the *man* pages for *mount*, *fstab* and *amd* to configure NFS access to other workstations on your network.

Modify the amd Automounter Configuration File

If you are using the *amd* automounter, do the following to modify the *amd* configuration file to redirect the mount point to */hosts* instead of */net*.

To modify the amd automounter configuration file:

1. Log in as root to the render node.
2. Stop the *amd* automounter daemon by typing:
   `/etc/init.d/amd stop`
3. Open the */etc/amd.conf* configuration file in a text editor.
4. Locate the following lines:
   ```
   #DEFINE AN AMD MOUNT POINT
   [ /net ]
   ```
5. Change */net* to */hosts*.
   The lines should now be the following:
   ```
   #DEFINE AN AMD MOUNT POINT
   [ /hosts ]
   ```
6. Save and close the file.
7. Restart the *amd* daemon. Type:
   `/etc/init.d/amd start`

Modifying the Loopback Host Name

By default, Red Hat Linux sets the loopback setting with the host name of the Linux server. However, Stone and Wire require a generic host name in the loopback setting so that the network functions correctly for background rendering. To enable communication with Stone and Wire, modify the loopback host name in the */etc/hosts* file on all Linux servers.

To modify the loopback host name:

1. Log in as root.
2. Using a text editor, open the file */etc/hosts*.
   The *hosts* file appears. By default, the loopback setting is identified by the IP address 127.0.0.1, for example:
   ```
   127.0.0.1 discreet.localhost.localdomain localhost
   ```

3 Remove the host name preceding the `localhost.localdomain` from the loopback setting.

   For example, if you remove the host name `discreet`, which is the default name assigned to Burn render nodes in a Red Hat Enterprise Linux Workstation installation, the loopback setting should appear as follows:

   ```
   127.0.0.1 localhost.localdomain localhost
   ```

4 On a new line, type the IP address for the render node, press **Tab**, and then enter the explicit host name for the node.

   The new line defines the network IP address and host name for the render node. For example, if your render node's IP address is 172.16.100.22 and its explicit host name is *burn_02*, your *hosts* file should appear as follows:

   ```
   127.0.0.1 localhost.localdomain localhost
   172.16.100.22 burn_02
   ```

5 Save and close the *hosts* file.

Enabling Multicasting

Stone and Wire require multicasting throughout the connected Linux and Windows networks. Enable multicasting on the Burn render node by doing the following:

- Use the *netconfig* utility to set the general network parameters for the render node.
- Check the IP address, hostname, and network gateway in */etc/sysconfig/network*.
- Check the route to the network gateway for the ethernet adapter configuration file in */etc/sysconfig/network-scripts/ifcfg-*`<xxx>`.

To use the netconfig utility to set up network parameters:

1 Log in to the render node as root.

2 Start the *netconfig* utility by typing:

   `netconfig`

   The *netconfig* utility appears, asking if you want to set up general networking parameters for the render node.

3 Choose Yes.

A screen appears where you can enter networking parameters, such as the IP address.

4 Do not enable DHCP/NNTP name resolution.

5 Enter valid IP addresses for the following parameters.

For:	Enter:
IP address	The IP address for the Burn render node.
Netmask	The subnet mask for the network.
Default gateway	The IP address of the gateway for the network.
Primary nameserver	The IP address of the primary nameserver for the network.

NOTE Verify this information with the system administrator.

6 Click OK.

The *netconfig* utility sets up the networking parameters for the render node.

7 Continue to the next procedure to make sure the hostname and network gateway for the render node are set up properly in the */etc/sysconfig/network* file.

To check the /etc/sysconfig/network file:

1 Log in as root.

2 Using a text editor, open the file */etc/sysconfig/network*.

The *network* file appears. Your file should look similar to the following example:

```
NETWORKING=yes
HOSTNAME=discreet
GATEWAY=62.122.5.83
```

By default, the hostname of a render node is set to `discreet` when you use the kickstart file to install Red Hat Enterprise Linux Workstation or CentOS. Change this to the actual hostname for the render node.

3 Change the `HOSTNAME` keyword to the explicit hostname for the render node.

4 If either the GATEWAY keyword or the IP address for the network gateway are missing, enter this information.

NOTE Be sure to verify this information with the system administrator.

5 Save and close the *network* file.

6 Continue to the next procedure to make sure network gateway is properly set up in the */etc/sysconfig/network-scripts/ifcfg-<xxx>* file.

To check the */etc/sysconfig/network-scripts/ifcfg-<xxx>* file:

1 As root, view the *ifcfg-<xxx>* file by typing:

more /etc/sysconfig/network-scripts/ifcfg-<xxx>

where <xxx> is the interface value assigned to the network card, such as eth1. Your file should look similar to the following example:

```
DEVICE=eth1
ONBOOT=yes
IPADDR=172.200.17.51
NETMASK=255.255.255.0
```

NOTE You can associate network gateway routes with specific interfaces if a default gateway route in */etc/sysconfig/network* is not set. However, a default gateway route in */etc/sysconfig/network* overrides the *ifcfg-<xxx>* default routes. See the man page for *ifconfig* or contact your system administrator to determine the method for enabling multicasting that is best suited for your network.

2 If the GATEWAY keyword is missing, add the IP address for the network gateway using a text editor.

NOTE Verify this information with the system administrator.

Configuring the NFS and amd Services to Start Automatically

By default, the *NFS* and *amd* automounter services must be started manually once the render node has booted. Once you have set up all network settings, do the following procedure to start these services automatically with the render node.

To configure NFS and amd as start-up services:

1. Log in as root to the render node.
2. Configure the *NFS* and *amd* automounter services to be started automatically by typing the following commands:

   `chkconfig nfs on`

   `chkconfig amd on`
3. Confirm that the *NFS* and *amd* automounter services are configured to start at run-levels 2 to 5.

   - Type: **`chkconfig --list | grep`** `<service name>`

   where `<service name>` is either `nfs` or `amd`.
4. Reboot the render node to ensure all network settings take effect. Type:

   `reboot`

   Once the render node has rebooted, you are ready to install Burn.

Installing Burn on the Render Nodes

Install Burn and Backburner Server on each render node on the rendering network using the provided installation script. You can install Burn 2010 along with earlier versions of Burn.

To install Burn:

1. Log in to the render node as root, and open a terminal.
2. If you are installing from an application DVD, mount the disc, and open the Burn installation directory. For example:

   `cd /mnt/cdrom/Burn_2010_LINUX64_RHEL4`

   NOTE If using the HP ProLiant DL160 G5 Render Node, the correct path is **`cd /media/cdrom/Burn_2010_LINUX64_RHEL4`**.

   NOTE The same Burn installation package is used for Red Hat Enterprise Linux and CentOS.
3. If you are installing from a downloaded 2010 *tar* file, unpack the file to a temporary directory.
4. Run the installation script, by typing:

./INSTALL_BURN

The Burn, and Backburner Server packages are installed. Step 5 is required if graphics are enabled on the render node. Otherwise, go to step 6.

5 If graphics are enabled and the following prompts appear, click Yes to each one.

- **Backburner Manager configuration: Do you want to automatically run the Backburner Manager on this machine?** Click yes
- **Backburner Server configuration: Do you want to automatically run the Backburner Server on this machine?** Click yes
- **Backburner Server configuration: Do you want to enter the manager for this server?** Click yes
  The *manager.host* file appears. If you know the hostname or IP address of your Backburner Manager, enter it here. Otherwise, close it without saving.

  You may edit this file later. See the procedure for Connecting the Render Node to Backburner Manager on page 24.

6 Check that the render node can identify remote framestores using Stone and Wire. Open the */usr/discreet/sw/cfg/sw_probed.cfg* configuration file for the render node in a text editor.

The *sw_probed.cfg* configuration file appears, showing parameters for the *sw_probed* daemon.

7 Make sure that multicasting is enabled for Stone and Wire by doing the following:

- Enable the SelfDiscovery option, if necessary (by removing the pound sign on its line in the *sw_probed.cfg* file); this option should be enabled by default when Stone and Wire is installed on the render node.
- Use the Scope parameter to define the scope for the multicast.

See the entries for *sw_probed* and *sw_probed.cfg* in the *Autodesk Stone and Wire Filesystem and Networking Guide*.

NOTE If the workstations and Burn nodes in your facility are on separate networks connected through routers, use the *ttl* option in the *sw_probed.cfg* file to specify the number of router hops for a multicast. Burn-related transfers across multiple routers may cause bottlenecks at network bridges, especially with jobs involving film frames. Using the *ttl* option may reduce multicast-related traffic and improve general network performance in your facility.

8. Save and close the *sw_probed.cfg* configuration file.
9. Restart the *sw_probed* daemon. Type:

   ```
   /etc/init.d/stone+wire restart
   ```

10. If necessary, install additional fonts on the render nodes that are not included with your Autodesk application.
11. Activate the render node so it is ready to receive jobs from Burn.

Installing Fonts on the Linux Servers

When you install Burn, the same fonts that are installed by default with your Autodesk Visual Effects and Finishing application are also installed on the render nodes. However, if you installed additional fonts not provided with your application, you must also install those fonts on the render nodes. Contact your third-party font supplier(s) for information about Linux support for those fonts.

Verify that any 3D Text fonts used with Action nodes in the Batch setups you submit to Burn are installed.

Connecting the Render Node to Backburner Manager

With Burn installed, validate the connection to Backburner Manager using the following procedure to check settings in the *manager.host* configuration file. You must repeat this procedure for each render node on the network.

To specify the Backburner Manager connection:

1. Log in as root on the render node.
2. In a text editor, open the */usr/discreet/backburner/cfg/manager.host* configuration file.

3 Replace the text in the file with the hostname or IP address of the workstation running Backburner Manager. For example:

172.16.25.2

You can test these values by using the Backburner Monitor, or by pinging the Backburner Manager's IP address or hostname from the node. Contact your network administrator if you are unsure of these values.

NOTE The name or IP address you specify here for Backburner Manager must be the same as the one specified in the application's software initialization configuration file (*init.cfg*) and cannot contain comments. See Configuring Burn-related Keywords for the Application on page 11.

---

4 Save and close the *manager.host* file.

5 Restart Backburner Server on the render node by typing:

/etc/init.d/backburner_server restart

Disabling Local Stone and Wire I/O on Burn Nodes

If the Burn render node has a slower connection to a shared storage device (such as a SAN) than the Visual Effects and Finishing workstation it is preferable for Stone and Wire I/O operations to be performed on the Visual Effects and Finishing workstation, rather than on the render node.

Perform the following procedure to disable all local Stone and Wire I/O operations on a Burn render node.

To disable local Stone and Wire I/O operations:

1 Open a terminal on the Burn render node, and log in as root.

2 Stop Stone and Wire by typing:

/etc/init.d/stone+wire stop

3 Open the */usr/discreet/sw/cfg/stone+wire.cfg* configuration file in a text editor.

4 Locate the *DisableLocalIO* keyword in the *[Initialization]* section.

5 Uncomment the keyword if necessary, and set its value to true:

```
DisableLocalIO=True
```

6 Save and close the configuration file.

7 Restart Stone and Wire by typing:

```
/etc/init.d/stone+wire start
```

Licensing Burn

Although you can install Burn without a license, you must license it before you can render jobs submitted from Inferno, Flame, Flint, Smoke, or Backdraft Conform. In Burn, you use the License Server to license Burn for render nodes on the network. The following procedure outlines how to license Burn for your rendering network.

To license a Burn network:

1 Install the License Server for Burn. See Installing the Burn License Server on page 26.

2 Obtain a license code for the License Server. See Obtaining License Codes for Burn on page 27.

3 Configure the License Server to distribute licenses to the render nodes. See Configuring the License Server on page 28.

4 Configure each render node to retrieve a license from the License Server for rendering Burn jobs. See Configuring Licensing for Render Nodes on page 29.

Installing the Burn License Server

Install the License Server to set up a licensing system that provides “floating” licenses for all Burn nodes on the Distributed Queueing System.

There are two components to this system.

License Server A Linux daemon that provides concurrent licenses to render nodes on the Distributed Queueing System as needed.

Licensing client The render node on the network running Burn that requests a license from the License Server.

Perform the following procedure to install the License Server on a render node designated as the License Server for the Distributed Queueing System.

To install the License Server:

1. Log in to the Linux host designated as the License Server as root.
2. If you are installing from the application DVD, mount the disc, and open the Burn installation directory, for example:

   ```
   cd /media/cdrom/Burn_2010_LINUX64_RHEL4
   ```

   NOTE The same Burn installation package is used for Red Hat Enterprise Linux and CentOS.

3. If you are installing from a downloaded *tar* file, unpack the file to a temporary directory.
4. Check the contents of the directory. Type:

   ```
   ls
   ```

   Verify that the directory contains the installation script file *INSTALL_LICSERV*.

5. Type:

   ```
   ./INSTALL_LICSERV
   ```

   The License Server is installed. Next, obtain Burn license codes for the Distributed Queueing System.

Obtaining License Codes for Burn

You can obtain license codes by registering the application with the Autodesk Media and Entertainment Licensing Department by e-mail, or telephone.

NOTE For emergencies, you can acquire an immediate temporary license code by going to the Autodesk Web page (*www.autodesk.com*), clicking the Services & Support link, selecting your product, then clicking Register Your Product and following the step-by-step instructions. A temporary license code is e-mailed to the address you provide.

To obtain license codes for Burn, you must generate a unique host ID for the system acting as the Burn License Server, using a utility installed with the License Server. This ID is used to confirm your Burn license(s) and issue license codes for your render nodes.

To obtain license codes for Burn:

1. Log in as root on the render node where the License Server is installed.
2. Go to the */usr/local/bin* directory:

   `cd /usr/local/bin`
3. Run the *dlhostid* utility to generate a unique host ID for the system. Type:

   `./dlhostid`

   A message indicating the host ID of the system appears:

   `The Discreet host ID of this machine is "DLHOST01=<host ID>"`

   You send the host ID of the system to the Licensing Department to receive the license codes for the render nodes on your network. When you record the host ID, make sure you include the `DLHOST01=` string as well as the number; for example, `DLHOST01=886C2B75E8E57E4B03D784C3A2100AC0`.
4. Contact the Autodesk Media and Entertainment Licensing Department using one of the following methods and register Burn to obtain license codes.

   Registering by E-mail To acquire a license code by e-mail, submit a request with the host ID to *me.support @autodesk.com*.

   Registering by Telephone You can speak to a licensing representative by calling the Licensing Department toll-free in North America at 1-800-925-6442 between 8 AM and 8 PM eastern standard time (EST). Outside of North America, call 1-514-954-7199.
5. Once you receive your license codes from the Licensing Department, you need to configure the licenses for your network. See Configuring the License Server on page 28.

Configuring the License Server

Configure the License Server with the Burn licenses you received using the following procedure.

To configure the Render Node acting as the License Server for the network:

1. Log in as root to the render node where the License Server is installed.

2 Go to the *licenses* directory for Burn by typing:

cd /usr/discreet/licserv/licenses

3 In a text editor, create a file called *DL_license.dat*.

4 Enter the information provided by the Licensing Department in this file.

This information is in a format similar to the following example:

```
SERVER exuma-001
DLHOST01=886C2B75E8E57E4B03D784C3A2100AC0
DAEMON discreet_l discreet_l
USE_SERVER
FEATURE burn_x86_64_r_2010 discreet_l 2010.999
30-nov-2009 4 \
6D8387D36A5B5F69B44A ck=59
```

5 Save and close this file.

This file sets up the floating licenses available for distribution by License Server to the render nodes on the Distributed Queueing System.

6 Start the License Server by typing:

/etc/init.d/license_server start

NOTE The License Server cannot start unless the license is entered correctly in *DL_license.dat*. Check the *boot.log* file to make sure the License Server is started and working properly.

Configuring Licensing for Render Nodes

Configure the render nodes to retrieve Burn licenses from the License Server using the following procedure. You must repeat this procedure for each render Node on the network.

NOTE If the License Server for your network is running on a render node, make sure you perform the following procedure on this node as well. Otherwise, this node is able to distribute Burn licenses to other render nodes, but is unable to retrieve a license for itself.

To configure a Render Node with a license:

1 Log in as root to the render node.

2 Create a *licenses* directory by typing:

mkdir -p /usr/local/flexlm/licenses

3 Go to this directory by typing:

cd /usr/local/flexlm/licenses

4 In a text editor, create a file called *DL_license.dat*.

5 Configure the node to contact the License Server for a license by copying the first three lines of the information provided by the Licensing Department to the *DL_license.dat* file.

This information appears similar to the following example:

```
SERVER exuma-001
DLHOST01=886C2B75E8E57E4B03D784C3A2100AC0
DAEMON discreet_l discreet_l
USE_SERVER
```

6 Save and close the file.

You are ready to run Burn with Inferno, Flame, Flint, Smoke, or Backdraft Conform.

NOTE To determine if a render node has any problems contacting or obtaining a license from the License Server, check the *backburnerServer.log* file in */usr/discreet/backburner/Network*.

Changing the Default Port Used by the License Server

In order to avoid conflicts with other applications or license servers in your facility, you may need to change the default port setting used by the License Server. This requires a minor change to the *DL_License.dat* file on the system hosting the License Server, as well as every render node on the Burn network.

To change the default port used by the License Server:

1 Log in as root to the system where the License Server is installed.

2 Go to the *licenses* directory for Burn by typing:

cd /usr/discreet/licserv/licenses

3 Using a text editor, edit the file *DL_license.dat*.

4 Find the SERVER line.

The line should look similar to the following example:

```
SERVER exuma-001
DLHOST01=886C2B75E8E57E4B03D784C3A2100AC0
```

This gives the License Server a default port number in the range of 27000-27009.

5 Enter the port address information at the end of the SERVER line.

For this example, to use a fixed port with an address of 62222 you would get the following:

```
SERVER exuma-001
DLHOST01=886C2B75E8E57E4B03D784C3A2100AC0 62222
```

6 For the vendor daemon, discreet_l, a different port number is assigned by the operating system at each runtime.You have the option of fixing this port number. Insert the following VENDOR line just before the existing DAEMON line. In this example, the *discreet_l* daemon is set to use port 12344.

```
VENDOR discreet_l discreet_l port=12344 DAEMON
discreet_l discreet_l
```

7 Save and close this file.

8 Verify that the new port settings are correct. Type the following commands:

/etc/init.d/license_server stop

/etc/init.d/license_server start

cat /usr/discreet/licserv/log/license_server.log

9 Look for messages similar to the following examples in the output, and verify that the port numbers are what you requested.

```
15:08:49 (lmgrd) lmgrd tcp-port 62222 15:08:49
(lmgrd) Starting vendor daemons ... 15:08:49 (lmgrd)
Using vendor daemon port 12344 specified in license
file 15:08:49 (lmgrd) Started discreet_l (internet
tcp_port 12344 pid 5013)
```

Now you need to edit the *DL_licenses.dat* file on every render node in the network, including the render node hosting the License Server.

10 Log in as root to the render node.

11 Go to the *licenses* directory by typing:

cd /usr/local/flexlm/licenses

12 Using a text editor, edit the file *DL_license.dat*.

13 Find the SERVER line.

The line should look similar to the following example:

```
SERVER exuma-001
DLHOST01=886C2B75E8E57E4B03D784C3A2100AC0
```

14 Enter the same port address value at the end of the SERVER line that you entered for the License Server.

For example, using port address 62222, you would get the following:

```
SERVER exuma-001
DLHOST01=886C2B75E8E57E4B03D784C3A2100AC0 62222
```

15 Save and close this file.

16 Repeat steps 10 to 15 for every render node in the network.

Background Rendering Over the Network

3

Topics in this chapter:

Background Rendering with Burn

Once Burn is installed and licensed, send jobs from your Autodesk Visual Effects and Finishing applications for rendering by the Autodesk Backburner Distributed Queueing System. The *Distributed Queueing System* refers to all the render nodes on the physical network that are used for rendering. The following procedure provides a general overview for using Burn to do background rendering and assumes that the network is configured properly, including the TCP/IP settings.

To render jobs using Burn:

1. Start the following rendering network components:
   - Backburner Manager on the Windows or Linux workstation
   - Backburner Server on each Burn render node

   See Starting Backburner Manager and Backburner Server on page 34.
2. If necessary, start the correct Burn client on the workstation. See Running Multiple Versions of Burn on page 37.
3. Prepare and submit a job from Batch, the EditDesk, or a clip library. See Submitting Jobs to the Distributed Queueing System on page 42.
4. During an active render session, monitor and manage jobs that are being processed on the network. See Monitoring and Managing Burn Rendering Jobs on page 43.
5. Preview the result of the submitted job, if necessary. See Previewing Results on page 47.

Starting Backburner Manager and Backburner Server

Backburner Manager and Backburner Server must be running before you can submit jobs to the rendering network. Use the following procedures to start Backburner Manager and Backburner Server:

- To start Backburner Manager on either a Linux or a Windows workstation, see Starting Backburner Manager on page 35.

  NOTE Only one Backburner Manager may be running on the Distributed Queueing System.

- To start Backburner Server on each Linux render node, see Starting Backburner Server on page 36.

Starting Backburner Manager

Start the Backburner Manager for the Distributed Queueing System on either the Windows or Linux workstation where it is installed by doing one of the following:

- To start Backburner Manager on a Linux workstation, see Starting Backburner Manager on a Linux Workstation on page 35.
- To start Backburner Manager on a Windows workstation, see Starting Backburner Manager on a Windows Workstation on page 36.

You need to perform only one of these procedures, since only one Backburner Manager can run at any time in a rendering network. Otherwise, Burn jobs submitted to the network may not be processed.

Starting Backburner Manager on a Linux Workstation

By default, Backburner Manager is installed as a Linux service (daemon) that automatically starts with the workstation. Use the following procedure to see if Backburner Manager is running on a Linux workstation and restart the Manager, if it is not running.

To start Backburner Manager on a Linux workstation:

1. Log in as root to the Linux workstation designated as the Backburner Manager for the rendering network and open a terminal.
2. Check if the Backburner Manager daemon (service) is running. Type:

   `ps -ef | grep -i Backburner`

   An output similar to the following appears:

   ```
   root 2997 1 0 Dec13 ? 00:00:03
   /usr/discreet/backburner/backburnerManager
   ```

   ```
   root 24204 24091 0 11:00 pts/4 00:00:00 grep -i
   backburner
   ```

   If `backburnerManager` does not appear in the output, continue to the next step to start this daemon. Otherwise, you are ready to start the Backburner Server on each render node. See Starting Backburner Server on page 36.
3. Start the Backburner Manager daemon manually by typing:

   `/etc/init.d/backburner_manager start`

Backburner Manager is started. To review the Manager's status, see the log files in the directory */usr/discreet/backburner/log*. See the *Autodesk Backburner Installation Guide*.

Starting Backburner Manager on a Windows Workstation

By default, Backburner Manager is installed under Windows as a separate application that you must start manually. Use the following procedure to start Backburner Manager on the workstation.

TIP You can also configure Backburner Manager to run as a Windows service that is started automatically with the workstation. See the *Autodesk Backburner Installation Guide* for instructions.

To start Backburner Manager on a Windows workstation:

1. Go to the Windows workstation designated as the Backburner Manager for the rendering network.
2. Choose Start | Program Files | Autodesk | Backburner | manager.

   The Backburner Manager is started and its application window appears. The message "Starting Network Manager" as well as server connection messages are displayed in the window.

   To view the status of the Backburner Manager, leave its application window open or view the log files in the *C:\Program Files\Backburner\log* folder. See the *Autodesk Backburner Installation Guide*.

Starting Backburner Server

Perform the following procedure on each Burn render node on the Distributed Queueing System.

To start Backburner Server:

➤ Go to each render node, log in as root, and then start Backburner Server by typing:

```
/etc/init.d/backburner_server start
```

Backburner Server starts and searches the network for the Backburner Manager application specified in its *usr/discreet/backburner/cfg/manager.hosts* file. Once this Backburner Manager is found, a message is logged to

indicate that Backburner Server has successfully established communication with the Manager.

Backburner Manager should also detect the Backburner Server running on each rendering node on the Distributed Queueing System. If error messages indicate that Backburner Manager cannot find Backburner Server or vice versa, see Connecting the Render Node to Backburner Manager on page 24.

Once both Backburner Manager and the Backburner Server on each render node have established connections, the Distributed Queueing System is ready to accept jobs from Burn.

Running Multiple Versions of Burn

Each version of an Autodesk Visual Effects and Finishing application includes its own version of Burn. Because your facility may have different versions of Inferno, Flame, Flint, or Smoke running with its corresponding version of Burn, you can run Burn 2010 with earlier versions of Burn.

Do the following to run multiple versions of Burn:

- Install the Burn 2010 server on each render node on the rendering network. See Installing and Configuring Burn on the Render Nodes on page 16.
- Determine the Burn client and server version needed for the version of the Autodesk Visual Effects and Finishing application that you are running. See Assessing Compatibility Between Client and Server Versions on page 38.
- Ensure there is a Burn server running on the rendering network that is capable of handling the jobs for the version of Burn you are running. See Managing Multiple Burn Servers on a Render Node on page 39.
- For Autodesk applications that require the Burn 1.x client, synchronize the Burn client with the application. See Synchronizing a Burn 1.x Client with the Autodesk Application on page 40.

Assessing Compatibility Between Client and Server Versions

For Burn versions earlier than 2.0, you must synchronize the version of the Burn client and Burn server with the version of the Autodesk Visual Effects and Finishing application you are running.

For Burn versions 2.0 and later, your Visual Effects and Finishing application sends jobs directly to the rendering network, removing the need to synchronize the client with the server.

The following table lists the compatibility between versions of Autodesk Visual Effects and Finishing applications and versions of the Burn client and server.

Burn Client Version	Burn Server Version	Visual Effects and Finishing Version
Burn client 1.0.x	Burn 1.0.x	Inferno 5.3.2, Flame/Flint 8.3.2
Burn client 1.5	Burn 1.5	Inferno 5.5, Flame/Flint 8.5
Burn client 1.6	Burn 1.6	Inferno 6.0, Flame/Flint 9.0, Fire® 6.5, Smoke 6.5
Burn client 1.7	Burn 1.7	Inferno 6.2, Flame/Flint 9.2, Fire 6.7, Smoke 6.7
N/A	Burn 2.0	Inferno 6.5, Flame/Flint 9.5,Fire 7.0, Smoke 7.0, Backdraft Conform 7.0
N/A	Burn 2007	Inferno 2007, Flame 2007, Flint 2007, Fire 2007, Smoke 2007, Backdraft Conform 2007
N/A	Burn 2008	Inferno 2008, Flame 2008, Flint 2008, Fire 2008, Smoke 2008, Backdraft Conform 2008
N/A	Burn 2009	Inferno 2009, Flame 2009, Flint 2009, Smoke 2009, Backdraft Conform 2009
N/A	Burn 2009 Extension 1	Inferno 2009 Extension 1, Flame 2009 Extension 1, Flint 2009 Ex-

Burn Client Version	Burn Server Version	Visual Effects and Finishing Version
		tension 1, Smoke 2009 Extension 1, Backdraft Conform 2009 Extension 1
N/A	Burn 2010	Inferno 2010, Flame 2010, Flint 2010, Smoke 2010, Backdraft Conform 2010

NOTE Autodesk Visual Effects and Finishing applications that are compatible with Burn 2010 are 64-bit applications. Jobs from these applications can be very memory-intensive and may cause a render node not equipped with sufficient memory to fail. See Assessing Compatibility between Jobs and Render Nodes on page 62.

Managing Multiple Burn Servers on a Render Node

You can have multiple versions of the Burn server installed on a render node to handle jobs from different Burn clients. For example, you can run the Burn 1.6 and Burn 2010 servers to allow the same render node to handle jobs from the Burn 1.6 client used by Flame 9.0 and Smoke 6.5, as well as jobs from other Autodesk applications that use Burn 2010.

Use the *select_burn* script to help you manage the Burn servers running on a render node. You can use this script to do the following:

- List the Burn server versions installed on a render node.
- Select the version of a Burn server responsible for rendering jobs for a particular version of Burn.
- Test the configuration of a Burn server to make sure it is working.

Use the following procedures to list the Burn server versions installed on a render node and set the type of Burn jobs that each server handles.

To list the Burn server versions installed on a render node:

1. On the render node, log in as root, and then open a terminal.
2. Navigate to the *bin* directory of the latest Burn version. Type:
   `cd /usr/discreet/burn_2010/bin`

3 Run the *select_burn* script in the directory to list the Burn server versions that are installed on the render node. Type:

./select_burn -l

All versions of the Burn server installed on the render node appear, showing their current status. Burn servers listed as *Enabled* process jobs of their type. If necessary, continue to the next procedure to select a specific version of the Burn server responsible for processing matching its type on the render node.

TIP Type **./select_burn -h** to learn more about the options and usage of this script.

To select the version of the Burn server used for rendering jobs on a render node:

1 On the render node, log in as root, and then open a terminal, if necessary.

2 Navigate to the *bin* directory of the latest Burn server. Type:

cd /usr/discreet/burn_2010/bin

3 Run the *select_burn* script to enable a particular version of the Burn server on the render node by typing the following:

./select_burn -c <job type, server version>

You must specify the type of Burn job and the job's corresponding server version to enable the correct server. For example, to enable the Burn 2008 SP1 server on a render node to handle Burn 2008 jobs, type:

./select_burn -c 2008,2008.SP1

The Burn server version is enabled.

4 To ensure that the Burn server version is enabled properly, review the status of all Burn servers installed on the render node. Type:

./select_burn -l

The status of the Burn server listed in step 3 now appears as *Enabled*.

Synchronizing a Burn 1.x Client with the Autodesk Application

Autodesk Visual Effects and Finishing applications that used Burn 1.x included a daemon called the Burn client. The Burn client linked the Autodesk

application with the Backburner Manager. Because a Burn 1.x client can only submit rendering jobs to its version of Burn, you must manually change the Burn client if you run different versions of Inferno, Flame, Flint, or Smoke on the same workstation.

If the workstations in your facility are running multiple versions of Autodesk applications, check the following list of versions that included the Burn client with Burn.

- Inferno 5.3.2 to Inferno 6.2
- Flame/Flint 8.3.2 to Flame/Flint 9.2
- Fire/Smoke 6.5 to Fire/Smoke 6.7

If a workstation in your facility is running one of the above applications, do the following procedure to synchronize the Burn client version with the application. Otherwise, you will be unable to submit Burn jobs from these applications.

To synchronize the Burn client version with its application:

1. If necessary, exit the Autodesk application and log in as root on the workstation.
2. Stop the Burn client currently running. Type:

   `/etc/init.d/burnclient stop`
3. Switch to the *burnclient* directory by typing:

   `cd /usr/discreet`
4. View the contents of this directory. Type:

   `ls`

   This directory contains the following:

   - A sub-directory for the current version of the Burn client.
   - Sub-directories for earlier versions of the Burn client, named *burnclient.previous*. *<number>*. For example, */usr/discreet/burnclient.previous.1* contains the last installed version of the Burn client.
   - A symbolic link called *burnclient* that points to the directory of the current version of the Burn client.

5 Remove the existing symbolic link to the current version of the Burn client. Type:

```
rm burnclient
```

6 Create a new link to the Burn client version that you want to run by typing:

```
ls -s <target directory> burnclient
```

where `<target directory>` is the name of the sub-directory containing the version of the Burn client you want to run.

7 Start the previous version of Burn client. Type:

```
/etc/init.d/burnclient start
```

The Burn client is started. You can now use a previous version of Burn with the Autodesk application.

NOTE Starting a Burn client with the **-C** option cleans the *burnclient* directory of completed Batch setups or timeline clips.

Submitting Jobs to the Distributed Queueing System

Burn provides background rendering on the Distributed Queueing System for Inferno, Flame, Flint, Smoke, and Backdraft Conform.

When the application is properly configured for Burn, a Burn button appears in the clip library or in various application modules. Click this button to submit a setup, timeline, or clip as a background rendering job. Refer to the application User Guide for details.

NOTE The Burn button appears when Burn-related keywords are enabled in an application's *init.cfg* file. See Configuring Burn-related Keywords for the Application on page 11. However, this button's appearance does not mean that the Distributed Queueing System is operational. Always check the status of the network using the Backburner Monitor or Web Monitor before submitting jobs to Burn. See Monitoring and Managing Burn Rendering Jobs on page 43.

Monitoring and Managing Burn Rendering Jobs

During an active render session, you can monitor and manage jobs on the Distributed Queueing System using the following:

- The Background I/O window in your Autodesk Visual Effects and Finishing application. See your application User Guide.
- The browser-based Backburner Web Monitor. See Managing Jobs Using the Backburner Web Monitor on page 43.
- The Windows-based Backburner Monitor. See Managing Jobs Using Backburner Monitor on page 45.

Managing Jobs Using the Backburner Web Monitor

Use the Backburner Web Monitor to monitor and manage your Burn jobs through a Web browser. You can use the Backburner Web Monitor to do the following:

- View all jobs on the Distributed Queueing System, including jobs for Burn and other Autodesk applications like Autodesk® Lustre® or Autodesk® 3ds Max®.
- Manage jobs on the network by stopping, restarting, archiving, and deleting jobs.
- Manage network resources by stopping, restarting, and grouping render nodes as well as scheduling render node availability.

The following procedure shows you how to start the Backburner Web Monitor, view jobs on the network, and see their details. See the *Autodesk Backburner User Guide*.

To view jobs using the Backburner Web Monitor:

1. Open a Web browser and enter the following URL in the address line:

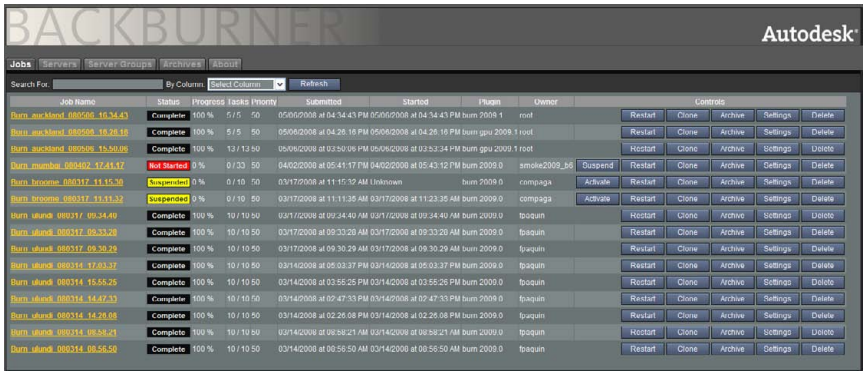
   `http://``<machine_name>`**`/backburner`**

   where `<machine_name>` is the name of the workstation running the Backburner Web Server.

2. If prompted, enter your Backburner Web Monitor username and password.

NOTE Backburner Web Monitor usernames and passwords may be required to access the Distributed Queueing System in your facility. Consult your network administrator for access to the Backburner Web Monitor.

The Backburner Web Monitor appears in the browser, showing all jobs on the rendering network.

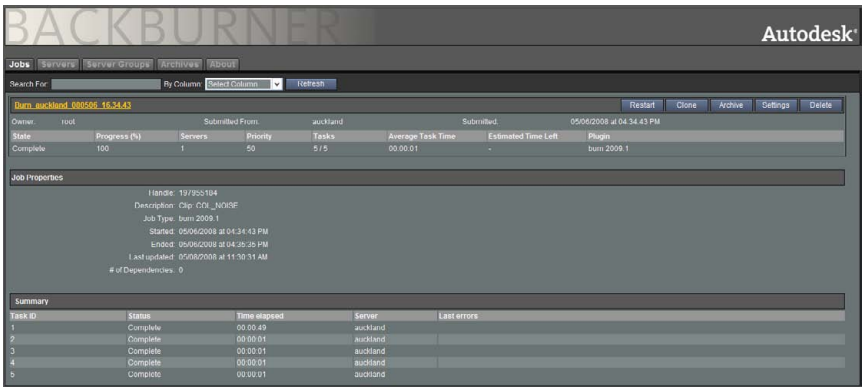


NOTE If the Backburner Web Monitor does not appear, consult your network administrator.

3 Optional: To view the details for a Burn job, click its name in the Backburner Web Monitor.

Details for the job appear in the Web Monitor, including the following information:

- The owner of the job
- The priority assigned to the job
- The number of individual rendering tasks required to render the entire Burn job
- The number of render nodes (servers) used to render the Burn job
- The Burn plug-in used to render the job
- Any errors or messages flagged during the processing of the job



Managing Jobs Using Backburner Monitor

If you are using a Windows workstation where Backburner is installed, use the Backburner Monitor to monitor and manage your Burn jobs. You can use the Backburner Monitor to do the following:

- View all jobs on the Distributed Queueing System, including jobs for Burn and other Autodesk applications like Autodesk Lustre or Autodesk 3ds Max.
- Manage jobs on the network by stopping, restarting, archiving, and deleting jobs.
- Manage network resources by stopping, restarting, and grouping render nodes as well as scheduling render node availability.

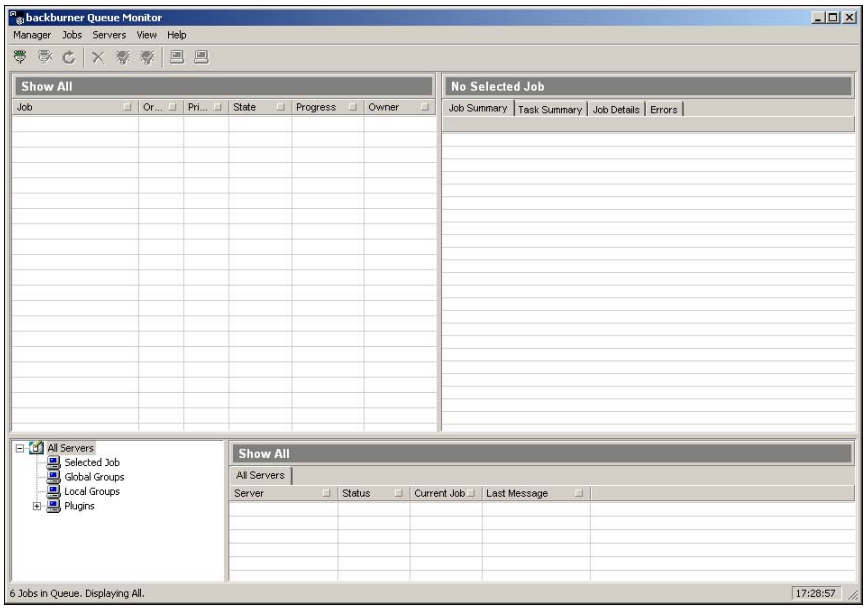
The following procedure shows you how to start the Backburner Monitor and view jobs on the network. See the *Autodesk Backburner User Guide*.

NOTE You can access a Windows-based Backburner Monitor from a Linux workstation using the open-source *rdesktop* application. This application is a remote desktop protocol client that you can use to access your Windows desktop from workstations running other operating systems. For more information, see *http://www.rdesktop.org*.

To view jobs using the Backburner Monitor:

1. On the workstation designated as the render monitor, choose Start | Programs | Autodesk | Backburner | monitor.

   The Monitor window appears.



2 Choose Manager | Connect or click Connect in the toolbar to connect the Monitor to a Backburner Manager.

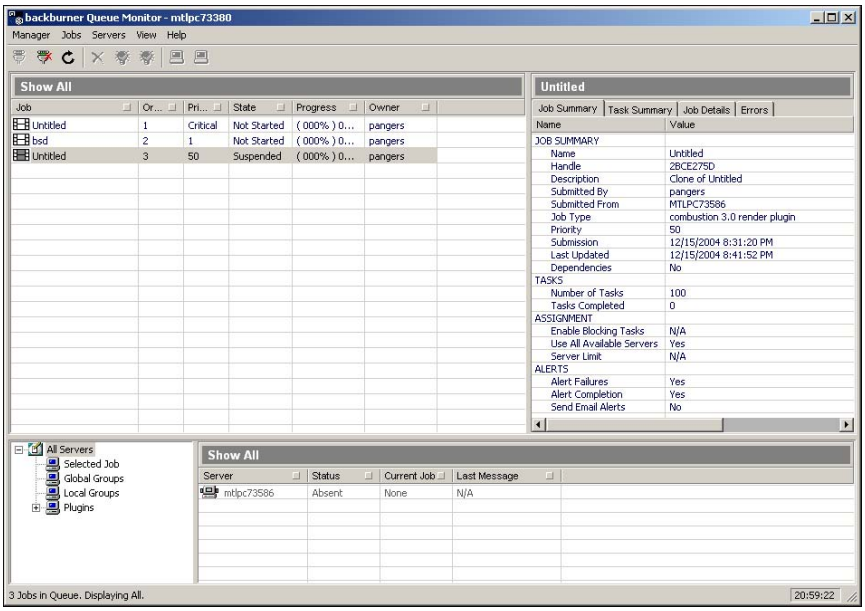
The Connect to Manager dialog appears.

3 If Automatic Search is enabled, disable it, unless the following situations apply:

- You are unsure of the Backburner Manager to which you should connect.
- The Backburner Manager that you want to connect to is on the same subnet.

Enter the IP address of the Backburner Manager you wish to view, and click OK.

The Backburner Monitor connects to the Backburner Manager and displays the jobs on the rendering network.



Previewing Results

You can preview rendering results for Burn jobs by viewing the *_Burn_* library in the clip library of an Autodesk Visual Effects and Finishing application. As you are previewing your result, the clip may appear semi-rendered. Frames are not rendered sequentially, so it is important to wait until the processing tasks are finished and the clip is completely rendered.

To preview a Burn result:

1. Open the clip library. See the User Guide for your application.
2. Open the *_Burn_* library from the Library box.

   Submitted Burn Jobs appear with a name that identifies the machine, the date and time of the job, as well as the job type. Jobs that require a render node equipped with a GPU contain the string “gpu” in their name.

   The *_input* reel contains jobs and source clips that are sent as input to Burn for rendering over the network. The *_output* reel indicates the rendered result that is copied back to the Linux workstation. A clip labelled PENDING RENDER indicates that the clip is being rendered by Burn.

3 To refresh the view of the rendered result, press the **F** hot key.

Troubleshooting Burn 4

Topics in this chapter:

Overview

Troubleshooting problems with rendering jobs on an Autodesk rendering network can be complex because Burn and Backburner components interact with Windows, and Linux platforms over the network. The following procedure outlines the general workflow for troubleshooting Burn and Backburner problems.

To troubleshoot Burn problems:

1. Shut down and restart all Burn and Backburner components, including the following:
   - All Autodesk Visual Effects and Finishing applications that submit jobs to Burn. See the User Guide for your application.

- The Backburner Manager. See Starting Backburner Manager and Backburner Server on page 34.
- The Backburner Server running on each render node rendering Burn jobs. See Starting Backburner Manager and Backburner Server on page 34.

In some cases, simply restarting these components is enough to fix problems with Burn. Otherwise, use the remaining steps to diagnose and correct the problem.

2. Verify that the problem with Burn is not being caused by a larger issue with the rendering network, such as a lack of network connectivity. See Troubleshooting Problems in the Distributed Queueing System on page 50.
3. Use the *GATHER_BURN_LOGS* script to compile and review the Burn and Backburner logs from render nodes on the network. See Reviewing Burn and Backburner Log Files on page 57.

   Reviewing the Burn and Backburner logs helps you determine the following:

   - The component of the Distributed Queueing System that is failing
   - The conditions under which the component fails
4. If these logs indicate that the problems may be due to rendering network components such as Wire networking, use scripts to troubleshoot the rendering network. See Testing Distributed Queueing System Components on page 59.
5. If these logs indicate that the problems may be due to memory issues on the render nodes, or if you suspect that the render nodes do not meet the graphics card requirements for a certain job type, see Assessing Compatibility between Jobs and Render Nodes on page 62.
6. If neither the logs nor the included scripts help you troubleshoot the problem with the rendering network, contact Autodesk Customer Support.

Troubleshooting Problems in the Distributed Queueing System

Problems with rendering jobs may not be directly related to Burn but to the general setup or operation of the Distributed Queueing System. For example,

problems in any of the following areas may cause Burn to appear to stop working:

- General network connectivity
- Stone and Wire connectivity and access
- Licensing issues

You can use basic tools like *ping* to check that these areas are not causing a problem for Burn. The following general troubleshooting procedure can be used to verify that the rendering network is working properly. More detailed procedures are provided in the following sections.

To verify that the rendering network is working properly:

1. Ensure that Burn network components are communicating with each other. See Ensuring Basic Network Connectivity Using ping on page 51.
2. Check that the mount points to workstations can be seen and accessed by render nodes. See Checking Mount Points on the Distributed Queueing System on page 53.
3. Verify that storage devices connected to workstations are available on the Wire network. See Verifying Stone and Wire Connectivity from the Distributed Queueing System on page 55.
4. Check that the Burn license for the rendering network is valid and is being used. See Checking Burn Licensing for Render Nodes on page 55.

Ensuring Basic Network Connectivity Using ping

The following information and procedure applies to all components of a Distributed Queueing System.

Burn works only when components on an Autodesk rendering network are communicating with each other. For example, if Backburner Manager cannot contact the render nodes over the network, jobs cannot be rendered remotely.

You can use the *ping* command to test connectivity between components on the rendering network. The following guidelines suggest ways to test this connectivity using the *ping* command:

- From the workstation, use *ping* to test communication between the Backburner Manager and each render node on the network.

- From the Backburner Manager workstation, use *ping* to test communication between the workstation and each render node on the network.
- From the render nodes on the network, use *ping* to test communication between the Backburner Manager and the workstations that submit jobs to the rendering network.
- If your network infrastructure supports jumbo packets, check that jumbo packets can be transferred between the workstation and the render nodes. See Ensuring Network Connectivity for Jumbo Frames on page 52.

NOTE Refer to the documentation included with your operating system to learn how to use the *ping* command.

Ensuring Network Connectivity for Jumbo Frames

Jumbo frames increase the efficiency of a rendering network, but must be supported throughout the network infrastructure. Burn problems can be caused by jumbo frames being sent to a switch or network adapter that is misconfigured or cannot handle these frames.

If you know that your network infrastructure supports jumbo frame switching, use the following procedure to test if jumbo frames can be sent between the workstations and render nodes.

To test network connectivity using jumbo frames:

1. On a workstation or render node, open a terminal.
2. Run *ping* using the **-s** option to set the packet size used for network communications. Type:

   ping -s 50000 <hostname>

   where <hostname> is the hostname or IP address of the workstation or render node you are trying to reach.
3. If *ping* fails with the command in step 2, ensure basic network connectivity between the same two network components by typing:

   ping <hostname>

   where <hostname> is the hostname or IP address of the workstation or render node you are trying to reach.

   If step 3 works while step 2 failed, recheck the configuration for your network adapter and/or switch to ensure the following:

   - Jumbo frames are supported by both network components.

- Both the network adapter and switch are properly configured for jumbo frame support.

The network infrastructure in your facility may not support jumbo frame switching. Consult your network administrator if you are unsure if your network supports jumbo frames.

Checking Mount Points on the Distributed Queueing System

The following information and procedure applies to workstations and render nodes on a Distributed Queueing System.

Certain directories on workstations, such as */usr/discreet/clip*, must be accessible as mount points so material for jobs can be retrieved by render nodes. Do the following procedure to check these mount points and ensure that render nodes can access material.

To check that mount points are accessible to the rendering network:

1. Check that the *amd* automounter and *nfs* filesystem services are running on each workstation submitting jobs to Burn.

   These services must be running on both network components for Burn to work. Otherwise, the render nodes are not able to access job material (such as frames) from the workstation that initially submitted the job.

2. Log in as root to a render node on the Distributed Queueing System and open a terminal, if necessary.

3. Connect to the */usr/discreet/clip* directory on a workstation that submits rendering jobs. Type:

   `cd /hosts/``<hostname>`**`/usr/discreet/clip`**

   where `<hostname>` is the name of the workstation submitting jobs to Burn.

4. If step 3 fails, check that the *amd* automounter and *nfs* filesystem services are set to start automatically on the render node, using the following command:

   `chkconfig --list | grep` `<service>`

   where `<service>` is either **`amd`** or **`nfs`**.

5 If the services in step 4 are not set to start running automatically, configure them to do so by using the following command:

chkconfig <service> **on**

where <service> is either **amd** or **nfs**.

6 Start the services using the following command:

/etc/init.d/<service> **start**

where <service> is either **amd** or **nfs**.

7 Try again to connect to the */usr/discreet/clip* directory, then create an empty file on the workstation that is submitting jobs to the rendering network by typing the following:

touch /hosts/<hostname>**/usr/discreet/clip/export_test**

where <hostname> is the hostname or IP address of the workstation.

If the *export_test* file cannot be created, permission to write to the required directories may not be assigned to the render nodes. See Enabling Export Permissions on page 14.

8 Check the permissions assigned to the file you created in step 6. Type:

ls -al /hosts/<hostname>**/usr/discreet/clip**

where <hostname> is the hostname or IP address of the workstation.

Filesystem details for the *export_test* file you created in step 6 appear. Ensure that the owner of this file is `root` and the group ID is `sys`. Otherwise, check the following in the Exports file of the workstation submitting jobs to the rendering network:

- For all workstations, the following directories are exported: the entire */usr/discreet directory*, or the sub-directories */usr/discreet/clip*, */usr/discreet/user*, */usr/discreet/Burn*, and */usr/discreet/project*.
- For workstations, the entry in the Exports file must contain the `rw,sync,no_root_squash` parameters.

9 If you changed the Exports file in step 7, enable the changes on the workstation by typing:

/usr/etc/exportfs -a

Verifying Stone and Wire Connectivity from the Distributed Queueing System

The following information and procedure applies to render nodes on a Distributed Queueing System.

Render nodes access frames on storage devices attached to the workstation using the Wire network. Do the following procedure to ensure that these storage devices are available to the render node.

To verify Stone and Wire connectivity from a render node:

1. Log in as root to a render node on the rendering network and open a terminal, if necessary.
2. View all storage devices available to the render node using the *sw_framestore_dump* utility. Type:

   `/usr/discreet/sw/tools/sw_framestore_dump`

   All storage devices attached to the Wire network appear.
3. If a storage device does not appear in the list of devices available to the render node, check the following:
   - The workstation is on the same network as the render node.
   - The workstation to which the device is attached is available on the network and can be contacted using the *ping* command.
   - The *probed* daemon running on the workstation is using the same port as the rest of the network. See the entries for *sw_probed* and *sw_probed.cfg* in the *Stone and Wire Filesystem and Networking Guide*.

Checking Burn Licensing for Render Nodes

The following information and procedures apply to all render nodes as well as the workstation or render node being used as the License Server for the Distributed Queueing System.

Render nodes must get Burn licenses from the License Server to process rendering jobs. If the License Server is not working, or if Burn licenses cannot be checked out, jobs are not rendered. Do the following procedure to ensure that licensing for Burn is working on the render nodes.

To check Burn licensing on the License Server workstation or render node:

1. Log in as root to the render node or workstation running the License Server for the rendering network.
2. Check if the License Server is set to start automatically. Type:

   ```
   chkconfig --list | grep license_server
   ```

3. If the License Server is not set to start running automatically, configure it to do so. Type:

   ```
   chkconfig license_server on
   ```

   If the License Server was not started, start it by typing:

   ```
   /etc/init.d/license_server start
   ```

4. View the *boot.log* file for the License Server to ensure that the Burn license is being read by the server. Type:

   ```
   more /usr/discreet/licserv/log/boot.log
   ```

   The *boot.log* file appears, showing the license information for Burn. Check the following information in this file to see if Burn is being licensed properly on the network:

   - The versions of Burn that are licensed for the network
   - The start and expiry dates for Burn licenses
   - The workstation or render node running as the License server for the rendering network
   - The availability of each Burn license for render nodes

5. View the *DL_license.dat* file to see the number of Burn licenses purchased. Type:

   ```
   more /usr/discreet/licserv/licenses/DL_license.dat
   ```

   The *DL_license.dat* file appears, showing the number of Burn licenses purchased and available to the network. Check to see if the number of purchased licenses are enough for your network. Otherwise, render nodes on your network may be unable to render Burn jobs due to a lack of licenses.

To check Burn licensing on a render node:

1. Log in as root to a render node that is not running the License Server.

2 Contact the render node or workstation running the License Server using the *ping* command by typing the following:

ping <hostname>

where <hostname> is the hostname or IP address of the render node or workstation running the License Server.

If this step fails, check the network connection between the render node and the render node or workstation running the License Server, then retry the *ping* command. Otherwise, continue to the next step.

3 View the *DL_license.dat* file to check that the render node is licensed for Burn. Type:

more /usr/local/flexlm/licenses/DL_license.dat

The *DL_license.dat* file appears. This file should look similar to the following:

```
SERVER exuma-001
DLHOST01=886C2B75E8E57E4B03D784C3A2100AC0
DAEMON discreet_l discreet_l
USE_SERVER
```

Contact Customer Support if the general format or number of lines in the *DL_license.dat* file for your render node is different than the example above.

4 Repeat steps 1 to 4 for the remaining render nodes on the Distributed Queueing System.

NOTE If the License Server for your network is running on a render node, make sure you perform this procedure on this node as well. Otherwise, this node is able to distribute Burn licenses to other render nodes, but is unable to retrieve a license for itself.

Reviewing Burn and Backburner Log Files

The following information and procedure applies to render nodes on a Distributed Queueing System.

Review logged events in Burn and Backburner to identify the source of problems with the rendering network.

Since a rendering network may include many render nodes, a script called *GATHER_BURN_LOGS* is provided to collate and organize the log files from many render nodes into a single text file.

The script collects the Burn log files from the */usr/discreet/log* directory, and the Backburner logs from the */usr/discreet/backburner/Network* directory of each Burn workstation.

Use the following procedure to troubleshoot your rendering network, using the *GATHER_BURN_LOGS* script.

To review Burn and Backburner logs from your rendering network:

1. If necessary, create a list of render nodes from which Burn and Backburner logs should be collected.
2. On your Autodesk Visual Effects and Finishing workstation, log into the account for your Autodesk application and open a terminal.
3. Navigate to the *bin* directory for your application by typing:

   `cd /usr/discreet/``<product_directory>`**`/bin`**

   where `<product_directory>` is the directory where your application is installed (such as `flame_2010`).

   This directory contains scripts and binary files for the application, including those used for submitting jobs to the rendering network.
4. Run the *GATHER_BURN_LOGS* by typing the following:

   `./GATHER_BURN_LOGS` `[options] <render_node_1 render_node_2 ...>`

   where:

   `<render_node_1 render_node_2 ...>` is the list of render nodes on your network that you created in step 1.

   `[options]` include the following options for this script.

Option	Function
`-h`	Shows the usage for the script.
`-v`	Shows the version of the *GATHER_BURN_LOGS* script.
`-V`	Enables additional verbosity.
`-r` `<version>`	Retrieves logs for the specified version of Burn, such as 2010.

Option	Function
-b <build>	Retrieves logs for the specified build of Burn.
-u <filename>	Writes the collected Burn logs to a single output file, ordered by each log file's timestamp. By default, the output file is named *burn.log* and is created in the */bin* directory of the application.
-a <filename>	Writes the collected Backburner logs to a single output file, ordered by each log file's timestamp. By default, the output file is named *backburner.log* and is created in the */bin* directory of the application.
-e <filename>	Writes the events in the collected Burn logs to a single output file, ordered by each event's timestamp. By default, the output file is named *burnevents.log* and is created in the */bin* directory of the application.
-l <username>	Uses the specified user name to log into the render node. If this option is omitted, you may be prompted to provide a username and password.

The *GATHER_BURN_LOGS* script collects Burn and Backburner logs from each render node listed in the script and collates these logs to a single text file.

NOTE The *GATHER_BURN_LOGS* script uses the *rsh* protocol to access Burn render nodes. To avoid being prompted for login information while running this script, create a user account on the workstation and populate the *.rhosts* file on each render node with the hostname of the workstation followed by the username of this account. You can then use the -l option to run the script as this user.

Testing Distributed Queueing System Components

Reviewing Burn and Backburner log files can help to indicate the source of a problem on the Distributed Queueing System. For example, log messages showing network timeouts for jobs may indicate a problem with the TCP/IP protocol or the Stone and Wire network.

Use the following scripts that are installed with Burn to test rendering network components and identify the source of the problem:

- The *verifySWConn* script is used to test Wire network connectivity in the rendering network. See Testing Stone and Wire Connectivity for Burn on page 60.
- The *verifyBurnServer* script is used to test whether a render node meets minimum hardware requirements for rendering Burn jobs on a network. See Testing Render Node Hardware for Burn on page 61.

Testing Stone and Wire Connectivity for Burn

The following information and procedure applies to workstations and render nodes on a Distributed Queueing System.

Use the *verifySWConn* script to see if rendering problems are due to Stone and Wire. This script can be run from a workstation or from a render node on the rendering network using the following procedure.

To test Stone and Wire connectivity:

1. Log into the workstation or render node as follows:
   - For workstations, log in using the account for your application and open a terminal.
   - For Burn render nodes, log in as root and open a terminal, if necessary.
2. Navigate to the *bin* directory for your application by typing:
   `cd /usr/discreet/``<product_directory>`**`/bin`**
   where `<product_directory>` is one of the following:
   - For workstations, the directory where your application is installed, such as `flame_2010`
   - For Burn render nodes, the directory containing the Burn version used on the rendering network, such as `burn_2010`

   This directory contains Burn-related scripts, including the *verifySWConn* script.
3. Run the *verifySWConn* script by typing the following:
   `./verifySWConn` `<options> <Workstation1 Workstation2 ...>`

where:

<Workstation1 Workstation2 ...> are the workstations or Burn render nodes on which Stone and Wire are to be tested. You must specify at least two workstations or render nodes to be tested.

<options> include the following options for this script.

Option	Function
-v	Enables additional verbosity.
-l <username>	Uses the specified user name to log into the render node. If this option is omitted, you may be prompted to provide a username and password. The username must be valid on the test host.

The *verifySWConn* script runs, testing Stone and Wire functionality between each network component specified in the script.

Testing Render Node Hardware for Burn

Use the *verifyBurnServer* script to see if a Linux system has the hardware requirements to be used as a Burn render node.

The script also checks if a render node has a supported GPU-accelerated graphics card and graphics driver version, necessary for rendering jobs that require a GPU.

To test render node hardware for Burn:

1. Log into the render node as root and open a terminal.
2. Navigate to the *bin* directory for Burn by typing:

   cd /usr/discreet/<burn_version>**/bin**

   where <burn_version> is the directory containing the Burn version used on the rendering network, such as burn_2010.
3. Run *verifyBurnServer* by typing the following:

   ./verifyBurnServer

NOTE This script takes no options.

The *verifyBurnServer* script checks the hardware of the system to ensure it meets the requirements for Burn render nodes, and displays the results.

Assessing Compatibility between Jobs and Render Nodes

Rendering Jobs that Require a GPU

Some of the jobs created in your Visual Effects and Finishing application (for example, floating point jobs, such as unclamped colors in Action, directional RGB blur, radial RGB blur) require a GPU-accelerated graphics card in order to be rendered.

While your workstation is equipped with a GPU-accelerated graphics card, and can render such jobs locally, your rendering network is unable to render these types of jobs if no Burn node is equipped with a GPU.

To see if a render node has the hardware capabilities to process jobs that require a GPU, use the *verifyBurnServer* script, Backburner Monitor, or Backburner Web Monitor.

If you attempt to submit a job that requires a GPU to a rendering network where no render node is equipped with a GPU, one of the following situations occurs:

- If the `BackburnerManagerGroupCapability` keyword in the application's *init.cfg* file is set up correctly, the application does not attempt to submit the job to the rendering network, and an error message is displayed. You must render the respective job locally on the Visual Effects and Finishing workstation.
- If the `BackburnerManagerGroupCapability` keyword is not set up properly, no error message is displayed and the application attemps to send the job to the rendering network. Since no render node can process the job, the job will be stuck in the rendering queue indefinitely. Use Backburner Monitor or the application's Background I/O window to remove the job from the queue, and then set the `BackburnerManagerGroupCapability` keyword properly to reflect the hardware capabilities of your rendering network. See Monitoring and Managing Burn Rendering Jobs on page 43, and Configuring Burn-related Keywords for the Application on page 11.

To avoid further problems, before attempting to submit a job that requires a GPU to your rendering network, make sure at least one of the render nodes is equipped with a GPU, and that the `BackburnerManagerGroupCapability` keyword in the application's

init.cfg file is set up correctly. See Hardware and Software Requirements on page 5, and Configuring Burn-related Keywords for the Application on page 11.

Diagnosing Memory Issues

Inferno 2010, Flame 2010, Flint 2010, Smoke 2010, and Backdraft Conform 2010 are all 64-bit applications, and can thus make full use of up to 16 GB of memory.

As a general rule, render nodes should have the same amount of RAM as the Visual Effects and Finishing workstation you are sending rendering jobs from.

A Burn server running on a render node equipped with less memory than what is installed on your Visual Effects and Finishing workstation, may fail when rendering these jobs due to their higher memory demands. However, do not assume that every problem on render nodes with less memory than your workstation is exclusively caused by memory issues.

This section explains how to diagnose and address problems that are caused by jobs submitted from workstations with more memory than the render node.

If you suspect that a render node has failed due to a Burn job exceeding the Burn nodes memory capacity, use the following procedure to view the Burn and Backburner server logs for memory-related errors.

To check the Burn server log on a render node:

1. If you are running graphics on the render node, log in as root and open a terminal. Otherwise, just log in as root.
2. Navigate to the */usr/discreet/log* directory.

   This directory contains logs of events for the Burn servers installed on the render node. You need to view the log created at the time the server failed.
3. Identify the Burn log file from the time of the Burn server failure using one of the following methods:
   - If the render node has just failed, look for the following file: *burn2010_<render_node_name>_app.log*; for example, *burn2010_burn7_app.log*.
   - If the render node failed previously and was brought back online, type **`ls`** and then look for the file

burn2010_<render_node_name>_app.log.## created around the time of the render node's failure; for example, *burn2010_burn7_app.log.5*.

4. View the contents of the Burn 2010 log file directly in the terminal or by using a text editor.

5. Review the messages in the log file for entries similar to the following:

```
[error] 8192 PPLogger.C:145 01/24/06:17:06:16.998
Cannot load video media in node "clip17" for frame
2

[error] 8192 PPLogger.C:145 01/24/06:17:06:17.210
Out of memory for image buffers in node "clip6"
(76480512 bytes).

Increase your memory token.
```

   These entries may indicate that the render node was experiencing memory problems at the time of failure. Next, check the Backburner Server log file to corroborate these findings.

6. Navigate to the */usr/discreet/backburner/log* directory.

7. Identify the Backburner Server log file *backburnerServer.log* from the time of the Burn server's failure, using the methods listed in step 3.

8. Review the messages in the Backburner Server log file in a text editor, looking for entries similar to the following:

```
[notice] 16387 common_services.cpp:45
01/24/06:17:06:10.069 Launching 'burn'

[error] 16387 common_services.cpp:37
01/24/06:17:06:48.182 Task error: burn application
terminated (Hangup)

[error] 16387 common_services.cpp:37
01/24/06:17:06:48.182 burn application terminated
(Hangup)

[notice] 16387 common_services.cpp:45
01/24/06:17:06:48.524 Application is down
```

   These log entries confirm that a Burn server failure occurred on the render node. Since you know that the Burn 2010 server failed around this time, you can deduce that the memory problem caused the Burn server to fail.

9. Optional: Identify the workstation running the application that submitted the job to Burn, and then look at the Batch setup, timeline segment, or clip to try and determine why the Burn server failed.

Problems that cause the Burn server to fail due to lack of memory on a render Node, usually arise due to:

- The size of images used in a project. For example, projects using higher resolution HD, 2K, and 4K images require more memory to store and render than SD projects.
- The complexity of the effect sent for rendering. For example, a complex Batch setup with many layers and effects requires more memory to render than a simple Batch setup.

Knowing what factors caused the render Node to fail may help you to gauge what jobs your render nodes can handle. It can also give you ideas about how to deal with this problem.

Addressing Memory Issues

If Burn 2010 servers on your render nodes are failing while rendering jobs, increase the amount of RAM set aside for rendering Burn 2010 jobs. You must repeat this procedure on each render node on your network running the Burn 2010 server.

To configure Burn to reserve a set amount of RAM for rendering jobs:

1. Log into the render node as root and open a terminal.
2. Stop the Backburner Server on the render node by typing the following:
   `/etc/init.d/backburner_server stop`
3. Open the */usr/discreet/burn_2010/cfg/init.cfg* file in a text editor.
4. Uncomment the *MemoryApplication* keyword.
   This keyword sets the amount of RAM in megabytes (MB) to be reserved for Burn jobs. This keyword is disabled by default so Burn can dynamically adjust the amount of RAM used for each rendering job based on the resolution of the project. When you enable this keyword, Burn reserves the same amount of memory for each rendering job regardless of the project's resolution.
5. If necessary, change the value for the *MemoryApplication* keyword to set the amount of RAM (in MB) to be reserved for each Burn job up to 1400 (about 1.4 GB).
   For example, to reserve 1 GB of RAM for Burn jobs, type the following:

```
MemoryApplication 1024
```

NOTE Setting the *MemoryApplication* keyword so that the (total render node memory) - (value of *MemoryApplication*) is less than 2600 MB may adversely affect the stability of the render node.

6 Save and close the *init.cfg* file.

7 Restart the Backburner Server on the render node by typing:

```
/etc/init.d/backburner_server start
```

Burn is restarted and reserves the amount of RAM set in the *MemoryApplication* keyword.

8 Optional: Implement the following guidelines for rendering Burn jobs:

- If you know that the size of images in your projects may cause render node failure, enforce guidelines about what can and cannot be sent to the Burn render nodes. For example, if you know that 2K and 4K images with Batch setups exceeding six layers may cause the render nodes to fail, ensure these setups are not sent to Burn.
- If you know that the complexity of the effects sent for rendering may cause render node failure, simplify effects by creating multiple Batch setups or by rendering memory-intensive effects locally. For example, if you know that complex Batch setups with multiple logic ops and colour correction may cause render nodes to fail, render these locally instead.

NOTE Although these guidelines are not mandatory, following them may help increase the success rate while rendering Burn jobs on render nodes with limited memory resources.

If, after following these guidelines, your render nodes still fail because of low memory, you might consider adding memory to the render nodes. Matching the amount of memory on the render nodes with the amount of memory found on your Visual Effects and Finishing workstation is the most effective solution to memory issues.

Installing Linux on a Burn Render Node

5

Topics in this chapter:

Do You Need to Install Linux?

The current version of Autodesk Burn runs under the custom Autodesk distribution of Red Hat Enterprise Linux Workstation 4.0 Update 3, or CentOS 4.6. Burn supports only 64-bit distributions of Linux.

Burn render nodes purchased from Autodesk ship with the custom Autodesk distribution of Red Hat Enterprise Linux Workstation 4, Update 3 already installed. Thus, you should only need to install Linux in the following situations:

- Your render node is running an earlier version of Red Hat Enterprise Linux Workstation or CentOS. In this case, you cannot upgrade to the current version; you must perform a fresh install.
- You are setting up a Burn render node that was not purchased from Autodesk. CentOS is the required operating system for render nodes not purchased from Autodesk. Make sure your render node hardware is supported by CentOS.
- Your render node experienced an unrecoverable hard disk failure.

NOTE If you must replace the hard disk, you need to update your Burn license. Contact Customer Support for assistance.

Preparing the Render Node

Before beginning the installation process, ensure the Burn render node is in the following state:

- Mouse, keyboard, and graphics monitor are connected and powered on.
- The CD/DVD drive is set as the primary boot device in the render node BIOS.
- There are no floppy disks, CDs, or DVDs in the drives of the render node.

Installation Workflow

This section describes the main steps involved in installing the custom Autodesk distribution of Red Hat Enterprise Linux Workstation 4 Update 3, or your distribution of CentOS 4.6 on a Burn render node. The remainder of the chapter provides details on each of these steps.

To install Linux on a Burn render node:

1. Locate the installation media for your Linux distribution.

Render nodes purchased from Autodesk ship with the Autodesk distribution of Red Hat Enterprise Linux Workstation 4, Update 3, on DVD. If you did not purchase your render node from Autodesk, you must get your own 64-bit Linux distribution of CentOS 4.6.

2 If installing CentOS, you need to customize your Linux distribution with the Autodesk kickstart file. The file is available in any of the following locations:

- The DVD of the Autodesk distribution of Red Hat Enterprise Linux Workstation 4, update 3.
- Your *Burn* installation package, from the Autodesk application DVD or from a downloaded *tar* file.
- The Autodesk FTP site. For information on how to download it, contact Customer Support.

You must copy the kickstart file to either the first CD/DVD of your CentOS distribution (recommended), or to a floppy disk (if you do not have a CD or DVD burner). See Preparing the Kickstart File on CD/DVD on page 71 or Preparing the Kickstart File on a Floppy Disk on page 72.

NOTE For help understanding the kickstart file, see Understanding the Autodesk Kickstart File on page 70.

3 Install Linux.

- If you are reinstalling the custom Autodesk distribution of Red Hat Enterprise Linux, or if you are installing CentOS using the Autodesk kickstart file on CD or DVD, see Installing Linux Using the Kickstart file on CD/DVD on page 72.
- If you are installing CentOS using the Autodesk kickstart file on a floppy disk, see Installing CentOS Using the Kickstart File on a Floppy Disk on page 74.

4 Perform the following post-installation tasks:

- Change the default root password to secure the system. The provided Autodesk kickstart file sets the password for the root account on your render node to *password*.
- Configure the network settings for your render node to match the ones used in your facility.
- Configure the time zone for your geographic location. The node is set to North American Eastern Standard Time (EST) by default.

NOTE If necessary, consult the Red Hat or CentOS online documentation to configure these settings.

- If you did not receive your render node from Autodesk, check for firmware or driver updates for your hardware after CentOS is installed. Refer to the CentOS and/or hardware manufacturer Web sites to ensure you have the correct firmware and drivers.

5 Install the Discreet Kernel Utilities (DKU).

DKU installation is mandatory whether you are using Red Hat Enterprise Linux or CentOS. See Installing the DKU on Render Nodes

NOTE On Burn nodes equipped with the NVIDIA® Quadro® FX 3700 graphics board, you need to boot Linux in text mode to install the DKU before being able to use Linux in graphic mode. See DKU Installation in Text Mode on GPU Burn Nodes on page 76.

6 Install Burn on the render node. See Installing Burn on the Render Nodes on page 22.

Understanding the Autodesk Kickstart File

A kickstart file is a text file that contains instructions for installing a Linux distribution. It is used to automatically configure the render node and install the packages required for Burn, some of which are not installed as part of a general installation

TIP The kickstart file can be used to automate the Linux installation process for multiple render nodes.

The *ks_burn.cfg* kickstart file is already included in the custom Autodesk distribution of Red Hat Enterprise Linux.

If you are using CentOS, you must manually add the file to your CentOS distribution CDs/DVDs. If you do not have a CD or DVD burner, the file can be made available on a floppy disk.

Preparing the Kickstart File on CD/DVD

This section describes how to create a new CD1 or DVD for your CentOS distribution so that it contains the Autodesk kickstart file.

To copy the Autodesk kickstart file to the first CD/DVD of your CentOS distribution:

1. On a computer running Linux and with a CD or DVD burner, log in as root.
2. Insert the CD1 or DVD of your CentOS distribution into the drive. You do not need to mount it at this time.
3. In a terminal, extract an ISO image of the disc by typing:

   dd if=/dev/<CD/DVD device>**of=/**<destination path for the extracted ISO image>

   For example:

   dd if=/dev/cdrom of=/tmp/CentOS4.6.iso

   Depending on the speed of your disc drive, this command may take several minutes to complete.
4. Eject the disc.
5. Access the *dist/kickstart* subdirectory of your Burn installation package (from the Autodesk application DVD or from a downloaded *tar* file).

   The directory contains a kickstart file, *RHEL40_CentOS4_kickstart.cfg*, as well as a script that adds the kickstart file to an ISO image.
6. Run the *build_kickstart_cd* script to add the kickstart file to the ISO image of your CentOS distribution CD1 or DVD:

   ./build_kickstart_cd RHEL40_CentOS4_kickstart.cfg <original ISO image name> <new ISO image name>

   For example, if the ISO image you created is called */tmp/CentOS4.6.iso* and you want the new ISO image to be called */tmp/CentOS4.6_KS.iso*, type:

   ./build_kickstart_cd RHEL40_CentOS4_kickstart.cfg /tmp/CentOS4.6.iso /tmp/CentOS4.6_KS.iso
7. Once the new ISO image of the CentOS distribution CD1 or DVD is created, burn it to a blank disc using a tool such as **cdrecord**.

NOTE Type **man cdrecord** for information about this utility.

The new disc that you burn contains the appropriate Autodesk kickstart file for your distribution of CentOS, and replaces the first CD or DVD in that distribution.

Preparing the Kickstart File on a Floppy Disk

This section describes how to copy the kickstart file to a floppy disk.

This floppy disk is used by the CentOS distribution discs when the workstation boots in Installation mode.

To prepare the kickstart floppy disk:

1. On any computer running Linux, log in as root and load a standard formatted 3½" floppy disk.
2. Open a terminal and format the floppy disk:
   ```
   fdfloppy
   ```
3. Once the floppy disk is formatted, create a Linux filesystem on it. Type:
   ```
   mkfs.ext2 /dev/fd0
   ```
4. Mount the floppy. Type:
   ```
   mount /mnt/floppy
   ```
5. Access the *dist/kickstart* subdirectory of your Burn installation package (from the Autodesk application DVD or from a downloaded *tar* file).
   The directory contains the *RHEL40_CentOS4_kickstart.cfg* kickstart file.
6. Copy the kickstart file to the floppy disk as *ks.cfg*. For example:
   ```
   cp RHEL40_CentOS4_kickstart.cfg /mnt/floppy/ks.cfg
   ```

NOTE If the kickstart file on the floppy disk is not named *ks.cfg*, the CentOS installation will not use it.

Installing Linux Using the Kickstart file on CD/DVD

Use the following procedure to install Linux on a Burn render node using the Autodesk kickstart file on the first CD, or the DVD, of your Linux distribution.

This procedure also applies if you are reinstalling the custom Autodesk distribution of Red Hat Enterprise Linux.

To install Linux using the Autodesk kickstart file on the Linux CD/DVD:

1. Insert the first CD, or the DVD, of the Linux distribution. This is the CD or DVD that contains the Autodesk kickstart file.
2. Restart the system.
3. At the boot prompt.

If you are using:	Type:
The Autodesk distribution of Red Hat Enterprise Linux Workstation 4 Update 3	`burn`
CentOS with the Autodesk kickstart file	`linux ks=cdrom`

   This launches the installation procedure. The remaining portion of the installation is automated.

   NOTE You cannot install Linux with the kickstart file from the graphical installation mode. Do not enter this mode if prompted. By default, the Autodesk kickstart file forces text-mode installation in order to avoid possible display detection problems.

4. If you replaced the render node's system disk, the installation procedure may ask you to initialize this disk. Follow the prompts to initialize the system disk, if necessary.
5. Optional: Insert the remaining Linux distribution CDs when prompted.
6. When prompted with "Congratulations," eject the disk from the drive.

   Linux is installed on the render node.
7. Click Exit to reboot the render node.

   When the reboot is complete, perform the post-installation tasks in the installation procedure. See Installation Workflow on page 68.

Installing CentOS Using the Kickstart File on a Floppy Disk

Use the following procedure to install CentOS on a Burn render node using the Autodesk kickstart file on a floppy disk.

To install Linux using the Autodesk kickstart file on a floppy disk:

1. Insert the first CD, or the DVD, of the CentOS distribution.
2. Restart the system.
3. When the Linux installation prompt appears, insert the kickstart floppy disk into the floppy drive.
4. At the prompt, type:

   `linux ks=floppy`

   This launches the installation procedure.

   NOTE You cannot install Linux with the kickstart file from the graphical installation mode. Do not enter this mode if prompted. By default, the Autodesk kickstart file forces text-mode installation, in order to avoid possible display detection problems.

5. When prompted for the installation method, select Local CD-ROM.

   WARNING If you are not prompted for the installation method, or a dialog other than the Time Zone selection appears, the kickstart file is not being read. Stop the installation, verify the integrity of the kickstart floppy disk, check that the actual filename is *ks.cfg*, and that it is located in the root directory. Make sure the disk and file can be read on another system or workstation, and then start again by booting the system.

   The installation procedure is launched. The remaining portion of the installation is automatic.
6. If you replaced the render node's system disk, the installation procedure may ask you to initialize this disk. Follow the prompts to initialize the system disk, if necessary.
7. Insert the remaining CentOS distribution CDs as prompted.
8. When prompted with "Congratulations," eject the disks from the CD-ROM/DVD-ROM and floppy drives.

CentOS is installed on the render node.

9 Click Exit to reboot the render node.

When the reboot is complete, perform the post-installation tasks in the installation procedure. See Installation Workflow on page 68.

Installing the DKU on Render Nodes

Before installing Burn on the Render Node, perform the following procedure to install the Discreet Kernel Utilities (DKU).

The DKU is available on the application DVD, or as a download from Autodesk. For the download links, consult the release announcement or contact Customer Support

NOTE On Burn nodes equipped with the NVIDIA® Quadro® FX 3700 graphics board, you need to boot Linux in text mode to install the DKU.

To install the DKU on a render node:

1 Consult the application Release Notes for information on the DKU version required for this release.

2 Optional: Check the currently installed DKU version, if applicable. As root, open a terminal and type:

```
head -n1 /etc/DKUversion
```

If the DKU version output by the command does not match the version required for the current release, perform the remaining steps in this procedure.

3 Access the DKU installation directory.

- If you downloaded the DKU, unpack the tar file to a temporary directory, by typing:
  tar -zxvf DKU_<DKU_version_number>**.tar.gz**
- If you are installing the DKU from the application DVD, mount the DVD, and go to the DKU directory by typing:
  cd /mnt/cdrom/DKU_<DKU_version_number>

4 In DKU installation directory, launch the DKU installation script:

```
./INSTALL_DKU
```

NOTE The DKU installs all necessary drivers for HP ProLiant DL160 render nodes using Red Hat Enterprise Linux WS4 Update 3. For any other hardware platform or Linux version, the DKU only installs a generic configuration, and the NVIDIA driver needed by Burn GPU.

5 When the DKU installation completes, reboot the system. Type:

```
reboot
```

DKU Installation in Text Mode on GPU Burn Nodes

Perform this procedure to boot Linux into text mode in order to install the DKU on your render node.

The DKU installs and configures the driver for the new graphics card, and makes the appropriate changes in the */etc/X11/xorg.conf* file.

To boot Linux in text mode and install the DKU:

1 Once Linux is installed, reboot your Burn node.

The Linux boot loader appears.

2 Press any key to stop the normal boot process and enter the GRUB boot menu.

3 Using your keyboard up and down arrows, select your Linux kernel (usually the first one in the list), and press **A** to edit its boot parameters.

4 Add a *3* to the end of the kernel line to force Linux to boot into text mode.

The modified line should be similar to the following:

```
ro root=LABEL=/ hdc=ide-scsi selinux=0
bigphysarea=73729 powernow-k8.disable=1 3
```

5 Press ENTER to accept changes and boot the kernel.

Linux boots into text-only mode.

6 Log in as root and follow the normal DKU installation procedure. See Installing the DKU on Render Nodes.

HP ProLiant DL160 G5 Render Node

6

Topics in this chapter:

Overview

The HP ProLiant DL160 Generation 5 (G5) is the recommended render node for Autodesk Burn configurations, and the only supported hardware platform for running both Autodesk Burn and Autodesk Incinerator on the same render node.

HP DL160 G5 render nodes equipped with the NVIDIA Quadro FX3700 graphics board represent the recommended configuration for GPU-enabled Burn.

Refer to the following table for the most important technical specifications of the HP DL160 Burn render node.

Component	Description
CPU	Dual Quad-Core Intel XEON E5472 CPU at 3.0GHz
Front-Side Bus	1600MHz
Memory	■ Minimum: 8GB ■ Recommended: 16GB
Hard Drive	160GB SATA drive at 7500 RPM
Optical Drive	DVD-ROM
PCIe Expansion slot 1 (full length)	NVIDIA Quadro FX3700 graphics board, mandatory for GPU-enabled Burn render nodes
PCIe Expansion slot 2 (half length)	Optional QLogic™ InfiniBand 9000 board

Power and Air Conditioning Requirements

The following table summarizes the peak (at startup) power consumed by the HP DL160 G5 system and the heat it generates under the maximum processing load. For detailed specifications, including noise output, see the documentation provided by the manufacturer.

Startup Amps (120V / 240V)	Max. Amps (120V / 240V)	Watts	Heat (BTUs)
7.3 / 3.5	5.7 / 3.0	650	2220

You must be able to meet the startup power requirement and have a climate control system with the capacity to maintain the temperature of these components under the maximum processing load.

You should also consider the following safety guidelines for your hardware configuration:

- Place all components in an air-conditioned environment. All hardware components generate heat and must be kept cool.

- Make sure the rack on which hardware components are mounted is open or well ventilated. Follow the ventilation specifications that apply to your system.
- Keep all hardware components in a clean, dust-free location.
- Minimize vibration and humidity.
- Do not block the vents on the component housing.
- Do not drape anything, such as a jacket or a blanket, over hardware components.
- Minimize electromagnetic noise by separating digital data and power cables from analog audio cables and running them in different cable ducts.

When installing any hardware equipment, take the following precautions to prevent damage to sensitive components from static discharge:

- Make sure power is turned off on the component you are working on. It is a good idea to unplug components until all other connections are configured.
- Always wear a grounded static wrist strap. Attach the strap's alligator clip to any grounded metal surface on the component's chassis that you are working on. Place the wristband around your wrist.
- Do not handle any components unnecessarily, particularly boards and cards that slide in and out of slots on their parent hardware components.

HP DL160 Wiring Diagram

Refer to the following wiring diagram when setting up an HP ProLiant DL160 G5 system as a Burn render node.

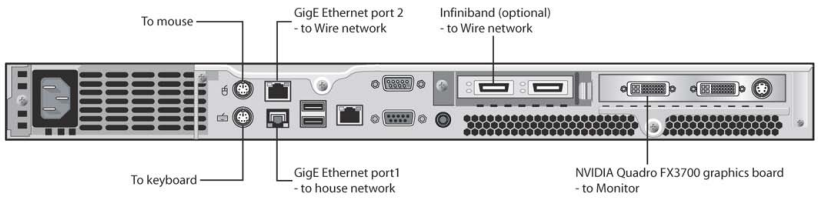
The diagram depicts the HP DL160 with an NVIDIA Quadro FX3700 graphics board and a QLogic InfiniBand 9000 board installed.

The NVIDIA graphics board is required only for GPU-enabled Burn nodes. If you configure an HP DL160 Burn node without the NVIDIA graphics board, connect the monitor to the node’s on-board graphics port.

The Infiniband board is optional in all Burn configurations. If you configure an HP DL160 Burn node without the Infiniband board, connect the render node to the Wire network using the bottom on-board GigE ethernet port (port

one). The Infiniband board is required if you want to run both Burn and Incinerator on the same HP DL160 G5 render node.

HP ProLiant DL160 G5 Burn Render Node



NOTE You can connect the keyboard, mouse, and monitor directly to the render node, or you can use KVM switches to connect multiple nodes to the same keyboard, mouse, and monitor.

HP DL160 BIOS

In most cases, hardware integration and application installation are done on delivery by an authorized technician, so you should not have to verify or upgrade the BIOS. But, if you are upgrading your the application yourself, perform the following procedures.

System BIOS

System configuration is done prior to delivery by an authorized technician. The procedures in this section may not be necessary, and are provided here for informational purposes only.

Verifying Your System BIOS

The system BIOS on your HP DL160 must correspond to the certified version required. Check the application Release Notes for information on the certified BIOS version for your release.

If the BIOS version on your system does not correspond to the one in the Release Notes, you must update to the certified version.

Updating Your System BIOS

The BIOS version installed on your system appears on the screen while booting the Burn render node. The following procedure describes how to update the Burn render node to the certified BIOS version required.

To update the BIOS on your workstation:

1. Insert the application DVD in the drive of the workstation.
2. Open a terminal, log in as root and mount the DVD.
3. Go to the *DKU_<version_number>/Utils/BIOS* directory on the DVD, where <version_number> represents the version of the DKU. For example:

   cd /media/cdrom/DKU_4.0.0/Utils/BIOS

   TIP If the DKU has already been installed on the workstation, you can access its contents in the */usr/discreet/DKU* directory, without having to use the application DVD.

4. Type:

   cd HPproliantDL160_G5

5. Open the *README.txt* file and follow the instructions.
6. Verify system BIOS settings. See HP DL160 BIOS on page 80.
7. Select File | Save Changes and Exit.

HP DL160 BIOS Settings

Before installing the Linux operating system on a new HP DL160 render node, use the following table to validate your BIOS settings.

The table lists the proper Autodesk certified BIOS settings. Items not listed are set to their default factory settings.

BIOS Menu	Submenu	Item	Value
Advanced	CPU Configuration	C1E Support	Enabled

BIOS Menu	Submenu	Item	Value
		Hardware Prefetcher	Enabled
		Adjacent Cache Line Prefetch	Enabled
		Intel Virtualization Tech	Disabled
		Intel SpeedStep Tech	Enabled
Advanced	ACPI Configuration	Chipset ACPI Configuration, High Precision Event Timer	Enabled
	PCI Bus Configuration	PCI Bus Compatibility Mode	GEN 2
		Video Card Support	Enabled
		Snoop Filter	Enabled
Boot	Boot Device Priority	1st	DVD
		2nd	SATA
		3rd	Embedded NIC 1

Operating System Requirements

HP DL160 G5 Burn render nodes purchased from Autodesk come with the Red Hat Enterprise Linux 4 Update 3 operating system pr-installed.

On render nodes not purchased from Autodesk, you need to install CentOS 4.6.

NOTE The custom Autodesk distribution of Red Hat Enterprise Linux 4 Update 3 is the only supported operating system for running Autodesk Burn and Autodesk Incinerator on the same render node.

The Discreet Kernel Utilities (DKU) must be installed on the render node after the operating system is installed. See Installing Linux on a Burn Render Node on page 67.

NOTE If the `acpi=off` parameter is present in the Linux boot configuration file */etc/grub.conf*, it needs to be removed for the render node to boot properly.

To remove the acpi=off boot parameter:

1. Log into the render node as root and open a terminal.
2. Open the */etc/grub.conf* file in a text editor
3. Locate the `acpi=off` parameter in the kernel line, and delete the parameter from the line.
4. Save and close the *grub.conf* file.

Index

T

W