

Installation Guide

Autodesk® Backburner™ 2011

© 2010 Autodesk, Inc. All rights reserved. Except as otherwise permitted by Autodesk, Inc., this publication, or parts thereof, may not be reproduced in any form, by any method, for any purpose.

Certain materials included in this publication are reprinted with the permission of the copyright holder.

Portions relating to MD5 Copyright (C) 1991-2, RSA Data Security, Inc. Created 1991. All rights reserved. License to copy and use this software is granted provided that it is identified as the "RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing this software or this function. License is also granted to make and use derivative works provided that such works are identified as "derived from the RSA Data Security, Inc. MD5 Message-Digest Algorithm" in all material mentioning or referencing the derived work. RSA Data Security, Inc. makes no representations concerning either the merchantability of this software or the suitability of this software for any particular purpose. It is provided "as is" without express or implied warranty of any kind. These notices must be retained in any copies of any part of this documentation and/or software.

Trademarks

The following are registered trademarks or trademarks of Autodesk, Inc., and/or its subsidiaries and/or affiliates in the USA and other countries: 3DEC (design/logo), 3December, 3December.com, 3ds Max, Algor, Alias, Alias (swirl design/logo), AliasStudio, AliasWavefront (design/logo), ATC, AUGI, AutoCAD, AutoCAD Learning Assistance, AutoCAD LT, AutoCAD Simulator, AutoCAD SQL Extension, AutoCAD SQL Interface, Autodesk, Autodesk Envision, Autodesk Intent, Autodesk Inventor, Autodesk Map, Autodesk MapGuide, Autodesk Streamline, AutoLISP, AutoSnap, AutoSketch, AutoTrack, Backburner, Backdraft, Built with ObjectARX (logo), Burn, Buzzsaw, CAICE, Civil 3D, Cleaner, Cleaner Central, ClearScale, Colour Warper, Combustion, Communication Specification, Constructware, Content Explorer, Dancing Baby (image), DesignCenter, Design Doctor, Designer's Toolkit, DesignKids, DesignProf, DesignServer, DesignStudio, Design Web Format, Discreet, DWF, DWG, DWG (logo), DWG Extreme, DWG TrueConvert, DWG TrueView, DXF, Ecotect, Exposure, Extending the Design Team, Face Robot, FBX, Fempro, Fire, Flame, Flare, Flint, FMDesktop, Freewheel, GDx Driver, Green Building Studio, Heads-up Design, Heidi, HumanIK, IDEA Server, i-drop, ImageModeler, iMOUT, Incinerator, Inferno, Inventor, Inventor LT, Kaydara, Kaydara (design/logo), Kynapse, Kynogon, LandXplorer, Lustre, MatchMover, Maya, Mechanical Desktop, Moldflow, Moonbox, MotionBuilder, Movimento, MPA, MPA (design/logo), Moldflow Plastics Advisers, MPI, Moldflow Plastics Insight, MPX, MPX (design/logo), Moldflow Plastics Xpert, Mudbox, Multi-Master Editing, Navisworks, ObjectARX, ObjectDBX, Open Reality, Opticore, Opticore Opus, Pipeplus, PolarSnap, PortfolioWall, Powered with Autodesk Technology, Productstream, ProjectPoint, ProMaterials, RasterDWG, RealDWG, Real-time Roto, Recognize, Render Queue, Retimer, Reveal, Revit, Showcase, ShowMotion, SketchBook, Smoke, Softimage, SoftimageXSI (design/logo), Sparks, SteeringWheels, Stitcher, Stone, StudioTools, ToolClip, Topobase, Toxik, TrustedDWG, ViewCube, Visual, Visual LISP, Volo, Vtour, Wire, Wiretap, WiretapCentral, XSI, and XSI (design/logo).

All other brand names, product names or trademarks belong to their respective holders.

Disclaimer

THIS PUBLICATION AND THE INFORMATION CONTAINED HEREIN IS MADE AVAILABLE BY AUTODESK, INC. "AS IS." AUTODESK, INC. DISCLAIMS ALL WARRANTIES, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE REGARDING THESE MATERIALS.

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

Images Courtesy of: Absolute Post, B Sky B Creative Services, Creative bubble, Duckling A-S, Finish

Autodesk Backburner Installation Guide for Autodesk 3ds Max

Document Version: 3

Date: April 1, 2010

Contents

	Part 1: Installation	1
Chapter 1	Introduction	3
	Welcome	3
	Intended Audience and Prerequisites	4
	Backburner Terminology	4
	Packets, Buckets and Backburner	5
	Backburner Architecture	5
	Backburner Manager	6
	Backburner Monitor	6
	Render Nodes	6
	Deployment Models	7
	Deployment Model One: Standalone Workstation	7
	Deployment Model Two: Render Farm	7
	Command-Job Utility	8
	Backburner SDK and Third-Party Development	8
	System Requirements	8
	Recommendations	8
	What was New in Backburner 2011	9
	Backburner on Mac OS X	9
	New Backburner Web Monitor	9
	Support for Smoke 2010 for Mac OS X	9
	Support for Maya 2010 on Mac OS X	9
	New Command-Job Adapter	9
	Combustion Compatibility Break	10
	Notation Conventions	10
	Related Documentation	10
	Contacting Customer Support	10
Chapter 2	Configuring a Standalone Workstation	11
	Overview	11

3ds Max	11
3ds Max Composite	13
Inferno, Flame, Flint, Smoke, Flare & Backdraft Conform	13
Background I/O	13
Cleaner XL	17
Burn	18
WiretapCentral	18
Smoke for Mac OS X	19
Background I/O	19
Cleaner XL	23
Burn	24
WiretapCentral	25
Lustre	25
Maya	26
Maya Composite	26
Chapter 3 Setting up a Render Farm	27
Overview	27
Render Farm Deployment Options	28
Dedicated Render Farm	28
Shared Render Farm	29
Isolating the Backburner Manager	29
Setting up Monitoring	30
Windows Monitor versus Web Monitor	31
Backburner Web Monitor and User Management	33
Setting Up Render Nodes	34
Handling Bulk Data	35
3ds Max	35
Visual Effects and Finishing	36
3ds Max	37
3ds Max Dedicated Render Farm	37
3ds Max Shared Render Farm	38
Visual Effects and Finishing	39
Visual Effects and Finishing with Cleaner XL	41
Visual Effects and Finishing with WiretapCentral	41
Smoke for Mac OS X	42
Smoke for Mac OS X with Cleaner XL	43
Smoke for Mac OS X with WiretapCentral	43
Lustre	44
Lustre Background Rendering	44
Lustre Background Transcoding	45
Maya	46
Maya Dedicated Render Farm	46
Maya Shared Render Farm	48
Maya Composite / 3ds Max Composite	51
Composite Dedicated Render Farm	51
Composite Shared Render Farm	52
Part 2: Installation Reference	53
Chapter 4 Backburner Components – Windows	55
Overview	55
Backburner Manager	56
Backburner Server	56
Installing the Backburner Components on Windows	56
Starting and Configuring the Backburner Manager	57
Running the Backburner Manager as a Windows Service	59

	Starting and Configuring the Backburner Windows Monitor	60
	Starting and Configuring the Backburner Server	60
	Configuring the Backburner Server	62
	Setting Up Backburner Server as a Windows Service	63
	Creating a Privileged User Account	63
	Configuring Backburner Server as a Windows Service	64
	Assigning the Privileged User Account to the Service	65
Chapter 5	Backburner Manager – Linux	67
	Overview	67
	Installing the Backburner Components on Linux	68
	Starting and Configuring Backburner Manager	68
Chapter 6	Backburner Web Monitor – Linux	71
	Overview	71
	Installing Backburner Web Monitor	72
	Installing the Apache HTTP Server for Linux	72
	Installing the Apache HTTP Server Software	72
	Setting Up Access to the Backburner Web Monitor	73
	Assigning Backburner Web Monitor Administrator Privileges	73
Chapter 7	Backburner Components – Mac	75
	Overview	75
	Backburner Manager Installation Overview	76
	Installing the Backburner Components on Mac OS X	76
	Configuring the Backburner Manager	77
	Setting Up Access to the Backburner Web Monitor	78
	Assigning Backburner Web Monitor Administrator Privileges	78
	Configuring the Backburner Server	79
Appendix A	Backburner Manager Configuration Settings	81
	Overview	81
	TCP/IP Settings	82
	General Settings	82
	Failed Node Behavior	83
	Job Path Settings	83
	Default Job Handling Behavior	84
	Index	87

Part 1: Installation

Part 1 includes these chapters:

- [Introduction](#) on page 3
- [Configuring a Standalone Workstation](#) on page 11
- [Setting up a Render Farm](#) on page 27



Image courtesy of Creative Bubble

Introduction

1

Topics in this chapter:

- [Welcome](#) on page 3
- [Backburner Terminology](#) on page 4
- [Backburner Architecture](#) on page 5
- [Deployment Models](#) on page 7
- [Command-Job Utility](#) on page 8
- [Backburner SDK and Third-Party Development](#) on page 8
- [System Requirements](#) on page 8
- [What was New in Backburner 2011](#) on page 9
- [Notation Conventions](#) on page 10
- [Related Documentation](#) on page 10
- [Contacting Customer Support](#) on page 10

Welcome

Welcome to the *Autodesk Backburner Installation Guide*. Backburner™ is the Autodesk® queue manager for background processing and distributed network processing. It allows multiple jobs, such as I/O operations, grading, composites, and animation scenes, to be processed by many computers working collectively on the same network. Backburner provides the means to monitor and control the jobs you submit, as well as tools for managing the Backburner network itself.

Backburner is leveraged by the following Autodesk applications.

■ Flint®	■ Flame®	■ Inferno®	■ Smoke®	■ Flare™	■ WiretapCentral™
■ 3ds Max®	■ Lustre®	■ Maya®	■ Cleaner®	■ Burn®	■ Backdraft® Conform

NOTE Consult the 3ds Max documentation for information on the version of Backburner supported.

This guide relates to installation and configuration. For help using Backburner, itself, refer to the *Autodesk Backburner User Guide for Autodesk 3ds Max*. For information on submitting jobs to Backburner from an Autodesk application, refer to the applicable user guide.

Intended Audience and Prerequisites

This guide is for those responsible for installing and setting up Backburner. Familiarity with the following topics will be helpful:

- Autodesk applications
- Linux® and/or Windows® and/or Mac OS®
- Computer networking

If you require assistance, please contact Autodesk Media and Entertainment customer support. See [Contacting Customer Support](#) on page 10.

Backburner Terminology

Familiarity with the following terms will help you to understand this guide.

Term	Definition
Backburner	The Autodesk distributed job management system for execution of rendering and I/O jobs in the background.
Job	A set of one or more tasks submitted to Backburner for processing, such as a 3ds Max scene, Flame Batch setup, or background I/O job.
Task	The smallest unit of work that can be sent to a render node for processing. This could be a single frame, a portion of a frame, or an entire background I/O job, depending on the job type and the creative application submitting it.
Block	A group of tasks sent to a render node as a unit. For example, when you submit a Burn job from Smoke, each render node normally receives a number of frames for processing at once.
Backburner Manager	Coordinates jobs submitted by creative applications and delegates them to the Backburner servers on the Backburner network. There should be just one instance of the Backburner Manager running on the network at a time.
Backburner Windows Monitor Backburner Web Monitor	Front-end interfaces for management and control of the Backburner Manager.
Backburner Web Server	A collection of Common Gateway Interface (CGI) scripts and Adobe® Flash® SWF files, residing on the same workstation as the Backburner Manager, and responsible for generating the UI for the Backburner Web Monitor.

Term	Definition
Backburner Server	The job-processing component of Backburner, residing on each render node, that invokes the local processing engine.
Adapter	The means by which job-specific processing engines are integrated into Backburner. Sits between the Backburner server and the processing engine.
Processing Engine	The server-side process responsible for carrying out the tasks assigned by the Backburner Manager—rendering animations, scenes and frames, or performing background I/O. A renderer is a particular kind of processing engine.
Server Group	A named collection of Backburner servers.

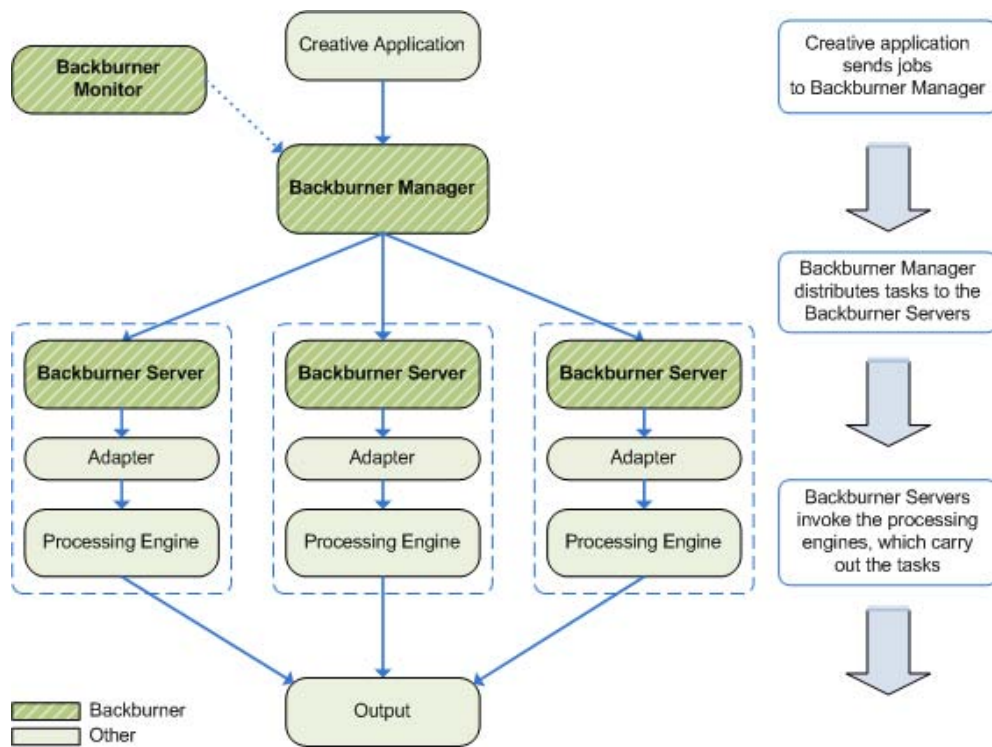
Packets, Buckets and Backburner

In other industry literature, the term *packet* is sometimes used to mean the number of frames sent to a render node for processing. A “packet” size of 5, for example, means 5 frames are sent to each available render node. In Backburner, the term *packet* retains its traditional meaning under TCP/IP. The blocks of tasks automatically sent to render nodes for processing are similar to what other software manufacturers call packets.

Similarly, *bucket rendering* is a commonly used term for the concept of explicitly dividing a single image into slices (or tiles), for processing on different render nodes. The same concept applies in Backburner (for 3ds Max), but no particular term is used to describe the process. Backburner simply receives jobs from its render clients, and sends them to the render nodes for processing as blocks of tasks. Whether tasks represent entire frames, or portions of frames, is transparent to its operation.

Backburner Architecture

Backburner consists of the Backburner Manager, Backburner Monitor, and Backburner Servers, as illustrated in the following diagram. These operate in the greater context of creative applications (such as Autodesk Visual Effects and Finishing applications) and processing engines (such as Burn™).



Backburner Manager

At the centre of Backburner is the Backburner Manager. It receives jobs from the clients, which it then distributes to the render nodes on the network. The Backburner Manager maintains status information about its network of Backburner servers. It also maintains a database of submitted, active, and completed jobs.

Backburner Monitor

End-user and administrator interaction with the Backburner Manager is by way of the Backburner Monitor. It is used to monitor the progress of a job, suspend and restart jobs, and perform administrative tasks.

Render Nodes

Render nodes consist of a Backburner Server, adapters, and processing engines. The Backburner Server is tasked with carrying out the jobs assigned to it by the Backburner Manager. It does so by passing the jobs on to the correct processing engine via the adapter. The adapter and processing engine are furnished by the creative application. The adapter receives instructions from the Backburner Server and controls the rendering engine.

The kinds of jobs a render node can process depend on the adapters and processing/rendering engines installed upon it. Autodesk Visual Effects and Finishing applications share the Burn rendering engine and Wire® processing engine. Some Autodesk applications, such as 3ds Max®, have their own rendering engine. Cleaner®, also, provides its own rendering engine.

NOTE Installing and configuring the background/network processing engine for particular products is beyond the scope of this document. For Inferno, Flame, Flint, Fire, Smoke, or Backdraft Conform, refer to the *Autodesk Burn User Guide*. For Lustre, refer to the *Autodesk Lustre Installation and Configuration Guide*.

Deployment Models

There are two main deployment models for Backburner: the standalone workstation and the render farm. Often you will find that everything needed for background rendering on a standalone workstation (a single machine) is already in place, since Backburner is installed, by default, with most creative applications. For a render farm, more research, planning, and configuration is needed.

Deployment Model One: Standalone Workstation

The following figure shows the most basic situation, in which all Backburner components are installed on the same workstation as the creative application.

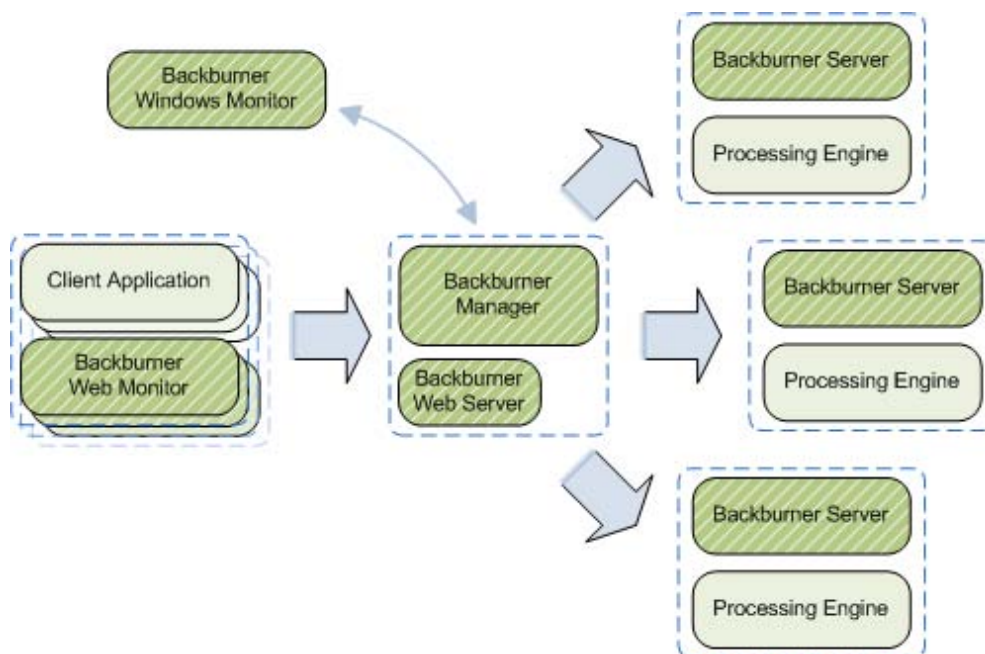


In this scenario, Backburner enables background processing with queue management for a single machine—there is no network rendering. The advantage is that the artist can continue working in the foreground while jobs are being processed in the background. Queue control is also possible. In this case, resources are shared between the foreground and background tasks.

For most creative applications, this is the “default” configuration, automatically created as part of the standard install. This subject is covered in [Configuring a Standalone Workstation](#) on page 11.

Deployment Model Two: Render Farm

The following figure shows Backburner deployed as a render farm. In this case, the off-loading and parallelization of processing increases productivity tremendously. A render farm can be created for a single creative application, or to service multiple creative applications at the same time.



The render farm deployment model requires planning and consideration. It is the subject of [Setting up a Render Farm](#) on page 27.

Command-Job Utility

The Backburner command-job utility, *cmdjob*, lets you submit batch, executable, or script files to Backburner as “custom” jobs from the command line, for execution on render nodes. Two distinct command-job adapters are installed on the render nodes to handle the custom jobs. One adapter executes jobs as the *root* user. The other retains the user account of the user that submitted the job. See the *Backburner User Guide*.

Backburner SDK and Third-Party Development

The Backburner Manager is a Wiretap™ server. This makes it possible to create custom job and server monitoring and managing tools for all instances of Backburner Manager in your facility. See the *Wiretap SDK Developer Guide*.

System Requirements

NOTE This section presents the system requirements for Backburner components. For client applications and render nodes, please refer to the specific application's installation guide, user guide, and/or online help.

Component	Windows	Linux	Mac OS
Backburner Manager	■ XP Professional SP2 ■ Vista (32- and 64-bit)	■ Red Hat® Enterprise Linux WS 4 (and later)	■ Mac OS X
Backburner Web Server	■ <i>not applicable</i>	■ Red Hat Enterprise Linux WS 4 (and later) ■ Apache HTTP web server 2.0.59 (or later)	■ Mac OS X ■ Apache HTTP web server 2.2.3 (or later)
Backburner Windows Monitor	■ XP Professional SP2 ■ Vista (32- and 64-bit)	■ <i>not applicable</i>	■ <i>not applicable</i>
Backburner Web Monitor	■ web browser ■ Adobe Flash Player 10 (or later)	■ web browser ■ Adobe Flash Player 10 (or later)	■ web browser ■ Adobe Flash Player 10 (or later)

Recommendations

When setting up a render farm, please note the following recommendations:

- **Dedicated Workstation for Backburner Manager:** The Backburner Manager performs best on a workstation reserved for its exclusive use. In particular, avoid running the Backburner Manager on a workstation that is also being used as a render node or by a client application.
- **Backburner Web Monitor:** For most monitoring needs, the Backburner Web Monitor will suffice, and is the recommended option, since it has little impact on system performance (see [Windows Monitor versus](#)

[Web Monitor](#) on page 31). Note that it requires that a Web server be installed on the workstation running the Backburner Manager.

- **Network Card:** A network interface card is required for each workstation in the Backburner network. Backburner will work with any standard network card, but for best results we recommend one that is Gigabit Ethernet capable.

What was New in Backburner 2011

This section presents new features and changes implemented in Backburner 2011.

Backburner on Mac OS X

In previous releases, Backburner was deployable on both the Linux and Windows operating systems. With Backburner 2011, Backburner adds Mac OS X to its list of supported platforms. The Backburner Manager, Backburner Server, and Backburner *cmdjob* command-line utility are all included in the port, enabling Autodesk applications to leverage Backburner for render farms consisting of managers and render nodes on Mac workstations.

New Backburner Web Monitor

This release of Backburner features a new version of the Backburner Web Monitor. The new Web Monitor is a full-featured, desktop-like application that runs entirely inside a web browser. It is an Adobe Flash Player compatible application that replaces the older html-based Backburner Web Monitor, which is no longer available as of this release. The new Backburner Web Monitor is available for Linux and Mac OS only.

Support for Smoke 2010 for Mac OS X

Backburner is now leveraged by the new Autodesk Smoke for Mac OS X application, in both standalone and render-farm deployments. In the standalone deployment, Smoke for Mac OS X uses Backburner for background I/O, including background proxy generation. In the render farm deployment, Smoke 2010 for Mac OS X can submit jobs to a render farm consisting of Burn render nodes installed on Linux workstations.

Support for Maya 2010 on Mac OS X

In the previous release, Maya leveraged Backburner for network rendering on Linux and Windows workstations. With Maya 2010 and Backburner 2011, network rendering is also supported on the Mac OS X platform. Network rendering allows animation scenes to be rendered by many computers working collectively on the same network.

New Command-Job Adapter

The Backburner command-job utility, *cmdjob*, lets you submit batch, executable, or script files to Backburner as “custom” jobs from the command line, for execution on the render nodes. This release introduces a new command-job adapter, with the result that two distinct command-job adapters are now installed on the render nodes.

The original adapter runs all received jobs as the *root* user account. The new adapter runs its jobs as the account that submitted the job, allowing enforcement of a more stringent security mechanism. Those concerned with security now have the option of removing the older adapter.

To invoke the new adapter, use the new *-userRights* job submission switch. The new adapter is available for Linux and Mac OS only. It is not available for Windows render nodes.

Combustion Compatibility Break

With this release, Combustion network rendering using Backburner is no longer supported. Note, however, that Backburner is not needed for local rendering, nor for local encoding using Cleaner XL. The following table summarizes the role of Backburner in Combustion.

Processing Option	Backburner
Local rendering using the Autodesk Combustion Render Queue™	n/a
Network rendering using the Autodesk Combustion Render Queue	n/a
Network rendering using Backburner	not supported
Local encoding with Cleaner XL	n/a
Network encoding with Cleaner XL	<i>supported</i>

Notation Conventions

A number of style conventions are used throughout this guide. 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 font shown in the example. You must press the Enter key after each command.	rpm -qa
Variable names appear in Courier, enclosed in angle brackets.	<variable_name>
Variables that appear enclosed in square brackets are optional.	[<filename>]
Feedback from the command line or shell appears in Courier.	limit coredumpsize
Directory names, filenames, URLs, and command line utilities appear in italics.	<i>/usr/discreet</i>

Related Documentation

This release has documentation that helps you install, configure, and use the software. For a list of all the documentation available to you, visit <http://www.autodesk.com/me-documentation>. From this page you can access the complete documentation library.

You should also refer to your product's release notes for all late-breaking release information.

Contacting Customer Support

For a list of contact information for Autodesk Media and Entertainment Customer Support, consult your release notes, or 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>.

Configuring a Standalone Workstation

2

Topics in this chapter:

- [Overview](#) on page 11
- [3ds Max](#) on page 11
- [Inferno, Flame, Flint, Smoke, Flare & Backdraft Conform](#) on page 13
- [Smoke for Mac OS X](#) on page 19
- [Lustre](#) on page 25
- [Maya](#) on page 26

Overview

In a standalone deployment, all the Backburner and creative application components needed for background processing are installed on a single workstation. No render farm is necessary. For some creative applications, all the necessary components are already installed. All you need to do is a little configuration.

This chapter tells you everything you need to know to set up background rendering on a standalone workstation. It takes a look at the standalone possibilities for each creative application leveraged by Backburner. It provides step-by-step instructions, and advises where to look for additional information, when needed. Its purpose is to help you determine what you can do without setting up a render farm, and provide the detailed steps on how to do it.

3ds Max

Backburner is installed by default during the 3ds Max and 3ds Max Design installation. In this standalone configuration, Backburner can be used to perform background rendering with queue management on a single machine. It can also be used to break a large image into horizontal or vertical “strips” to reduce the

resources—such as memory—required for rendering. The standalone configuration for 3ds Max is illustrated in the following figure.



In the standalone configuration, specifying 'network' rendering when submitting jobs from 3ds Max invokes Backburner. 3ds Max submits the job to the locally installed Backburner Manager. The Backburner Server then invokes the 3ds Max renderer, which carries out the task of rendering. The Backburner Monitor can be used to manage queued tasks.

To set up local background rendering in 3ds Max:

NOTE This procedure assumes 3ds Max, including all Backburner components, are already installed on the workstation. If this is not the case, please refer to the *3ds Max Installation Guide*.

- 1 Start the Backburner Manager. From the Start menu, choose Programs, Autodesk, Backburner, and then Manager.
- 2 Start the Backburner Monitor. From the Start menu, choose Programs, Autodesk, Backburner, and then Monitor.
- 3 Connect the monitor to the manager. From the Manager menu, choose Connect. Enter the following for the manager name/IP address:
localhost
- 4 Verify the connection. From the Manager menu, choose Properties. Manager details appear in a dialog box. If no connection was made, this option is greyed-out.
- 5 Start the Backburner Server. From the Start menu, choose Programs, Autodesk, Backburner, and then Server. The General Properties dialog appears. If it does not, from the Edit menu, choose General Settings.
- 6 Verify the following items are set correctly in the General Properties dialog:
 - *Server Name or IP Address*: Used by the Backburner Manager to identify and communicate with the server. This field should contain the machine name or IP address of the current workstation.
 - *Automatic Search*: Unchecked.
 - *Enter Manager Name or IP Address*: Used by the Backburner Server to locate and communicate with the manager. The keyword **localhost** should suffice, since it resolves to the IP address of the local workstation.
- 7 Verify the connection in the monitor window or server window.

For more information on:

- Configuring the Backburner Manager to start with the OS and run in the background, see [Running the Backburner Manager as a Windows Service](#) on page 59.
- Configuring the Backburner Server to start with the OS and run in the background, see [Setting Up Backburner Server as a Windows Service](#) on page 63.
- Submitting rendering jobs for processing in the background, see *3ds Max User Guide*.

3ds Max Composite

There is no standalone deployment model for Backburner with Composite, since Backburner is not needed for local background rendering. In Composite, Backburner is used for network rendering only.

See [Setting up a Render Farm](#) on page 27.

Inferno, Flame, Flint, Smoke, Flare & Backdraft Conform

The Visual Effects and Finishing applications leverage Backburner for the following purposes:

- Background I/O
- Video encoding
- Network rendering
- Media conversion

NOTE Smoke for Mac OS X is discussed in its own section. See [Smoke for Mac OS X](#) on page 19.

The following table explains the role of Backburner in each case.

Item	Description	Role of Backburner
Background I/O	Visual Effects and Finishing applications have the built-in ability to <i>wire</i> transfer, resize, and generate proxies for clips as background tasks.	Queue manager
Video encoding	Cleaner XL provides video encoding to a variety of formats.	Queue manager
Network rendering	Burn is the rendering engine for Visual Effects and Finishing applications.	Queue manager
Media Conversion	WiretapCentral allows you to view and convert the media content of your facility without the need for a running Visual Effects and Finishing application.	Off-loading the conversion, and for scalability

NOTE Visual Effects and Finishing applications have some queue monitoring abilities built into the user interface. For more complete queue control you can install the Backburner Monitor.

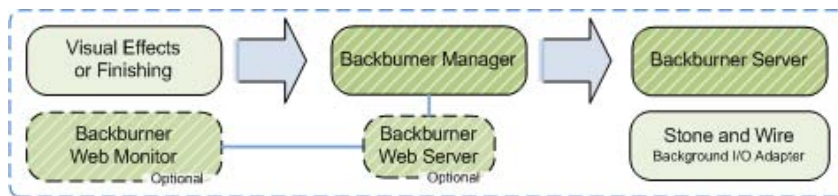
Background I/O

Background I/O is a media management tool for performing I/O with minimum impact on the performance of the creative workstation. The tool has both a command-line and an embedded interface, and can be used to perform the following operations:

- Transfer clips using Wire
- Publish clips
- Stonify soft-imported clips
- Resize clips
- Generate proxies for clips

NOTE The embedded version of the tool makes use of Backburner for two operations only: to transfer clips using *Wire* and to generate proxies for clips. The other operations do not require Backburner. In contrast, the command-line version of the tool uses Backburner for all operations.

The necessary components for the background I/O features that make use of Backburner are illustrated in the following figure.



NOTE 1. Stone® and Wire are automatically installed during installation of the Visual Effects and Finishing application. 2. Not all Backburner components may have been installed during product installation. 3. The Backburner Web Monitor and Backburner Web Server are optional, for the purposes of Background I/O.

Requesting a “Background Wire” transfer from the Clip Library menu invokes Backburner. The Visual Effects and Finishing application submits the job to the locally installed Backburner Manager. The Backburner Manager then invokes the Backburner Server, which calls upon the Wire daemon to carry out the task of transferring the media. Monitoring is embedded in the Visual Effects and Finishing application. It can also be done using the Backburner Web Monitor (optional).

To set up background I/O in Visual Effects and Finishing:

NOTE This procedure assumes that the Visual Effects and Finishing application and all Backburner components are already installed on the workstation. It also assumes the Backburner Manager and Backburner Server have been configured to run as daemons, and start with the workstation's operating system. If this is not the case, please install the Visual Effects and Finishing software and then proceed to [Installing the Backburner Components on Linux](#) on page 68.

- 1 Ensure the Backburner Manager and Backburner Server are installed and set to run as daemons by typing the following:

```
chkconfig --list | grep -i Backburner
```

Output similar to the following should appear:

```
backburner_manager 0:off 1:off 2:on 3:on 4:on 5:on 6:off
backburner_server  0:off 1:off 2:on 3:on 4:on 5:on 6:off
```

Levels “3”, “4”, and “5” must be “on” for the daemons to start automatically. If this is not the case, reinstall the Backburner components.

- 2 Ensure the token **localhost** appears in the following file:

```
/usr/discreet/backburner/cfg/manager.host
```

The **manager.host** file is used by the Backburner Server to locate the manager. For background I/O, it must contain a token indicating the IP address or host name of the local machine. In this case, the token **localhost** will suffice, since it resolves to the IP address of the local workstation.

- 3 Restart the Backburner Manager and Backburner Server so they pick up the new settings:

```
/etc/init.d/backburner restart
```

- 4 Ensure the **BackburnerManagerHostname** keyword in the following file is set to the IP address of a valid manager:

```
/usr/discreet/<product_home>/cfg/init.cfg
```

For example, in most cases you can set it to **localhost**:

BackburnerManagerHostname localhost

NOTE The **BackburnerManagerHostname** keyword is also used to indicate the manager for Burn and Cleaner XL. If either of these other systems has been set up for this seat of the creative application, you can indicate that manager instead.

- 5 Restart the Visual Effects and Finishing application so it picks up the new configuration file setting and begins communicating with Backburner.

For more information on:

- Installing and configuring the Backburner Web Server, see [Backburner Web Monitor – Linux](#) on page 71.
- Configuring Backburner Manager behavior, see [Backburner Manager Configuration Settings](#) on page 81.
- Setting up the Backburner Web Monitor, see [Backburner Web Monitor – Linux](#) on page 71.
- Submitting background I/O jobs, see the application-specific user guide.

To verify background proxy generation is operational:

This procedure verifies background I/O using Backburner is operational by importing an image sequence from the filesystem into a clip library and generating proxies for it, in the background.

- 1 Create a new project for which proxy generation is always on or is conditional.
To do this, edit the project's settings in the Preferences Menu, Project Management group.

NOTE It is not recommended that you enable proxies or otherwise change proxy settings for an already existing project, since it can result in the generation of proxies for all clips already in the project in the *foreground*, an operation that can take considerable time. The purpose of this procedure is to test proxy generation in the *background*.

- 2 In the library, click Import Image.
- 3 In the Clip Media group, enable Soft-Import Full Res.
- 4 In the Background Operations group, enable Generate/Import Proxies.

NOTE The Proxy Generation button in this area of the software does not determine whether proxies are generated or not. It specifies *how* they are generated: in the background (with Backburner) or the foreground (Backburner not needed).

- 5 Use the file browser to select and import media files from your filesystem.
- 6 Verify the proxies are generated in the background, in a timely manner:
 - View progress of the background I/O in the Background Import/Export Queue. In the Library menu, click the Background Monitor button (or press **Ctrl+O** from anywhere in the application).
 - Review messages appearing in the message bar in the Message History window. In the Preferences menu, Miscellaneous group, click the Message History button (or press **Ctrl+Alt+F7** from anywhere in the application).
 - If the playable thumbnails indicate “Pending Render” for too long, this is a sign that the proxies are not being generated.

For more information on:

- Setting proxy options, refer to the *Autodesk Smoke User Guide*.
- Managing the Background Import/Export Queue, refer to the *Autodesk Smoke User Guide*.
- Message History window, refer to the *Autodesk Smoke User Guide*.

To verify background *Wire* transfer of clips is operational:

This procedure verifies background I/O using Backburner is operational by importing an image sequence from a remote clip library into a local clip library. This operation makes use of the background I/O *Wire* feature, as well as Backburner. In general, requesting a “Background Wire” transfer from the Clip Library menu invokes Backburner and the *Wire* transfer.

NOTE In order to perform this procedure, Autodesk Stone and Wire must be installed on a remote system. Stone and Wire are automatically installed during the installation of all Visual Effects and Finishing applications.

- 1 Create a new project or open an existing project.
- 2 Disable any settings that might cause proxies to be generated or the clip to be resized.
To do this, edit the project's settings in the Preferences Menu, Project Management group.
- 3 Open the clip library and select the following options:
 - Dual Library View
 - Show All Libraries
 - Copy on Drag
- 4 In the Clip Library menu, click Network.
The local system is listed at the top of the network library. Remote systems are listed below it, in alphabetical order.
- 5 Select a library on a remote system and exit the network library.
Selected libraries are highlighted and marked with a “C” (connect) or “AC” (auto-connect).
- 6 Return to the clip library, navigate to the remote library in one view, and the local library in the other.
- 7 In the clip library Wire Options area, enable Background Wire.
- 8 Drag and drop a clip from the remote library to your local library.
- 9 Visually verify the selected clip is successfully copied.
Additional verification is also possible:
 - View progress of the background I/O in the Background Import/Export Queue. In the Library menu, click the Background Monitor button (or press **Ctrl+O** from anywhere in the application).
 - Review messages appearing in the message bar in the Message History window. In the Preferences menu, Miscellaneous group, click the Message History button (or press **Ctrl+Alt+F7** from anywhere in the application).

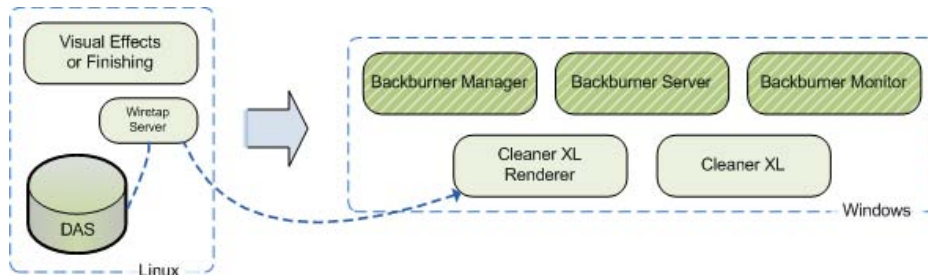
For more information on:

- Copying clips from other projects, refer to the *Autodesk Smoke User Guide*.
- Copying local and remote clip libraries to your local project, refer to the *Autodesk Smoke User Guide*.

Cleaner XL

Cleaner XL encodes jobs to a variety of formats, including Quicktime®, Windows Media®, and Real®. It receives jobs from the creative application, and saves the output files to disk. For Visual Effects and Finishing applications, Backburner provides the communication between the creative application and the Windows workstation running Cleaner XL.

The following diagram shows a typical setup for a Visual Effects or Finishing workstation and Cleaner XL.



As you can see, for a Visual Effects and Finishing application, the setup is not strictly speaking “standalone”, since a separate workstation is required. Visual Effects and Finishing applications run under Linux, while Cleaner XL is a Windows application (it can be run on a laptop). Note that the Wiretap daemon is installed by default during the Visual Effects and Finishing installation, and is used by Cleaner XL to acquire media from the creative workstation's direct attached storage (DAS). The Backburner components are normally installed during the Cleaner XL installation process.

The action of sending a clip from the Visual Effects or Finishing application to Cleaner XL invokes Backburner. The Backburner Manager receives the job and invokes the Backburner Server on the Windows machine. The server, in turn, calls upon the Cleaner XL renderer to carry out the task of encoding the media.

An important element in the workflow is the role played by Cleaner XL “output profiles”. Output profiles specify the type of encoding desired, audio and video filter settings, and the output destinations. Output profiles must be created in Cleaner XL and manually transferred to the Linux workstation.

Note that the Cleaner XL renderer pulls raw, uncompressed RGB data from the Linux workstation's storage, by way of the Wiretap daemon. As a result, the Linux and Windows machines must be on the same subnet/switch, or severe network bottlenecks can occur.

To set up a Visual Effects or Finishing application for Cleaner XL:

NOTE This procedure assumes the Visual Effects and Finishing application, Cleaner XL, and Backburner components are already installed on their respective workstations. If this is not the case, please refer to the application-specific installation guide.

- 1 On the Windows workstation, start the Backburner Manager. From the Start menu, choose Programs, Autodesk, Backburner, and then Manager.
- 2 Start the Backburner Monitor. From the Start menu, choose Programs, Autodesk, Backburner, and then Monitor.
- 3 Connect the monitor to the manager. From the Manager menu, choose Connect. Enter the following for the manager name/IP address:
localhost
- 4 Verify the connection. From the Manager menu, choose Properties.
Manager details appear in a dialog. If no connection was made, this option is greyed out.
- 5 Start the Backburner Server. From the Start menu, choose Programs, Autodesk, Backburner, and then Server.

The General Properties dialog appears. If it does not, from the Edit menu, choose General Settings.

- 6 Verify the following items are set correctly in the General Properties dialog:
 - *Server Name or IP Address*: Used by the Backburner Manager to identify and communicate with the server. This field should contain the machine name or IP address of the current workstation.
 - *Automatic Search*: Unchecked.
 - *Enter Manager Name or IP Address*: Used by the Backburner Server to locate and communicate with the manager. The keyword **localhost** should suffice, since it resolves to the IP address of the local workstation.
- 7 Verify the connection in the monitor window or server window.
- 8 On the Linux workstation, configure the Visual Effects and Finishing application to point to the Backburner Manager by editing the following file:
`/usr/discreet/<product_home>/cfg/init.cfg`
- 9 In `init.cfg`, set the **BackburnerManagerHostname** keyword as follows:
BackburnerManagerHostname <WindowsWorkstation>
where <WindowsWorkstation> is the machine name or IP address of the Windows workstation where Cleaner XL and Backburner are installed.

For more information on:

- Detailed setup instructions and troubleshooting, refer to the *Using Autodesk Cleaner XL with Autodesk Effects and Finishing Applications* guide.
- Sending jobs to Cleaner XL, refer to the application-specific user guide.
- Details on encoding using Cleaner XL, refer to the *Autodesk Cleaner XL User Guide*.

Burn

Burn is the rendering engine shared by Visual Effects and Finishing applications. When you submit rendering jobs to the Backburner network from a creative application, it is Burn that performs the rendering. There is no standalone deployment model for Burn. It is always deployed as a render farm.

For more information on:

- Guidelines for network rendering, see [Setting up a Render Farm](#) on page 27.
- Detailed instructions, refer to the *Burn Installation and User Guide*.

WiretapCentral

Visual Effects and Finishing applications use WiretapCentral to read and decode formats such as OpenEXR and Red (.r3d) files, to a Stone FS or Standard FS filesystem. WiretapCentral leverages Backburner to off-load transcoding. In a standalone installation, all the background processing components—Backburner server, Backburner Manager, and the MIO Adapter—are installed on the same workstation as the Visual Effects and Finishing application.

This is the simplest configuration, since all components are automatically installed and set up by the Visual Effects and Finishing installer. However, since all media processing is performed locally, it can have a significant impact on system performance. The standalone configuration is best-suited for occasional media encoding and decoding. For more demanding uses, a render farm is the recommended configuration.

For detailed installation, setups and configuration instructions, refer to the *WiretapCentral and Wiretap Gateway Installation and Configuration Guide*.

Smoke for Mac OS X

Smoke for Mac OS X leverages Backburner for the following purposes:

- Background I/O
- Video encoding
- Network rendering
- Media conversion

The following table explains the role of Backburner in each case.

Item	Description	Role of Backburner
Background I/O	Smoke for Mac OS X has the built-in ability to <i>Wire</i> transfer clips from remote framestores into the local library, plus resize and/or generate proxies for the transferred clips. Using Backburner, all these operations can be performed in the background.	Queue manager
Video encoding	Cleaner XL provides video encoding to a variety of formats.	Queue manager
Network rendering	Burn is the rendering engine for Smoke for Mac OS X.	Queue manager
Media Conversion	WiretapCentral allows you to view and convert the media content of your facility without the need for a running Smoke for Mac OS X.	Off-loading the conversion, and for scalability.

NOTE Smoke for Mac OS X has some queue monitoring ability built into the user interface. In the Library menu, click the Background Monitor button (or press **Ctrl+O** from anywhere in the application). For more complete queue control you can make use of the Backburner Monitor, installed by default with the application software.

Background I/O

In the standalone deployment, Smoke for Mac OS X leverages Backburner for background I/O. This provides the ability to *Wire* transfer clips from remote framestores into the local library, resize and/or generate proxies for the clips, entirely in the background. Having the work performed in the background means you can continue working in the current Smoke for Mac OS X user interface, without delay or interruption.

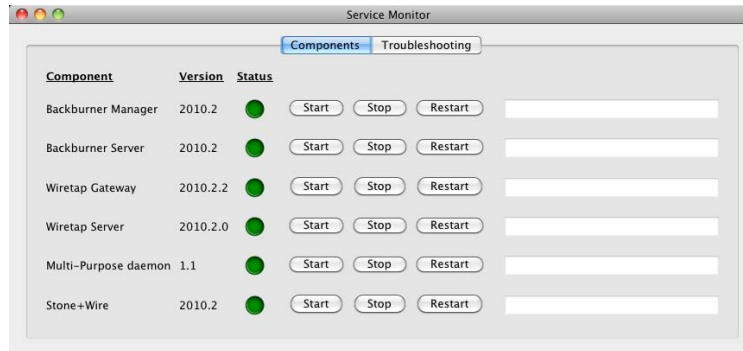
All the components needed for background I/O are installed and configured automatically with the Smoke for Mac OS X software. No special configuration is needed. This section provides supplementary information that can be used to verify all the needed components have been installed and auto-configured correctly, and to test that background I/O is functional.

NOTE For a more complete description of background I/O that includes a diagram, see [Inferno, Flame, Flint, Smoke, Flare & Backdraft Conform](#) on page 13.

To verify Backburner components are installed and running:

The simplest way to verify all needed Backburner components are installed and running is by using the Service Monitor.

- 1 To launch the Service Monitor, go to *Applications / Autodesk / Smoke Common Utilities* in Finder, and open the *Service Monitor* application.



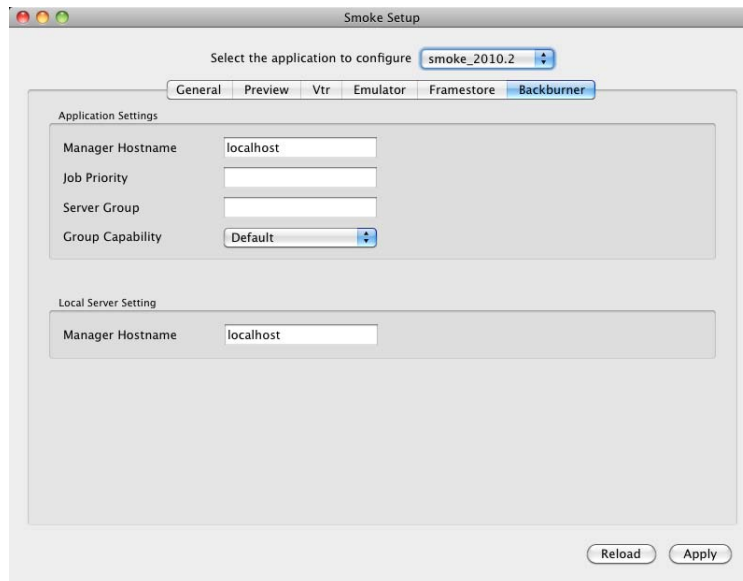
- 2 If the Backburner Manager and Backburner Server are not running, click Start or Restart.

For more information, refer to the *Autodesk Smoke Mac for OS X Installation and Configuration Guide*.

To verify Smoke and Backburner are configured correctly:

Once you have verified all needed Backburner components are installed and running, you can verify Backburner is configured correctly. This is done using the Smoke Setup application.

- 1 To launch, go to *Applications / Autodesk* in Finder, and open the *Smoke Setup* application, then select the Backburner tab.



The following table explains the contents of this tab.

Area	Field	Description
Application Settings		This area contains settings that enable Smoke for Mac OS X to send jobs to Backburner. Values you enter here are written to Smoke's locally stored <i>init.cfg</i> file.

Area	Field	Description
	Manager Hostname	Identifies the Backburner Manager to which Smoke for Mac OS X sends its jobs. Ensure this is set to the IP address or host name (workstation name) of a valid Backburner Manager. For a standalone setup, this can be set to the value <i>localhost</i> , since it resolves to the IP address of the local workstation.
	Job Priority	The priority assigned to jobs sent from this creative application. Valid values are 0–100. Default is 50. Can be left blank.
	Server Group	A server group is a named collection of servers that can assist in implementing a job processing strategy. Applies to render farm deployments only. Leave blank for a standalone implementation.
	Group Capability	The type of server group—for example, GPU-enabled or non-GPU. Applies to a render farm deployment only. Leave blank for a standalone deployment.
Local Server Setting	This area contains information used by the locally installed Backburner Server. Values you enter here are written to Backburner Server's locally stored <i>manager.host</i> file.	
	Manager Hostname	Used by the Backburner Server to locate the Backburner Manager. For background I/O, ensure it is set to the IP address or host name of a valid Backburner Manager. For a standalone setup, this can be set to the value <i>localhost</i> , since it resolves to the IP address of the local workstation.

NOTE The **Manager Hostname** in the **Local Server Setting** area is also used to indicate the Backburner Manager for Burn and Cleaner XL. If either of these other systems has been set up for this seat of Smoke for Mac OS X, you can indicate that manager instead. The same Backburner Manager can be used by multiple applications.

- 2 Make changes as needed, then click Apply.
The configuration files are updated accordingly.
- 3 Using the Service Monitor UI, restart the Backburner Manager and Backburner Server so they pick up the new settings.
- 4 Restart Smoke for Mac OS X.

To verify background proxy generation is operational:

This procedure verifies background I/O using Backburner is operational by importing an image sequence from the filesystem into a clip library and generating proxies for it, in the background.

- 1 Create a new project for which proxy generation is always on or is conditional.
To do this, edit the project's settings in the Preferences Menu, Project Management group.

NOTE It is not recommended that you enable proxies or otherwise change proxy settings for an already existing project, since it can result in the generation of proxies for all clips already in the project in the *foreground*, an operation that can take considerable time. The purpose of this procedure is to test proxy generation in the *background*.

- 2 In the library, click Import Image.
- 3 In the Clip Media group, enable Soft-Import Full Res.
- 4 In the Background Operations group, enable Generate/Import Proxies.

NOTE The Proxy Generation button in this area of the software does not determine whether proxies are generated or not. It specifies *how* they are generated: in the background (with Backburner), or the foreground (Backburner not needed).

- 5 Use the file browser to select and import media files from your filesystem.
- 6 Verify the proxies are generated in the background, in a timely manner:
 - View progress of the background I/O in the Background Import/Export Queue. In the Library menu, click the Background Monitor button (or press **Ctrl+0** from anywhere in the application).
 - Review messages appearing in the message bar in the Message History window. In the Preferences menu, Miscellaneous group, click the Message History button (or press **Ctrl+Alt+F7** from anywhere in the application).
 - If the playable thumbnails indicate “Pending Render” for too long, this is a sign that the proxies are not being generated.

For more information on:

- Setting proxy options, refer to the *Autodesk Smoke Mac OS X Help*.
- Managing the Background Import/Export Queue, refer to the *Autodesk Smoke Mac OS X Help*.
- Message History window, refer to the *Autodesk Smoke Mac OS X Help*.

To verify background Wire transfer of clips is operational:

This procedure verifies background I/O using Backburner is operational by importing an image sequence from a remote clip library into a local clip library. This operation makes use of the background I/O *Wire* feature, as well as Backburner. In general, requesting a “Background Wire” transfer from the Clip Library menu invokes Backburner and the *Wire* transfer.

NOTE In order to perform this procedure, Autodesk Stone and Wire must be installed on a remote system. Stone and Wire are automatically installed during the installation of all Visual Effects and Finishing applications.

- 1 Create a new project or open an existing project.
- 2 Disable any settings that might cause proxies to be generated or the clip to be resized.
To do this, edit the project's settings in the Preferences Menu, Project Management group.
- 3 Open the clip library and select the following options:
 - Dual Library View
 - Show All Libraries
 - Copy on Drag
- 4 In the Clip Library menu, click Network.
The local system is listed at the top of the network library. Remote systems are listed below it, in alphabetical order.
- 5 Select a library on a remote system and exit the network library.
Selected libraries are highlighted and marked with a “C” (connect) or “AC” (auto-connect).

- 6 Return to the clip library, navigate to the remote library in one view, and the local library in the other.
- 7 In the clip library Wire Options area, enable Background Wire.
- 8 Drag and drop a clip from the remote library to your local library.
- 9 Visually verify the selected clip is successfully copied.

Additional verification is also possible:

- View progress of the background I/O in the Background Import/Export Queue. In the Library menu, click the Background Monitor button (or press **Ctrl+O** from anywhere in the application).
- Review messages appearing in the message bar in the Message History window. In the Preferences menu, Miscellaneous group, click the Message History button (or press **Ctrl+Alt+F7** from anywhere in the application).

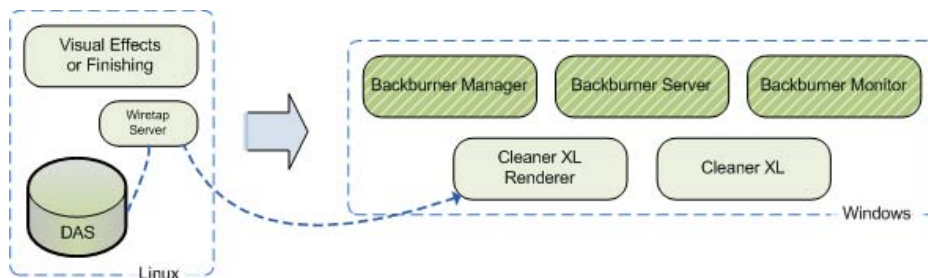
For more information on:

- Copying clips from other projects, refer to the *Autodesk Smoke Mac OS X Help*.
- Copying local and remote clip libraries to your local project, refer to the *Autodesk Smoke Mac OS X Help*.

Cleaner XL

Cleaner XL encodes jobs to a variety of formats, including Quicktime®, Windows Media®, and Real®. It receives jobs from the creative application, and saves the output files to disk. Backburner provides the communication between Smoke for Mac OS X on the Mac workstation, and the Windows workstation running Cleaner XL.

The following diagram shows a typical setup for a Smoke for Mac OS X workstation and Cleaner XL.



As you can see, for Smoke on Mac OS X, the setup is not strictly speaking “standalone”, since a separate workstation is required. Smoke for Mac OS X is a Mac OS application, while Cleaner XL is a Windows application (it can be run on a laptop). Note that the Wiretap daemon is installed by default during the Smoke for Mac OS X installation, and is used by Cleaner XL to acquire media from the creative workstation's direct attached storage (DAS). The Backburner components are normally installed during the Cleaner XL installation process.

The action of sending a clip from the Smoke for Mac OS X application to Cleaner XL invokes Backburner. The Backburner Manager receives the job and invokes the Backburner Server on the Windows machine. The server, in turn, calls upon the Cleaner XL renderer to carry out the task of encoding the media.

An important element in the workflow is the role played by Cleaner XL “output profiles”. Output profiles specify the type of encoding desired, audio and video filter settings, and the output destinations. Output profiles must be created in Cleaner XL and manually transferred to the workstation running Smoke for Mac OS X.

Note that the Cleaner XL renderer pulls raw uncompressed RGB data from the Mac workstation's storage, by way of the Wiretap daemon. As a result, the Mac and Windows machines must be on the same subnet/switch, or severe network bottlenecks can occur.

To set up Smoke for Mac OS X for Cleaner XL:

NOTE This procedure assumes the Smoke for Mac OS X application, Cleaner XL, and Backburner components are already installed on their respective workstations. If this is not the case, please refer to the application-specific installation guide.

- 1 On the Windows workstation, start the Backburner Manager. From the Start menu, choose Programs, Autodesk, Backburner, and then Manager.
- 2 Start the Backburner Monitor. From the Start menu, choose Programs, Autodesk, Backburner, and then Monitor.
- 3 Connect the monitor to the manager. From the Manager menu, choose Connect. Enter the following for the manager name/IP address:
localhost
- 4 Verify the connection. From the Manager menu, choose Properties.
Manager details appear in a dialog. If no connection was made, this option is greyed out.
- 5 Start the Backburner Server. From the Start menu, choose Programs, Autodesk, Backburner, and then Server.
The General Properties dialog appears. If it does not, from the Edit menu, choose General Settings.
- 6 Verify the following items are set correctly in the General Properties dialog:
 - *Server Name or IP Address*: Used by the Backburner Manager to identify and communicate with the server. This field should contain the machine name or IP address of the current workstation.
 - *Automatic Search*: Unchecked.
 - *Enter Manager Name or IP Address*: Used by the Backburner Server to locate and communicate with the manager. The keyword **localhost** should suffice, since it resolves to the IP address of the local workstation.
- 7 Verify the connection in the monitor window or server window.
- 8 On the Mac workstation, configure Smoke for Mac OS X to point to the Backburner Manager. In the Smoke Setup application, set the *Manager Hostname* field (in the Application Settings area) to the machine name or IP address of the Windows workstation where Cleaner XL and Backburner are installed.

For more information on:

- Detailed setup instructions and troubleshooting, refer to the *Using Autodesk Cleaner XL with Autodesk Effects and Finishing Applications* guide.
- Sending jobs to Cleaner XL, refer to the *Autodesk Smoke for Mac OS X Help*.
- Details on encoding using Cleaner XL, refer to the *Autodesk Cleaner XL User Guide*.

Burn

Burn is the rendering engine shared by all Visual Effects and Finishing applications, including Smoke for Mac OS X. When you submit rendering jobs to the Backburner network from Smoke for Mac OS X, it is Burn

that performs the rendering. There is no standalone deployment model for Burn. It is always deployed as a render farm.

NOTE While Burn is a Linux application, Smoke for Mac OS X is designed to operate seamlessly within the context of a Burn render farm.

For more information on:

- Guidelines for network rendering, see [Setting up a Render Farm](#) on page 27.
- Detailed instructions, refer to the *Burn Installation and User Guide*.

WiretapCentral

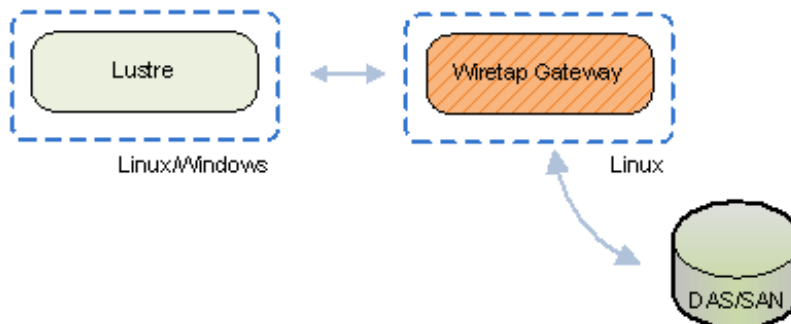
Visual Effects and Finishing applications, including Smoke for Mac OS X, use WiretapCentral to read and decode formats such as OpenEXR and Red (.r3d) files, to a Stone FS or Standard FS filesystem. WiretapCentral leverages Backburner to off-load transcoding. In a standalone installation, all the background processing components—Backburner Server, Backburner Manager, and the MIO Adapter—are installed on the same workstation as the Smoke for Mac OS X application.

This is the simplest configuration, since all components are automatically installed and set up by the Smoke for Mac OS X installer. However, since all media processing is performed locally, it can have a significant impact on system performance. The standalone configuration is best-suited for occasional media encoding and decoding. For more demanding uses, a render farm is the recommended configuration.

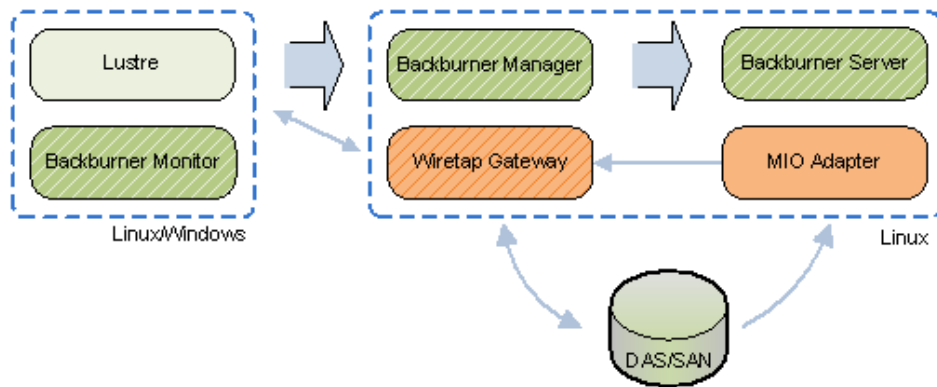
For detailed installation, setup and configuration instructions, refer to the *WiretapCentral and Wiretap Gateway Installation and Configuration Guide*.

Lustre

Lustre can work with streaming media—such as Red (.r3d) files—in two distinct ways. In the simplest configuration, Lustre reads material natively through the Wiretap Gateway. In this case, the Wiretap Gateway decodes the media on-the-fly, immediately returning proxy-quality images for use in Lustre. As shown in the following illustration, Backburner is not needed.



High-resolution background transcoding of streaming media formats is made possible by leveraging Backburner, as shown in the following illustration.



As in the native workflow, Lustre browses storage devices via the Wiretap Gateway. Here, however, transcoding jobs are sent directly from the Lustre UI to the Backburner Manager. The Backburner Manager sends the job to the Backburner Server, which invokes the Autodesk MIO Adapter processing engine to carry out the transcoding. The MIO Adapter reads from storage directly and writes its results to the Wiretap Gateway. The Backburner Monitor can optionally be used to monitor job progress.

For more information on:

- Installation instructions (Windows), refer to the *Autodesk Lustre Installation and Configuration Guide for Windows Workstations*.
- Installation instructions (Linux), refer to the *Autodesk Lustre Installation and Configuration Guide for Linux Workstations*.
- Guidelines for transcoding using a render farm, see [Lustre Background Transcoding](#) on page 45.

Maya

There is no standalone deployment model for Backburner with Maya. Backburner is not needed for local batch rendering in Maya. In Maya, Backburner is used for network rendering only.

See [Setting up a Render Farm](#) on page 27.

Maya Composite

For Maya Composite, the situation is identical to Maya. That is, there is no standalone deployment model for Backburner with Maya Composite. Backburner is not needed for local background rendering in Maya Composite. In Maya Composite, Backburner is used for network rendering only.

See [Setting up a Render Farm](#) on page 27.

Setting up a Render Farm

3

Topics in this chapter:

- [Overview](#) on page 27
- [Render Farm Deployment Options](#) on page 28
- [Isolating the Backburner Manager](#) on page 29
- [Setting up Monitoring](#) on page 30
- [Setting Up Render Nodes](#) on page 34
- [Handling Bulk Data](#) on page 35
- [3ds Max](#) on page 37
- [Visual Effects and Finishing](#) on page 39
- [Smoke for Mac OS X](#) on page 42
- [Lustre](#) on page 44
- [Maya](#) on page 46
- [Maya Composite / 3ds Max Composite](#) on page 51

Overview

Backburner can accommodate a number of different render farm deployment models and network topologies. Your goal may be to set up one or more render nodes for a single creative application; or, you may be planning to roll out a larger render farm to service numerous creative seats simultaneously. Whatever your goal, the guiding principles for establishing a render farm are similar.

First, you must determine a location for the Backburner Manager. The main factor affecting its location is the deployment model. A render farm dedicated to a single creative seat, and one to be shared by multiple seats, do not have the same needs. When planning a larger render farm, it is also important to consider the effect of network traffic upon the Backburner Manager.

Second, you must establish monitoring and control of the farm by installing and configuring the Backburner Monitor. The monitor comes in two versions, one running under Windows, and a browser-based version.

Your choice of which to use depends on your deployment model, traffic expectations, user management goals, and other factors.

NOTE WiretapCentral provides its own built-in Backburner monitor, designed for tracking the progress of transcoding jobs. See the *WiretapCentral User Guide*.

Finally, you must create the render nodes. This involves installing the processing engine and Backburner Server on each workstation to serve as a render node. For some creative applications, such as 3ds Max, creative seats themselves can be enlisted as render nodes. For others, such as the Visual Effects and Finishing applications, including Smoke for Mac OS X, render nodes are always separate workstations.

An additional—often critical—consideration is the ability of the network and storage to handle the flow of “bulk” data—that is, scene, texture or frame data. Each creative application has distinct bulk data needs. Taking them into consideration when designing the render farm will help to maximize its efficiency.

This chapter presents the design and configuration issues associated with a Backburner render farm. It begins with the general layout of a typical render farm—one not tied to any particular application. Next, it presents the steps to creating a render farm: isolating the Backburner Manager, setting up monitoring, and installing and configuring the servers. It then examines the more advanced subject of how to accommodate bulk data. The final sections provide instructions for setting up render farms for each of the creative applications leveraged by Backburner, with cross-references to detailed installation instructions, should they be required.

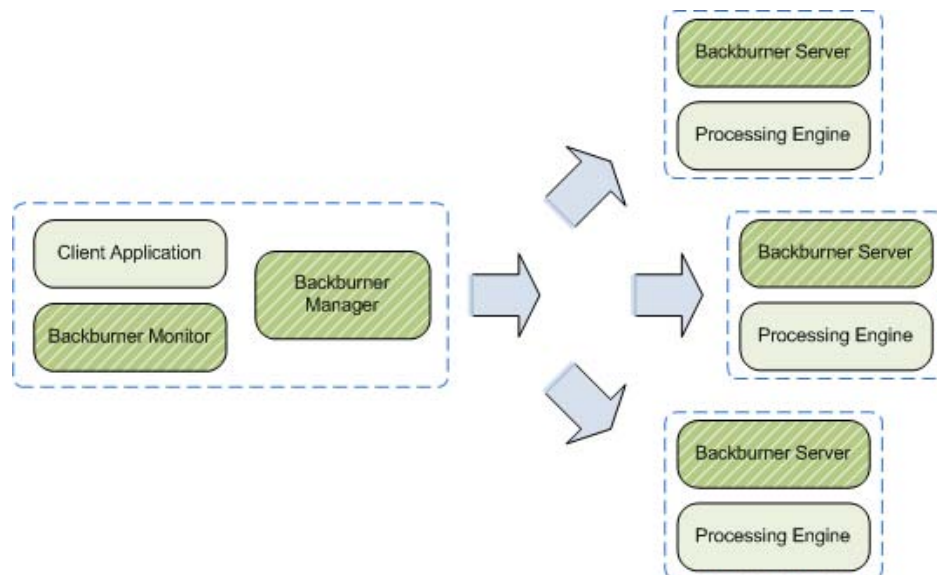
Render Farm Deployment Options

There are two basic forms of the render farm. The first is a render farm dedicated to a single creative seat. In the second, which is usually larger, render nodes are shared by multiple creative seats. This section presents an overview of each option.

Dedicated Render Farm

In a dedicated render farm, multiple render nodes service a single creative application. In this case, the creative seat serves as the “master” workstation that oversees and controls all jobs on the render farm. For this reason, both the Backburner Manager and Backburner Monitor are installed on the creative seat. Each render node contains the Backburner Server and appropriate processing engine.

The dedicated render farm is illustrated in the following figure.

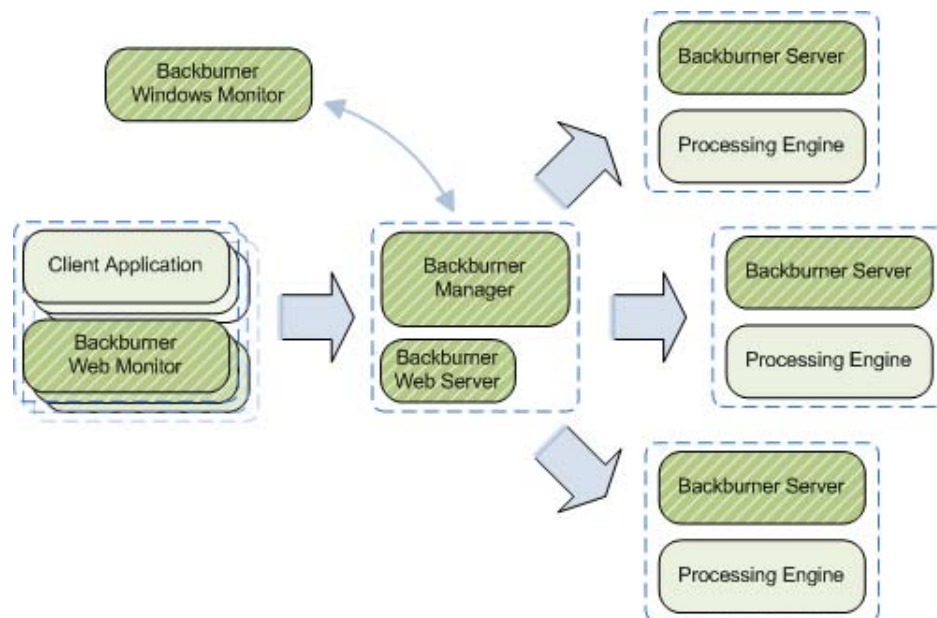


The render nodes in a dedicated render farm can include workstations used as creative seats, depending on the application. For 3ds Max, for example, you can take advantage of the unused processing power of creative workstations during off-peak hours, using Backburner Manager's scheduling abilities. This is not the case for Visual Effects and Finishing applications—Burn requires exclusive use of the render node.

Shared Render Farm

In a shared render farm, the render nodes service more than one seat of a creative application. In this case, it is recommended that you install the Backburner Manager on a workstation reserved exclusively for its use. The resource intensive Backburner Windows Monitor can reside on the same machine as the manager, designated as a “master” workstation (if a Windows machine is available). Each non-master creative application can make use of the less resource-intensive Backburner Web Monitor to monitor and control jobs. Note that to enable use of the Backburner Web Monitor, the Backburner Web Server must be installed on the master workstation.

This is illustrated in the following diagram.



Isolating the Backburner Manager

The Backburner Manager coordinates jobs submitted by the creative applications and delegates them to the servers on the render farm. Its role as a central point of communication is illustrated in the following diagram, showing a shared render farm consisting of three seats of the creative application and five render nodes.

Manager. This interaction includes pausing, restarting, and reassigning jobs and tasks to different render nodes. It also includes creating and managing render node groups. During the initial setup of a new render farm, monitoring is one of the ways you can quickly verify that the servers are up and running.

The initial goal when setting up a render farm is to install the Backburner Monitor and establish a connection to the Backburner Manager. If your network will be making use of the Windows version of the Backburner Manager, it can be helpful to install the Backburner Monitor on the same workstation. This allows you to easily start the manager and monitor and work out any connection details. As you set up each server, you can then observe as they come online, from one central location. On a large render farm, this could later be designated as the integrated administrator workstation.

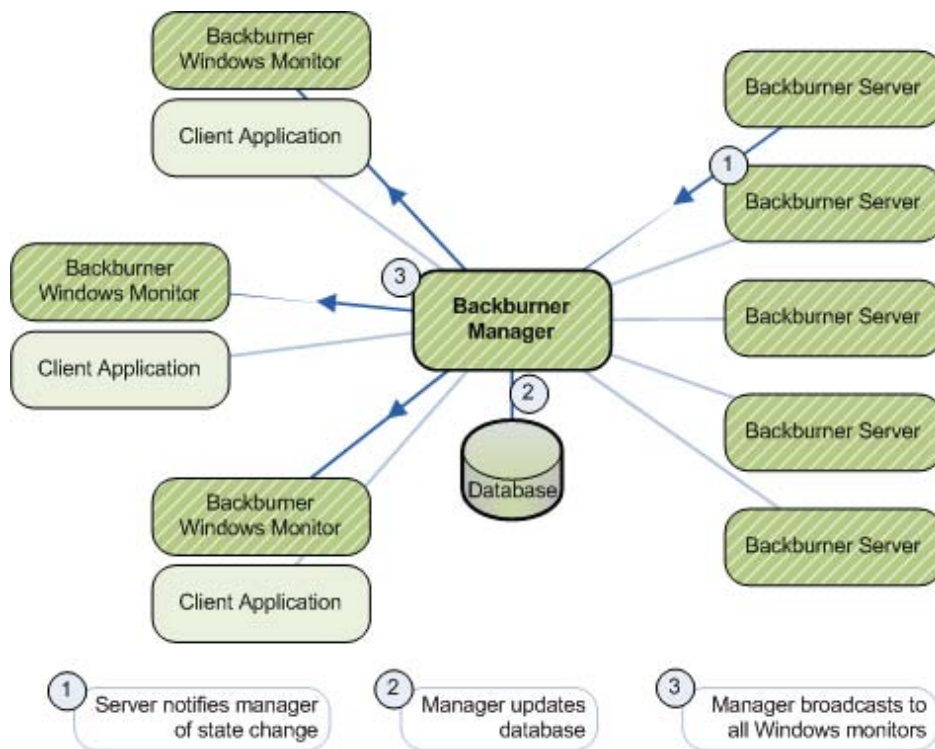
Windows Monitor versus Web Monitor

The Backburner network can be monitored via a Windows-based or browser-based monitor. The Windows monitor is well-suited for a small render farm with a single creative workstation, or the administrator workstation on a larger render farm. The browser-based monitor is designed for the non-administrator workstations on a larger render farm. The following table summarizes the major differences.

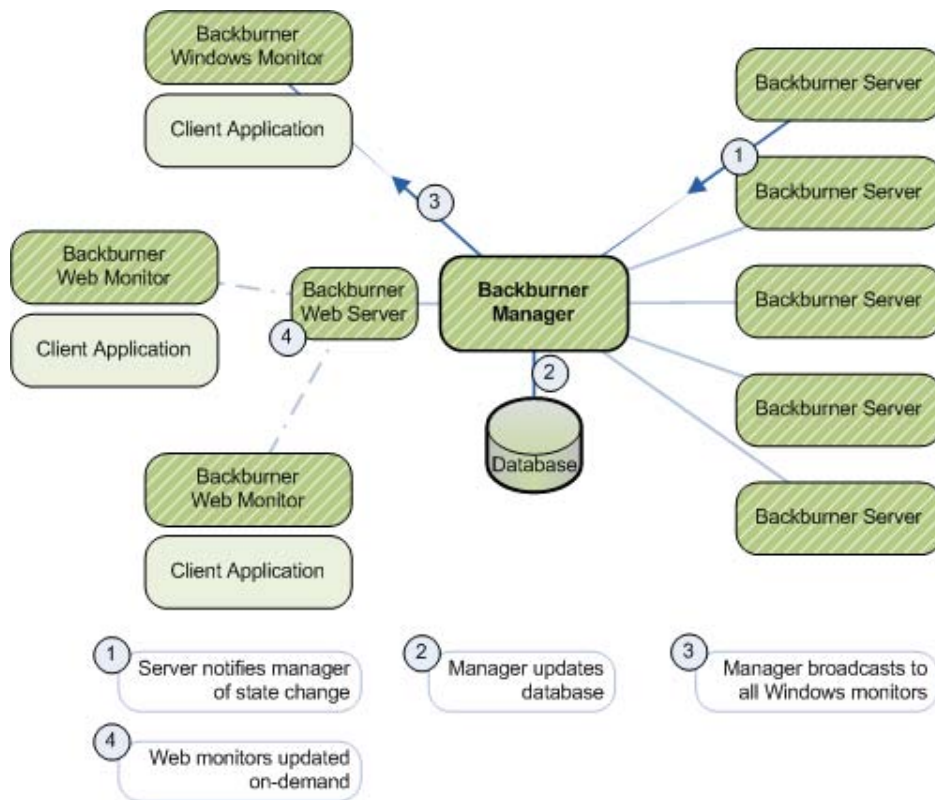
	Backburner Windows Monitor	Backburner Web Monitor
Operating System	■ Windows OS	■ OS independent (browser-based Adobe Flash Player compatible)
Monitoring Mechanism	■ Automatic, via broadcasts from Backburner Manager	■ On demand via end-user requests
Queue Control	<ul style="list-style-type: none"> ■ Obtained by first monitor to connect to the manager ■ Also, by request made from a non-controlling monitor 	■ On demand when changes are requested
User Management	■ Not applicable. All users operate with admin privileges	■ User account required to log in to web monitor
Suitability	<ul style="list-style-type: none"> ■ Small render farm with single creative workstation ■ Administrator workstation on larger render farm 	■ Non-administrator workstations on larger render farm

NOTE This guide documents the Backburner Windows Monitor and Backburner Web Monitor. For information on the monitoring capabilities built into some applications, see the application-specific user guide.

To understand the differences between the two monitors, it is helpful to examine the network and other activity associated with the Backburner Manager in some detail. The Backburner Manager maintains a database, which it updates with every change of state of the render nodes. It then broadcasts the changes to every workstation running a Backburner Windows Monitor, whether the end-user is actively viewing it or not. The result is a steady stream of data from the manager to the Windows monitors, as illustrated in the following diagram.



A more efficient model emphasizes web monitors over Windows monitors. Web-based monitors gracefully handle the increased traffic on busier render farms, since they are not updated with each change to the manager's database. Rather, a web-based monitor receives updates on demand, when the end-user explicitly refreshes the view. In addition, all traffic between the manager and the web monitor is channeled through a web server, which is well-equipped to handle increases in traffic. This scenario is illustrated in the following diagram (some connectivity lines have been omitted, for clarity).



The number of Windows-based monitors a system can sustain depends on numerous factors. These include the amount of Backburner Server activity, the type and size of jobs sent to the manager, the size of the objects pushed through the manager, etc. In most cases, a single instance of the Backburner Windows Monitor will suffice. This is the recommended configuration, when a Windows machine is available.

Backburner Web Monitor and User Management

The Backburner Web Monitor operates by connecting to a Backburner Web Server installed on the machine where the Backburner Manager is running. Upon connection, the Backburner Web Server requires a user name and password, information it uses to determine the kinds of operations that can be performed. If you will be setting up for monitoring using the Backburner Web Monitor, you must also create user accounts on the Backburner Web Server.

By default, end-users have complete control over the jobs that they submit to Backburner. They can perform all job operations upon them, such as restarting, deleting, and archiving. In order to control *all* jobs on the Backburner network, you must log on to the server with administrator privileges.

Generally speaking, the name used to log in to the workstation is associated with all jobs submitted to Backburner from that workstation. However, some applications pass on the account name used to start the *application* instead. In Smoke, for example, if the artist starts the application as user *smoke*, *smoke* owns the jobs. It is therefore necessary to create accounts on the web server with the same names. By matching the log in or application user names with the web server user names, you ensure the artist has control over the jobs he or she submits.

For more information on Installing and configuring the Backburner Monitor, see:

- [Backburner Web Monitor – Linux](#) on page 71.
- [Backburner Components – Mac](#) on page 75.

For more information on ensuring web monitor users have the permissions they need to manage their jobs, see:

- [Setting Up Access to the Backburner Web Monitor](#) on page 73 (Linux).
- [Setting Up Access to the Backburner Web Monitor](#) on page 78 (Mac).

For more information on configuring the creative application to point to the Backburner Windows Manager, refer to the application-specific user guide.

Setting Up Render Nodes

The final step in setting up a render farm is establishing the render nodes. To this end, the Backburner Server must reside on each workstation you intend to use as a render node. On Windows workstations, the server can run as an application (foreground) or as a service (background). On Linux/Mac OS, it runs as a daemon.

The server is delivered and installed with the application-specific renderer. For example, with 3ds Max, it is installed at the same time as 3ds Max. For Visual Effects and Finishing applications, it is installed with Burn. Similarly, Smoke for Mac OS X makes use of a Linux-based Burn render farm for network rendering. Depending on the creative application, it may be possible to explicitly select the Backburner Server for installation, it may be installed automatically as part of a “render node” installation option, or it may be installed by default, transparently, or otherwise.

NOTE Under normal circumstances there should be no need to install the server separately. The exception is when using Backburner's *cmdjob* utility to run custom jobs on remote workstations. In this case, Backburner operates as a pure queue manager, and you can install the Backburner Server without any associated rendering or processing application. See the *Autodesk Backburner User Guide*.

Once a server is installed, the next step is establishing a connection to the Backburner Manager, and optionally observing the connection by way of the Backburner Monitor. You can then set up multiple servers to service one client, or create a pool of renderers for multiple clients. In each case, the basic approach is the same: install the Backburner Server with the rendering software, then configure the Backburner Server to connect to the Backburner Manager.

NOTE Whatever deployment option you adopt, render nodes must have the same directory structures as the workstation (or workstations) originating the requests for network rendering. The Backburner subdirectory should be in the same place on each render node, for example. If it is on the C: drive on the creative workstation, it cannot be installed on the D: drive on the render nodes. This path is included as part of the job metadata that is sent to the render node. If the render node cannot find the directories, the job will fail. For similar reasons, any media server or output server should be mounted in the same manner for each render node.

For more information on:

- Installing the Backburner Server for background I/O, refer to the application-specific installation guide.
- Installing the Backburner Server for Burn, refer to the *Autodesk Burn Installation and User Guide*.
- Configuring the Backburner Server, see [Starting and Configuring the Backburner Server](#) on page 60.
- Operating the Backburner Server with Combustion, refer to the *Autodesk Combustion User Guide for Windows*.
- Operating Backburner Server with other applications, refer to the *Autodesk Backburner User Guide*.

Handling Bulk Data

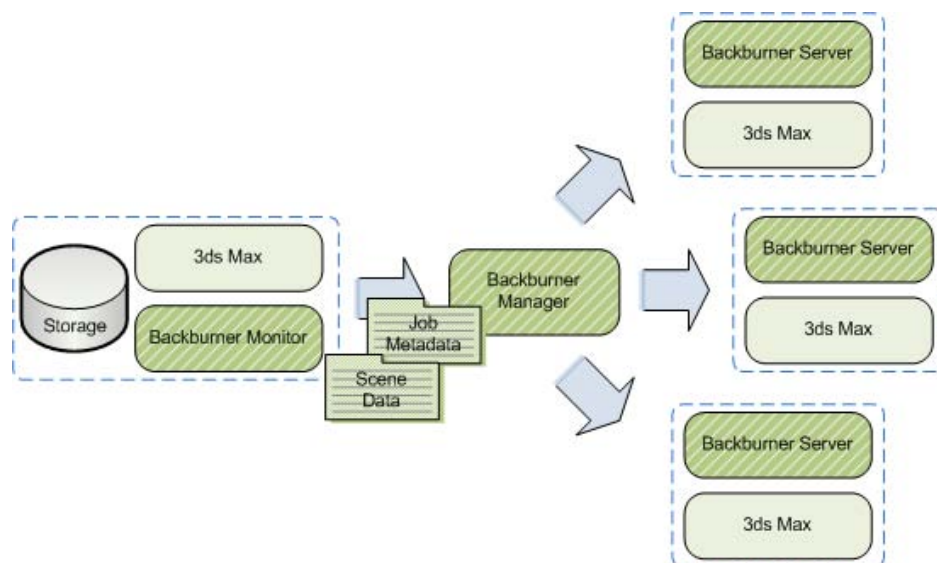
There are two distinct data flows on a Backburner render farm: job metadata and “bulk” data. Job metadata consists of the task descriptions sent to the render nodes and status data received from them. Bulk data refers to the heavier data required by the nodes to carry out their tasks. For 3ds Max and Composite, these are scenes, textures, motion capture, etc. For Maya, the heavy data is stored in the directories (or folders) associated with projects, which contain scenes, images, etc. For Visual Effects and Finishing applications, this is the media to be processed. Each creative application type has different bulk data needs, in terms of storage and network capacity. It is the system designer or administrator's job to map out the flow of this data and to plan storage and network capacity accordingly.

3ds Max

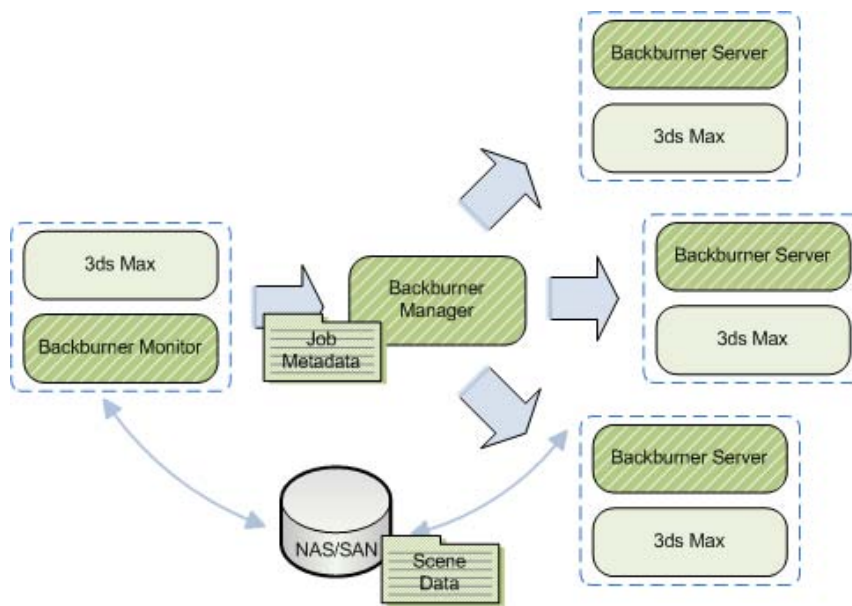
For 3ds Max and 3ds Max Design, it is important to avoid allowing bulk data to reside on locally attached storage. To illustrate, consider a small render farm that has been set up for a single creative seat of 3ds Max. In single-artist studios, it may be common to store scene data on the creative workstation's direct attached storage. In this case, a bottleneck will occur at the creative seat's direct attached storage when each render node tries to obtain the needed scene data. In addition, the heavy I/O will result in a general slow-down on the creative workstation submitting the job.

Note that the situation worsens if the texture map is included in the scene file. In this case, it is passed to the manager as part of the job, possibly creating an I/O bottleneck on that machine, if there is insufficient memory on the workstation and the manager needs to store it locally. In addition, the Backburner Manager is not designed for sending bulk data across the network. Its job is to send job descriptions and tasks to the render nodes and to report on job status to the monitors.

This scenario is represented by the following diagram.



The solution is to ensure the creative applications are set up so all bulk data does not reside locally, but on a NAS (Network Attached Storage) or SAN (Storage Area Network), where it is equally accessible to the creative workstation and all render nodes. The preferred solution is shown in the following diagram. Note that in this case, care must be taken to balance the I/O rate of the NAS with the network speed.

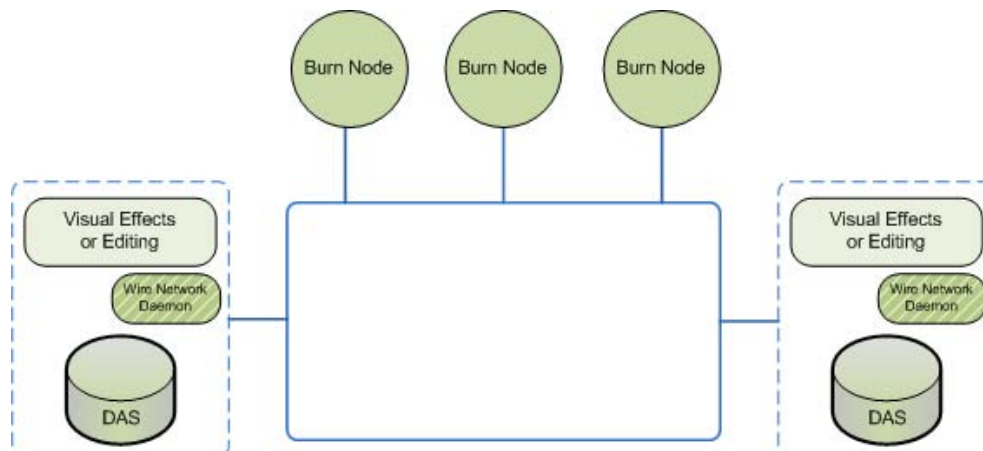


Visual Effects and Finishing

During a render, each Burn node requires access to the full-resolution version of the frames requiring processing. This results in a great deal of data moving across the Backburner network, particularly for projects involving higher resolutions, such as HD, 2K, or 4K film frames. As a result, for these applications, the constraining factor is network bandwidth. Whether your frames are stored on a direct attached storage (DAS) system, Network Attached Storage (NAS), or a Storage Area Network (SAN), similar considerations apply.

NOTE The following discussion applies also to Smoke for Mac OS X.

The following diagram serves to illustrate the discussion.



As you add new render nodes to the farm, the limitations of a DAS system are felt early on. On the one hand, Visual Effects and Finishing applications have built-in bandwidth protection for access to this kind of storage. On the other hand, the Wire network daemon provides the Burn nodes with independent access to obtain media for processing. Thus, there may not be contention for the creative workstation's CPU, but competition for network resources can arise.

For example, consider a creative seat that has submitted a job for processing on the render farm. As the Burn nodes process the job, they make repeated access to the DAS of the originating workstation. If the workstation

then tries to access media on a remote workstation, for example, performance will degrade. The same problem arises if a remote workstation attempts to retrieve media from the originating workstation.

One solution can be to centralize media on a NAS or SAN. The design and establishment of a render farm making use of SAN storage is complex, and beyond the scope of this document. Autodesk Professional Services is available and experienced in the design of storage solutions to meet the needs of the facility.

3ds Max

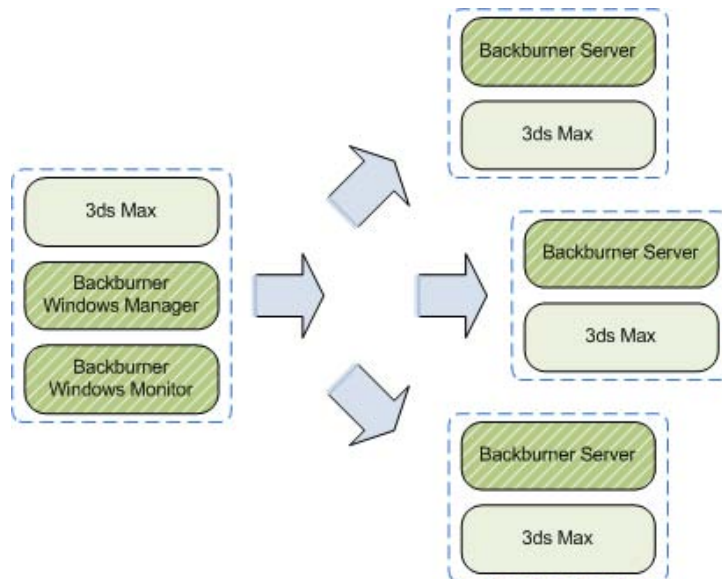
To create a render farm for 3ds Max or 3ds Max Design, you install the application (including the Backburner option) on each render node. There are two main scenarios. In the first scenario, a render farm is created to service a single 3ds Max creative seat. In the second, multiple creative seats are serviced by the render farm.

A facility with both 3ds Max and 3ds Max Design workstations can share the same render farm. In this case, simply install one Backburner Server, but both applications, on each render node. This is necessary, since each application can submit jobs to its own processing engine type only.

The above requirement does not necessarily limit your ability to render on a farm that consists of just one processing engine type. Scene (.max) files from one application are easily loaded into the other. Once loaded, a scene can be submitted for rendering on the “host” render farm. For example, you can load a 3ds Max Design scene into 3ds Max, then submit the job to a render farm consisting of 3ds Max render nodes. Ensure any needed plug-ins are installed on the render nodes.

3ds Max Dedicated Render Farm

The following diagram illustrates a typical 3ds Max render farm. In this case, a render farm has been created for a single seat of 3ds Max.



To set up a dedicated render farm for 3ds Max:

NOTE This procedure assumes 3ds Max, including all Backburner components, is already installed on the creative workstation. If this is not the case, please refer to the 3ds Max documentation first.

- 1 Start the Backburner Manager on the creative workstation. From the Start menu, choose Programs, Autodesk, Backburner, and then Manager.

- 2 Start the Backburner Monitor on the creative workstation. From the Start menu, choose Programs, Autodesk, Backburner, and then Monitor.
- 3 Connect the monitor to the manager. From the Manager menu, choose Connect. Deselect the Automatic Search checkbox, and enter the following for the manager name/IP address:
localhost
- 4 Verify the connection. From the Manager menu, choose Properties.
Manager details appear in a dialog. If no connection was made, this option is greyed-out.
- 5 On each render node, install the Backburner components using the 3ds Max installation disk.
- 6 On each render node, start the Backburner Server. From the Start menu, choose Programs, Autodesk, Backburner, and then Server.
The General Properties dialog appears. If it does not, from the Edit menu, choose General Settings.
- 7 Verify the following items are set correctly in the General Properties dialog:
 - *Server Name or IP Address*: Used by the Backburner Manager to identify and communicate with the server. This field should contain the machine name or IP address of the current workstation.
 - *Automatic Search*: Unchecked. Employing automatic searching is not recommended, since it makes extensive use of network resources and can take some time.
 - *Enter Manager Name or IP Address*: Used by the Backburner Server to locate and communicate with the manager. Note that this cannot be **localhost**—as used above—since the manager is not local to any of the servers.
- 8 Verify the connection in the monitor window or server window.
- 9 Optional: Configure the Backburner Server on each render node as a Windows's service, so it starts with the workstation's operating system and runs in the background, as described in [Setting Up Backburner Server as a Windows Service](#) on page 63.

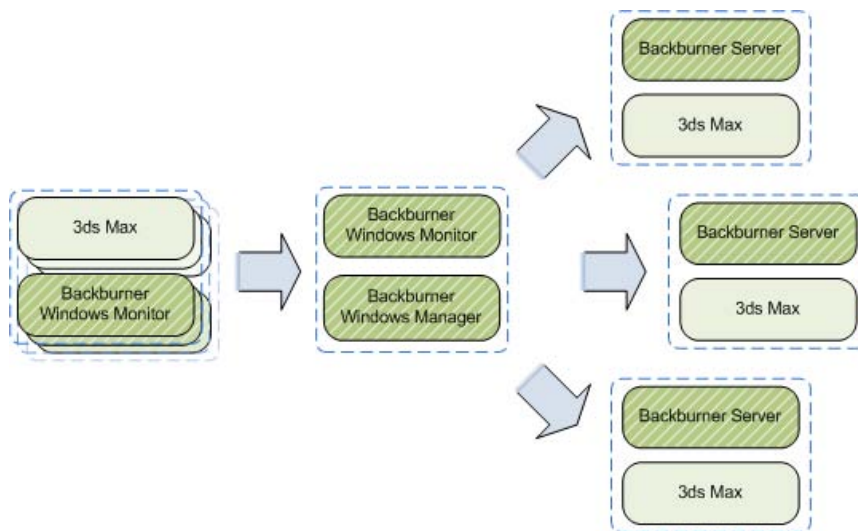
3ds Max Shared Render Farm

When the render farm is designed to service more than one creative seat of 3ds Max, the Backburner Manager should reside on a dedicated workstation, designated as the master workstation. In addition, the Windows version of the Backburner Monitor can be installed, either on the same workstation as the Backburner Manager, or off-loaded to a separate workstation.

If you decide to install more than one or two instances of the Backburner Windows Monitor, keep network traffic to a minimum by disconnecting them from the Backburner Manager when they are not in use. Otherwise, network slowdowns can occur. See [Setting up Monitoring](#) on page 30.

NOTE Idle 3ds Max workstations can easily be enlisted as render nodes, and scheduled for off-peak or weekend use, if necessary.

This scenario is illustrated in the following diagram.



NOTE The procedures in this section assume you have experience setting up a dedicated render farm for a single creative seat of 3ds Max. This allows the procedures to focus on the major steps only.

On the creative workstation:

- Install 3ds Max.

On the master workstation:

- 1 Install the Backburner Components. The Backburner Manager and Backburner Monitor are required.
See [Installing the Backburner Components on Windows](#) on page 56.
- 2 Connect the Backburner Windows Monitor to the Backburner Manager.
See [Starting and Configuring the Backburner Windows Monitor](#) on page 60.

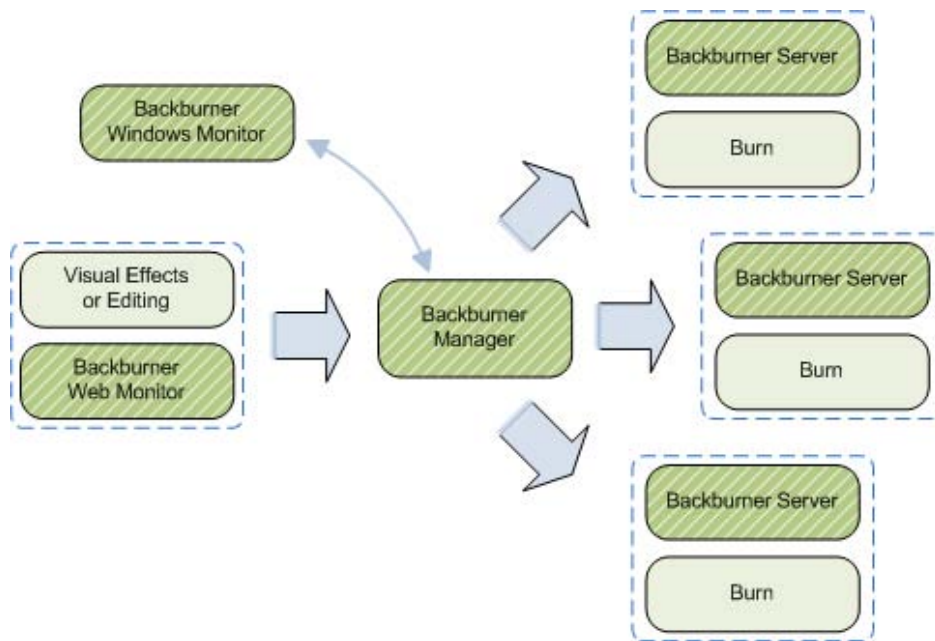
On each render node:

- 1 Install 3ds Max and the Backburner Components. The Backburner Server is required.
See [Installing the Backburner Components on Windows](#) on page 56.
- 2 Connect the Backburner Server to the Backburner Manager running on the master workstation.
See [Starting and Configuring the Backburner Server](#) on page 60.
- 3 Optional: Configure the server to start as a Windows service.
See [Setting Up Backburner Server as a Windows Service](#) on page 63.

Visual Effects and Finishing

To create a render farm for a Visual Effects or Finishing application, you install Backburner Server and Burn on each workstation designated to serve as a render node. As indicated in the following diagram, the Backburner Manager should reside on a dedicated workstation. In addition, the Windows version of the Backburner Monitor can be installed, either on the same workstation as the Backburner Manager (if that is a Windows machine), or off-loaded to a separate workstation. Whether you are setting up for a single creative seat, or multiple seats, the process is virtually the same.

NOTE Smoke for Mac OS X is discussed in its own section. See [Smoke for Mac OS X](#) on page 42.



NOTE As noted earlier, Visual Effects and Finishing applications can leverage Backburner in four distinct ways: for background I/O, encoding using Cleaner XL, network rendering using Burn, and media conversion using WiretapCentral. In a facility where more than one of these systems is operating, it can be efficient to allow the systems to share the same Backburner Manager. In this way, the user with administrator privileges can monitor and control each of the 'sub-systems' from a central location.

On the dedicated workstation:

- 1 Install the Backburner Components.
The Backburner Manager is required, while the Backburner Monitor is optional, depending upon the design of your render farm.
See:
 - [Installing the Backburner Components on Windows](#) on page 56.
 - [Installing the Backburner Components on Linux](#) on page 68.
- 2 To enable use of the Backburner Web Monitor from the creative workstation(s), install the Apache HTTP server (Linux).
- 3 Install and configure the Backburner Web Server, and verify that the Backburner Web Monitor is accessible to the default administrator user.
- 4 Set up end-user access to the Backburner Web Monitor.
- 5 Optional: Assign administrator privileges to selected end-users.
For steps 2–5, see:
[Backburner Web Monitor – Linux](#) on page 71
- 6 Optional: Connect the Backburner Windows Monitor to the Backburner Manager.
See [Backburner Components – Windows](#) on page 55.

On a Windows workstation used for monitoring (optional):

- 1 Install the Backburner components. The Backburner Windows Monitor is required.

See [Installing the Backburner Components on Windows](#) on page 56.

- 2 Connect the Backburner Windows Monitor to the Backburner Manager.
See [Starting and Configuring the Backburner Windows Monitor](#) on page 60.

On each render node:

- 1 Install Burn and the Backburner components. The Backburner Server is required.
See the *Autodesk Burn User Guide*.
- 2 Connect the Backburner Server to the Backburner Manager running on the dedicated workstation.
See the *Autodesk Burn User Guide*.
- 3 Restart the Backburner Server so it picks up the new settings:
`/etc/init.d/backburner restart`

On the creative workstation(s):

- 1 Install the Visual Effects and Finishing application.
See the *Autodesk Visual Effects and Finishing Installation and Configuration Guide*.
- 2 Configure the Visual Effects and Finishing application for Burn.
See the *Autodesk Burn Installation and User Guide*.
- 3 Monitor the Backburner network using the Backburner Web Monitor.
See the *Autodesk Backburner User Guide*.

Visual Effects and Finishing with Cleaner XL

Cleaner XL encodes the entire job on the same workstation. Thus, it is important to consider the workflow it will be a part of before creating a render farm. When encoding a single job, for example, the presence of additional workstations does not improve efficiency. However, when encoding multiple jobs at the same time, additional jobs are routed to different available workstations. As a result, it is recommended to set up Cleaner XL in a stand-alone configuration for Visual Effects and Finishing applications, as presented in [Configuring a Standalone Workstation](#) on page 11.

Visual Effects and Finishing with WiretapCentral

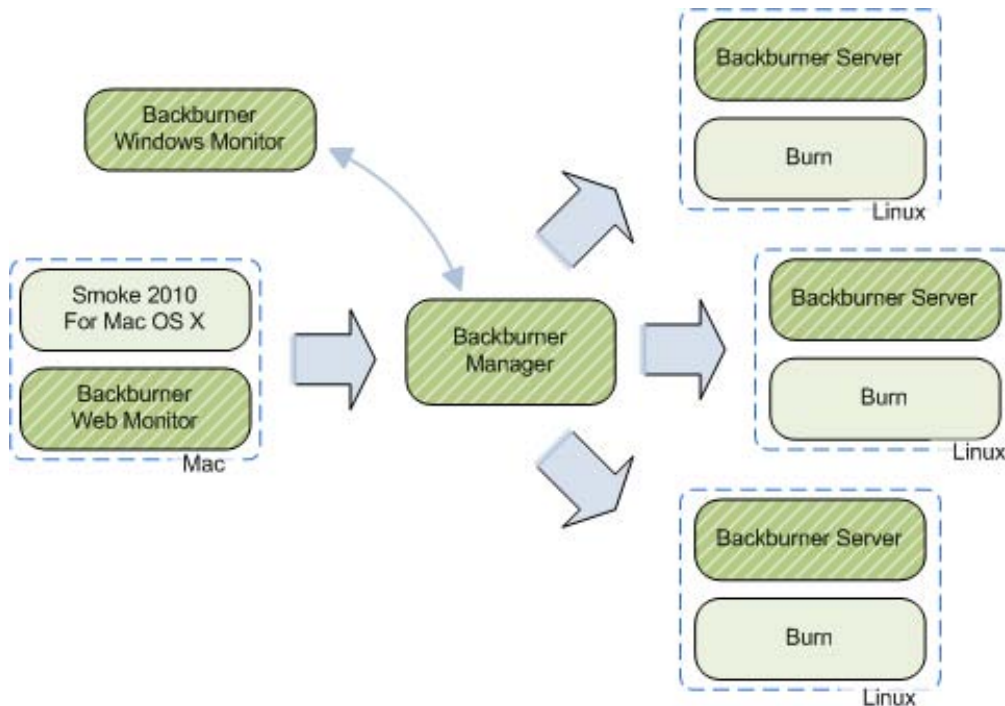
In a render farm configuration, WiretapCentral leverages Backburner for scalability of streaming media transcoding. As in a Burn render farm, WiretapCentral submits jobs to Backburner, which are sent for processing on render nodes. However, for transcoding of streaming media, each render node must be equipped with the Backburner Server and MIO Adapter (supplied with Burn).

As a result, the same design and planning considerations for a Burn render farm apply here. In fact, WiretapCentral can easily make use of an already-installed Burn render farm, since the needed MIO Adapter can be automatically installed alongside Burn. For older Burn render farms, you can add the MIO Adapter to the render node in a separate installation operation.

For more information, refer to the *Autodesk WiretapCentral and Wiretap Gateway Installation and Configuration Guide*.

Smoke for Mac OS X

To create a render farm for Smoke for Mac OS X, you install Backburner Server and Burn on each workstation designated to serve as a render node. As indicated in the following diagram, the Backburner Manager should reside on a dedicated workstation. In addition, the Windows version of the Backburner Monitor can be installed, either on the same workstation as the Backburner Manager, or off-loaded to a separate Windows workstation. Whether you are setting up for a single creative seat, or multiple seats, the process is virtually the same.



NOTE As noted earlier, Smoke for Mac OS X can leverage Backburner in four distinct ways: for background I/O, encoding using Cleaner XL, network rendering using Burn, and media conversion using WiretapCentral. In a facility where more than one of these systems is operating, it can be efficient to allow the systems to share the same Backburner Manager. In this way, the user with administrator privileges can monitor and control each of the 'sub-systems' from a central location.

On the dedicated workstation (Mac):

- 1 Install the Backburner Components. The Backburner Manager is required.
See [Installing the Backburner Components on Mac OS X](#) on page 76.
- 2 To enable use of the Backburner Web Monitor from the creative workstation(s), the Apache HTTP server must be installed.

NOTE The Apache HTTP server is the default web server under Mac OS X. There should be no need to install it independently.

- 3 Install and configure the Backburner Web Server, and verify that the Backburner Web Monitor is accessible to the default administrator user.
- 4 Set up end-user access to the Backburner Web Monitor.
- 5 Optional: Assign administrator privileges to selected end-users.
For steps 3–5, see:

[Setting Up Access to the Backburner Web Monitor](#) on page 78.

[Assigning Backburner Web Monitor Administrator Privileges](#) on page 78.

On a Windows workstation used for monitoring (optional):

- 1 Install the Backburner components. The Backburner Windows Monitor is required.
See [Installing the Backburner Components on Windows](#) on page 56.
- 2 Connect the Backburner Windows Monitor to the Backburner Manager.
See [Starting and Configuring the Backburner Windows Monitor](#) on page 60.

On each render node (Linux):

- 1 Install Burn and the Backburner components. The Backburner Server is required.
See the *Autodesk Burn User Guide*.
- 2 Connect the Backburner Server to the Backburner Manager running on the dedicated workstation.
See the *Autodesk Burn User Guide*.
- 3 Restart the Backburner Server so it picks up the new settings:
`/etc/init.d/backburner restart`

On the creative workstation(s):

- 1 Install Smoke for Mac OS X.
See the *Autodesk Smoke for Mac OS X Installation and Configuration Guide*.
- 2 Configure Smoke for Mac OS X for Burn.
See the *Autodesk Burn Installation and User Guide*.
- 3 Monitor the Backburner network using the Backburner Web Monitor.
See the *Autodesk Backburner User Guide*.

Smoke for Mac OS X with Cleaner XL

Cleaner XL encodes the entire job on the same workstation. Thus, it is important to consider the workflow it will be a part of before creating a render farm. When encoding a single job, for example, the presence of additional workstations does not improve efficiency. However, when encoding multiple jobs at the same time, additional jobs are routed to different available workstations. As a result, it is recommended to set up Cleaner XL in a stand-alone configuration for Smoke for Mac OS X, as presented in [Configuring a Standalone Workstation](#) on page 11.

Smoke for Mac OS X with WiretapCentral

In a render farm configuration, WiretapCentral leverages Backburner for scalability of streaming media transcoding. As in a Burn render farm, WiretapCentral submits jobs to Backburner, which are sent for processing on render nodes. However, for transcoding of streaming media, each render node must be equipped with the Backburner Server and MIO Adapter (supplied with Burn).

As a result, the same design and planning considerations for a Burn render farm apply here. In fact, WiretapCentral can easily make use of an already-installed Burn render farm, since the needed MIO Adapter can be automatically installed alongside Burn. For older Burn render farms, you can add the MIO Adapter to the render node in a separate installation operation.

For more information refer to the *Autodesk WiretapCentral and Wiretap Gateway Installation and Configuration Guide*.

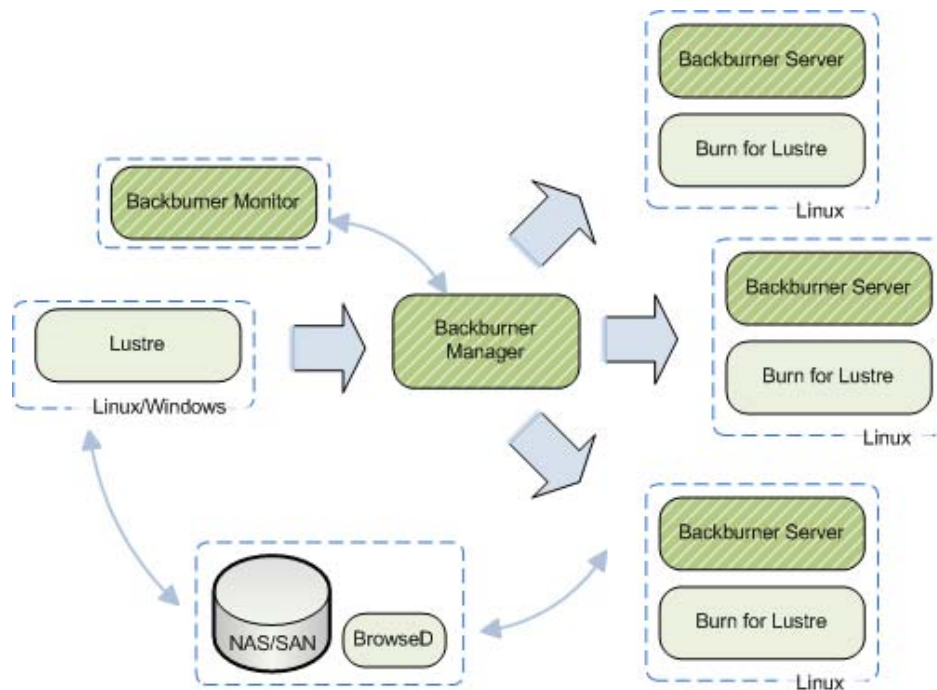
Lustre

There are two render farm possibilities for Lustre. The first is for network rendering of all shots in the timeline using Burn for Lustre. This frees up Lustre stations for interactive colour grading. The second is for background transcoding of streaming media, such as Red Raw (.r3d) files. In this case, a Backburner render farm off-loads the work associated with transcoding and ingesting complex and CPU-intensive streaming media formats.

Lustre Background Rendering

To set up background rendering for Lustre, create a Backburner network with render nodes consisting of Linux workstations running the Backburner Server and Burn for Lustre. To view and manage jobs on the network, you use Backburner Monitor. Off-loading rendering activities to Backburner frees Lustre stations for interactive color grading, and speeds up background rendering by splitting jobs amongst the render nodes.

The following diagram shows a typical Backburner network for Lustre. The Backburner Manager is shown residing on a dedicated workstation (recommended); however, it can also run on a Lustre workstation. Similarly, there are different installation options for the Backburner Monitor. Lustre's BrowseD file server is used to provide the render nodes access to storage, shown here as a NAS or SAN.



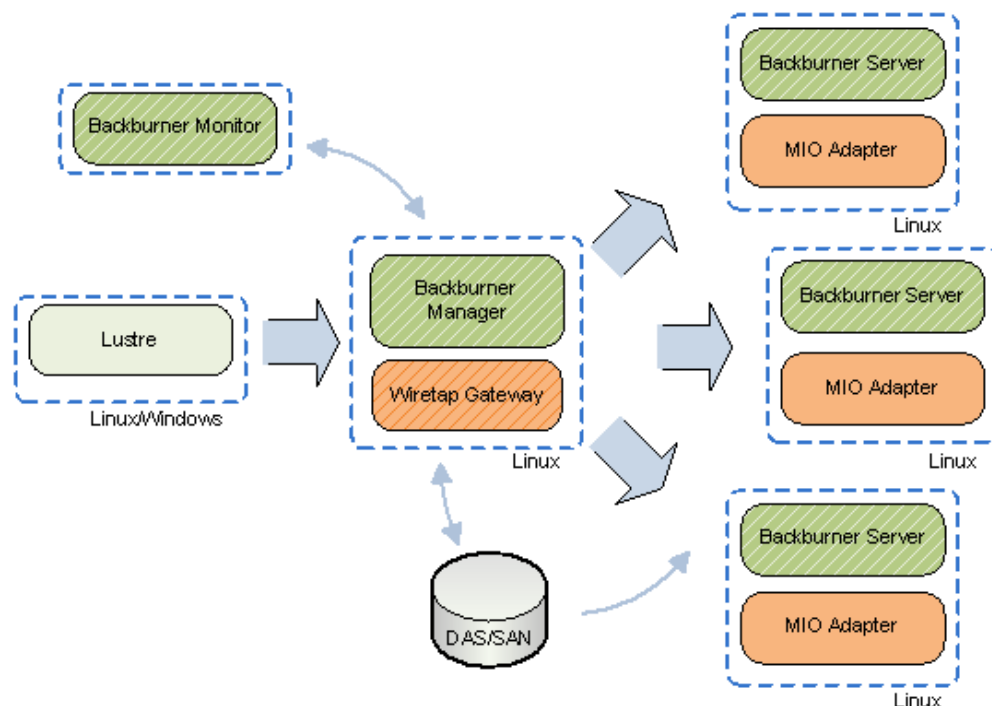
For more information on:

- Setting up Lustre for network rendering, refer to the *Autodesk Lustre Installation and Configuration Guide for Windows Workstations*.
- Installing Backburner components (Windows) see [Installing the Backburner Components on Windows](#) on page 56.

- Configuring Backburner Manager (Window) see [Starting and Configuring the Backburner Manager](#) on page 57.
- Setting up Backburner Manager as a Windows service, see [Running the Backburner Manager as a Windows Service](#) on page 59.
- Configuring the Backburner Monitor to detect the Backburner Manager, see [Starting and Configuring the Backburner Windows Monitor](#) on page 60.
- Installing Backburner components (Linux), see [Installing the Backburner Components on Linux](#) on page 68.
- Configuring Backburner Manager (Linux), see [Starting and Configuring Backburner Manager](#) on page 68.
- Installing and configuring Backburner Web Monitor (Linux), see [Backburner Web Monitor – Linux](#) on page 71.

Lustre Background Transcoding

To set up a render farm for background transcoding of streaming media formats, create a Backburner network with render nodes consisting of Linux workstations running the Backburner Server and MIO Adapter, as shown in the following illustration.



Lustre browses storage devices via the Wiretap Gateway, and submits transcoding jobs directly from its UI to the Backburner Manager. The Backburner Manager distributes the jobs to its render nodes, where the Backburner Server invokes the MIO Adapter processing engine to carry out the transcoding. The MIO Adapter reads from storage directly and writes its results to the Wiretap Gateway.

The Backburner Manager can reside on the Lustre workstation, or a dedicated workstation (recommended). Similarly, there are different installation options for the Wiretap Gateway and Backburner Monitor. You can optionally monitor job progress using the Backburner Monitor.

For more information on:

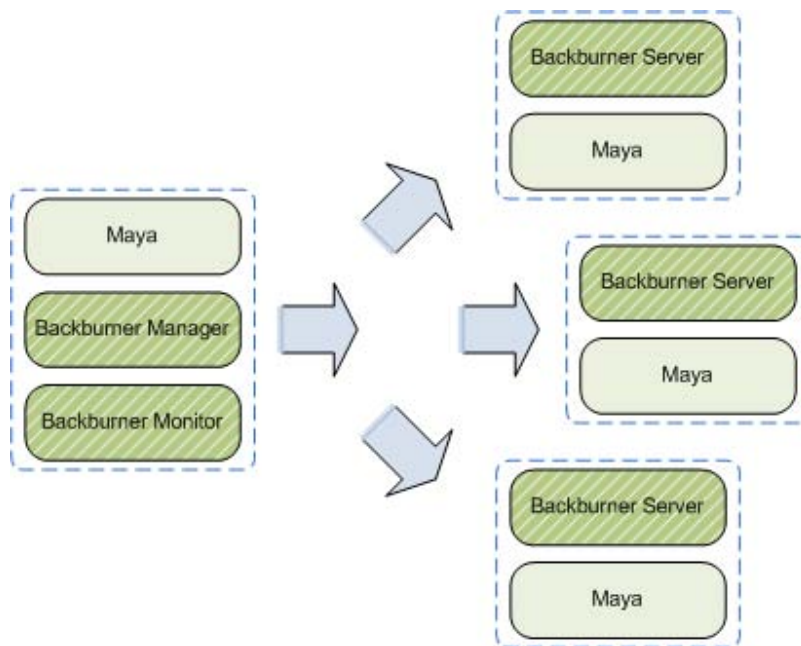
- General considerations when setting up a render farm, see the procedures in [Visual Effects and Finishing](#) on page 39.
- Installation instructions (Windows), refer to the *Autodesk Lustre Installation and Configuration Guide for Windows Workstations*.
- Installation instructions (Linux), refer to the *Autodesk Lustre Installation and Configuration Guide for Linux Workstations*.

Maya

To create a render farm for Maya, you install Maya (including the Backburner option) on each render node. There are two main scenarios. In the first scenario, a render farm is created to service a single Maya creative seat. In the second, multiple creative seats are serviced by the render farm.

Maya Dedicated Render Farm

The following diagram illustrates a typical Maya render farm. In this case, a render farm has been created for a single seat of Maya.



For more information on:

- Installation instructions (Windows/Linux/Mac), refer to the *Maya Installation and Licensing Guide*.
- Setup and configuration (Windows), see the procedures in [3ds Max Dedicated Render Farm](#) on page 37.

On the creative workstation (Linux/Mac):

- 1 Install Maya and the Backburner components (Backburner Manager and Backburner Web Server required). See the *Maya Installation and Licensing Guide*.

- 2 Enable use of the Backburner Web Monitor on the creative workstation by installing the Apache HTTP server (not shown in diagram).

NOTE The Apache HTTP server is the default web server under Mac OS X. There should be no need to install it independently on that platform.

- 3 Configure the Backburner Web Server, and verify that the Backburner Web Monitor is accessible to the default administrator user (not shown in diagram).
- 4 Set up end-user access to the Backburner Web Monitor.
- 5 Optional: Assign administrator privileges to selected end-users.
For steps 2–5, see:
 - [Backburner Web Monitor – Linux](#) on page 71.
 - [Installing the Backburner Components on Mac OS X](#) on page 76.

On each render node (Linux):

- 1 Install Maya and the Backburner components (Backburner Server is required).
See the *Maya Installation and Licensing Guide*.
- 2 Ensure the Backburner Server is installed and set to run as a daemon by typing the following:
chkconfig --list | grep -i backburner
Output similar to the following should appear:
backburner_server 0:off 1:off 2:on 3:on 4:on 5:on 6:off
Levels “3”, “4”, and “5” must be “on” for the daemon to start automatically. If this is not the case, reinstall the Backburner components.
- 3 Ensure the host name of the Backburner Manager appears in the following file:
/usr/discreet/backburner/cfg/manager.host
The **manager.host** file is used by the Backburner Server to locate the manager.
- 4 Restart the Backburner Server so that it picks up the new settings:
/etc/init.d/backburner restart

On each render node (Mac):

- 1 Install Maya and the Backburner components (Backburner Server is required).
See the *Maya Installation and Licensing Guide*.
- 2 Launch the Terminal application.
The Terminal application can be launched by selecting Go > Application > Utilities > Terminal.
- 3 Ensure the Backburner Server is installed and running by typing the following in the Terminal window:
ps -ef | grep backburner
Output similar to the following should appear:
0 35566 ... /usr/discreet/backburner/backburnerManager
0 37312 ... /bin/sh /usr/discreet/backburner/backburner_server run
0 37318 ... /usr/discreet/backburner/backburnerServer -m manager
Ellipses (...) in the above code output indicate material omitted from the sample output.

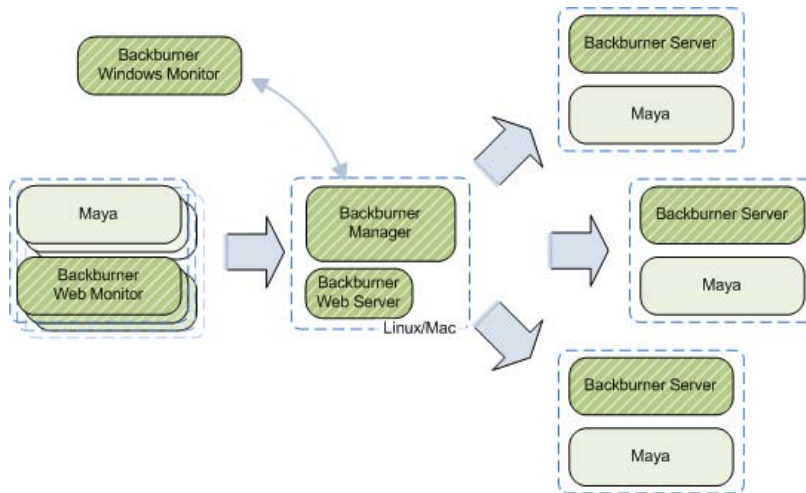
- 4 Ensure the host name of the Backburner Manager appears in the following file:
`/usr/discreet/backburner/cfg/manager.host`
The **manager.host** file is used by the Backburner Server to locate the manager.
- 5 Restart the Backburner Server so that it picks up the new settings:
`/usr/discreet/backburner backburner restart`

Maya Shared Render Farm

When the render farm is designed to service more than one creative seat of Maya, the Backburner Manager should reside on a dedicated workstation, designated as the master workstation. Other choices depend on whether the master workstation is running Linux, Mac OS, or the Windows operating system.

NOTE Idle Maya workstations can easily be enlisted as render nodes, and scheduled for off-peak or weekend use, if necessary.

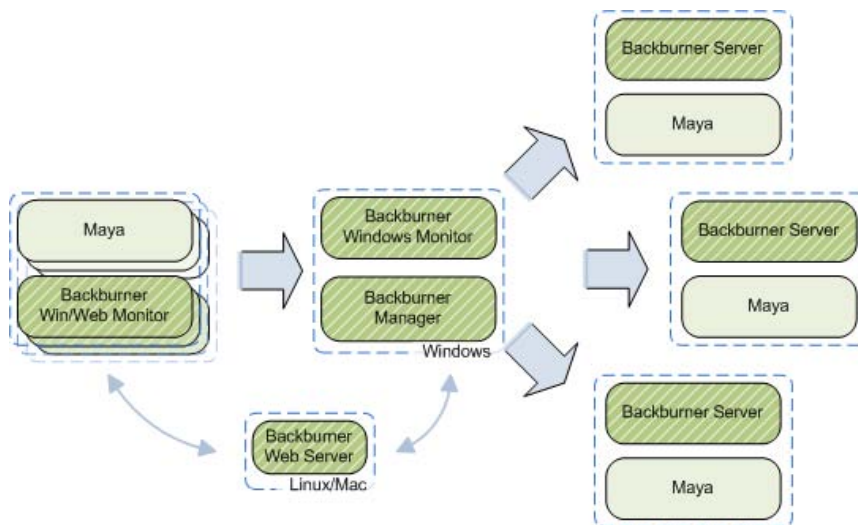
If the master workstation is a Linux or Mac OS machine, your Backburner network can have one instance of the resource-intensive Backburner Windows Monitor installed on a Windows machine, if available. Each of the creative seats can instead use the Backburner *Web* Monitor, reducing the load on the Backburner Manager and avoiding network congestion. This scenario is illustrated in the following diagram.



If the master workstation is a Windows machine, the Backburner Windows Monitor can be installed on the same machine, or on a separate Windows workstation. If you decide to install more than one or two instances of the Backburner Windows Monitor, keep network traffic to a minimum by disconnecting them from the Backburner Manager when they are not in use.

Optionally, non-administrator creative workstations can use the Backburner *Web* Monitor. In this case, you must install the Backburner Web Server on a Linux or Mac OS machine. The Backburner Web Server is not available on Windows.

This scenario is illustrated in the following diagram.



NOTE The procedures in this section assume you have experience setting up a dedicated render farm for a single creative seat of Maya. This allows the procedures to focus on the major steps only.

On the creative workstation:

- Install Maya.
See the *Maya Installation and Licensing Guide*.

On the master workstation:

- 1 Install the Backburner Components. The Backburner Manager is required. Backburner Windows Monitor is optional (Windows only).
See the *Maya Installation and Licensing Guide*.
- 2 To enable use of the Backburner Web Monitor from the creative workstation(s), install the Apache HTTP server (Linux/Mac).

NOTE The Apache HTTP server is the default web server under Mac OS X. There should be no need to install it independently on that platform.

- 3 Install and configure the Backburner Web Server, and verify that the Backburner Web Monitor is accessible to the default administrator user.
- 4 Set up end-user access to the Backburner Web Monitor.
- 5 Optional: Assign administrator privileges to selected end-users of the Backburner Web Monitor.
For steps 2–5, see:
 - [Backburner Web Monitor – Linux](#) on page 71.
 - [Installing the Backburner Components on Mac OS X](#) on page 76.
- 6 Optional: Connect the Backburner Windows Monitor to the Backburner Manager (Windows only).
See [Starting and Configuring the Backburner Windows Monitor](#) on page 60.

On a Windows workstation used for monitoring (optional):

- 1 Install the Backburner components. The Backburner Windows Monitor is required.

See the *Maya Installation and Licensing Guide*.

- 2 Connect the Backburner Windows Monitor to the Backburner Manager.
See [Starting and Configuring the Backburner Windows Monitor](#) on page 60.

On each render node (Windows):

- 1 Install Maya and the Backburner components (Backburner Server is required).
See the *Maya Installation and Licensing Guide*.
- 2 Connect the Backburner Server to the Backburner Manager running on the master workstation.
See [Starting and Configuring the Backburner Server](#) on page 60.
- 3 Optional: Configure the server to start as a Windows service.
See [Setting Up Backburner Server as a Windows Service](#) on page 63.

On each render node (Linux):

- 1 Install Maya and the Backburner components (Backburner Server is required).
See the *Maya Installation and Licensing Guide*.
- 2 Ensure the Backburner Server is installed and set to run as a daemon by typing the following:
chkconfig --list | grep -i backburner
Output similar to the following should appear:
backburner_server 0:off 1:off 2:on 3:on 4:on 5:on 6:off
Levels “3”, “4”, and “5” must be “on” for the daemons to start automatically. If this is not the case, reinstall the Backburner components.
- 3 Ensure the host name of the Backburner Manager appears in the following file:
/usr/discreet/backburner/cfg/manager.host
The **manager.host** file is used by the Backburner Server to locate the manager.
- 4 Restart the Backburner Server so that it picks up the new settings:
/etc/init.d/backburner restart

On each render node (Mac):

- 1 Install Maya and the Backburner components (Backburner Server is required).
See the *Maya Installation and Licensing Guide*.
- 2 Launch the Terminal application.
The Terminal application can be launched by selecting Go > Application > Utilities > Terminal.
- 3 Ensure the Backburner Server is installed and running by typing the following in the Terminal window:
ps -ef | grep backburner
Output similar to the following should appear:
0 35566 ... /usr/discreet/backburner/backburnerManager
0 37312 ... /bin/sh /usr/discreet/backburner/backburner_server run
0 37318 ... /usr/discreet/backburner/backburnerServer -m manager
Ellipses (...) in the above code output indicate material omitted from the sample output.
- 4 Ensure the host name of the Backburner Manager appears in the following file:


```
/usr/discreet/backburner/cfg/manager.host
```

The **manager.host** file is used by the Backburner Server to locate the manager.

- 5 Restart the Backburner Server so that it picks up the new settings:

```
/usr/discreet/backburner backburner restart
```

Maya Composite / 3ds Max Composite

To create a render farm for Maya Composite or 3ds Max Composite, you install Composite and the Backburner Server on each workstation to serve as a render node. There are two main options. In the first scenario, a render farm is created to service a single Composite creative seat. In the second, multiple creative seats are serviced by the render farm. In both scenarios, take care to install Backburner in the same manner on each render node, and to configure Composite so it makes use of Backburner. To render using Backburner, you publish the clip using the Backburner publish mode.

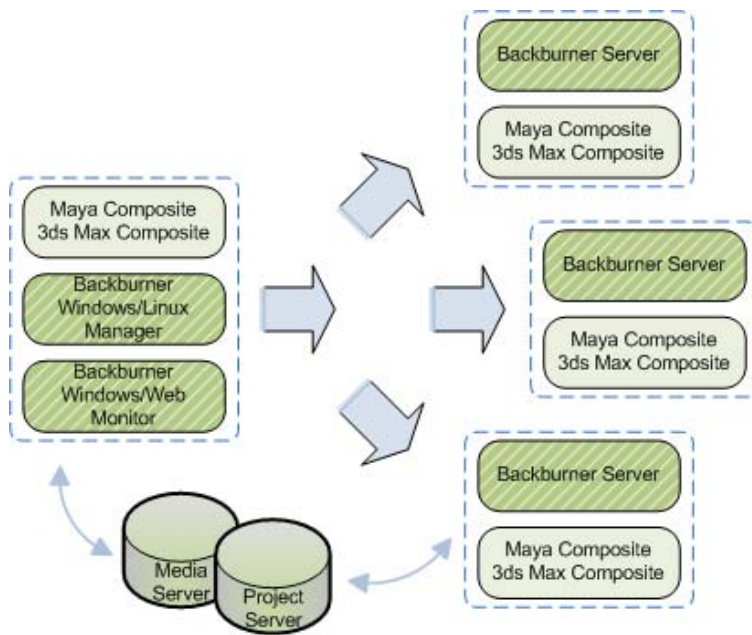
Composite render nodes must have the same directory structures as the workstation(s) originating the requests for network rendering. The Backburner executable must be on the same path on each render node. On Windows, for example, if the creative workstation has installed the Backburner executable on the C: drive, they cannot be installed on the D: drive on the render nodes. The path to the Backburner root directory is included as part of the job metadata that is sent to the render node. If the render node cannot find the directories, the job will fail. For similar reasons, the media server and project server (if used) should also be mounted in the same manner for each render node.

You must also ensure that on each creative workstation, Composite is set up to take advantage of network rendering using Backburner. Ensure the render executable command line is set to invoke Backburner, and that it indicates the machine name of the workstation where the Backburner Manager is installed. Also, ensure the *publish* directory is a shared folder accessible by all the render nodes.

Composite features a built-in progress bar indicating the progress of the rendering process and a means for observing jobs running in the background. However, for more detailed information, and full control over Backburner jobs, the Backburner Monitor can be used. Composite allows you to create a shortcut to the Backburner Monitor executable, so you can invoke it from the Composite UI. This will allow you to quickly bring up the Backburner Monitor when rendering using Backburner. To do so, set the monitor executable value in the Project Preferences window, Site Preferences tab.

Composite Dedicated Render Farm

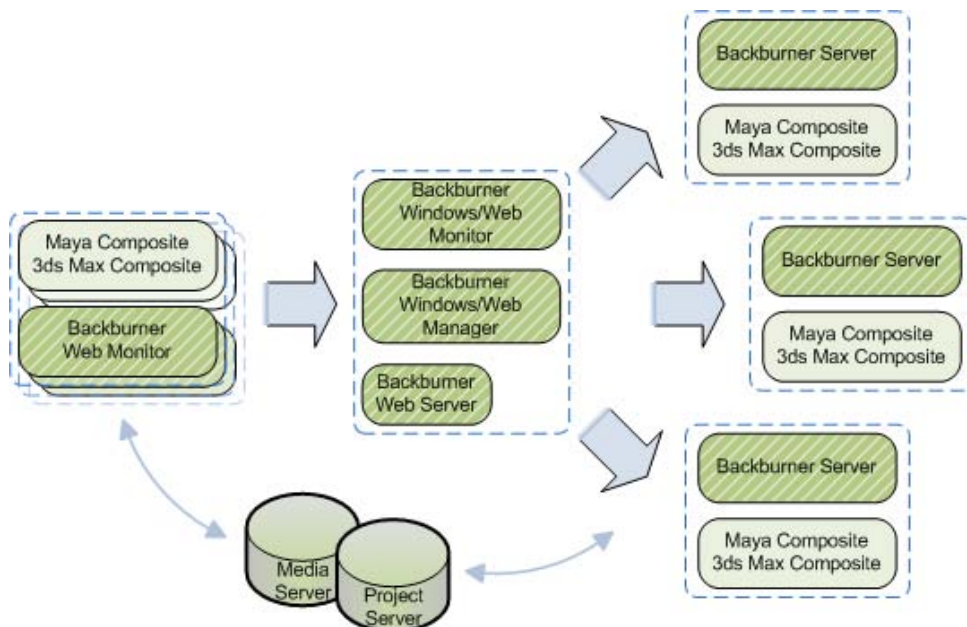
The following diagram illustrates a typical Composite render farm created for a single creative seat. In this scenario, the Backburner Manager and Backburner Monitor reside on the same workstation as the Composite creative application.



For instructions on setting up a dedicated render farm for Composite on Windows, refer to the procedures in [3ds Max Dedicated Render Farm](#) on page 37.

Composite Shared Render Farm

When the render farm is designed to service more than one creative seat of Composite, it is recommended that you install the Backburner Windows Monitor on another workstation, designated as the master workstation. Each of the creative seats can instead use the Backburner *Web* Monitor, reducing the load on the Backburner Manager. This scenario is illustrated in the following diagram. On a render farm serving multiple creative workstations, the Backburner Manager should be installed on a dedicated workstation.



For instructions on setting up a shared render farm for Composite on Windows, refer to the procedures in [3ds Max Shared Render Farm](#) on page 38.

Part 2: Installation Reference

Part 2 includes these chapters:

- [Backburner Components – Windows](#) on page 55
- [Backburner Manager – Linux](#) on page 67
- [Backburner Web Monitor – Linux](#) on page 71
- [Backburner Components – Mac](#) on page 75



Image courtesy of B Sky B Creative Services

Backburner Components – Windows

4

Topics in this chapter:

- [Overview](#) on page 55
- [Installing the Backburner Components on Windows](#) on page 56
- [Starting and Configuring the Backburner Manager](#) on page 57
- [Running the Backburner Manager as a Windows Service](#) on page 59
- [Starting and Configuring the Backburner Windows Monitor](#) on page 60
- [Starting and Configuring the Backburner Server](#) on page 60
- [Setting Up Backburner Server as a Windows Service](#) on page 63

Overview

The first step in setting up a render farm is installing the Backburner Manager, which can include relocating it from a workstation it shares with the creative application to a dedicated workstation. This must be the first step, since the Backburner Manager must be running before you can properly establish monitoring or configure render nodes. The second step is setting up monitoring of the Backburner network. The third step is setting up the render nodes.

Once the render farm is running smoothly, it can be convenient to set the Backburner Manager and Backburner Server to run as Windows *services*. This causes the applications to start with the workstation's operating system, and to run unobtrusively in the background. No GUI is presented on the workstation when an application runs as a service.

This chapter's primary aim is to assist in the above steps for a Windows-based Backburner render farm. It begins with instructions on installing the Backburner components. Next, it covers setting up the Backburner Manager, Backburner Monitor, and Backburner Server. It also contains instructions on setting up the Backburner Manager and Backburner Server as Windows services.

Backburner Manager

The following procedure shows the general workflow for setting up Backburner Manager in a render farm.

NOTE There should be just one instance of the Backburner Manager running on the network at a time.

To set up Backburner Manager:

- 1 Install Backburner Manager on the Windows workstation.
See [Installing the Backburner Components on Windows](#) on page 56.
- 2 Start and configure Backburner Manager.
See [Starting and Configuring the Backburner Manager](#) on page 57.
- 3 Optional: Set Backburner Manager to run as a Windows service.
See [Running the Backburner Manager as a Windows Service](#) on page 59.

Backburner Server

The following procedure shows the general workflow for setting up Backburner Server.

To set up Backburner Server:

- 1 Install Backburner Server on the Windows workstation.
See [Installing the Backburner Components on Windows](#) on page 56.
- 2 Start and configure Backburner Server.
See [Starting and Configuring the Backburner Server](#) on page 60.
- 3 Optional: Set Backburner Server to run as a Windows service.
See [Setting Up Backburner Server as a Windows Service](#) on page 63.

Installing the Backburner Components on Windows

In most cases, Backburner is installed by default with the creative application, or is an installation option. When this is not the case, you can use the following procedure to install the Backburner components manually.

NOTE This procedure presents a generalized method for installing the Backburner components on a Windows workstation from the Autodesk creative application CD/DVD. In most cases, an installation wizard appears automatically upon insertion of the CD/DVD, which can be used instead. In other cases, some of the steps may need to be adapted to your particular circumstances.

To install Backburner on a Windows workstation:

- 1 Log in to the workstation with administrator privileges.
- 2 In Windows Explorer, locate the *.zip* file for the Backburner installation package on the creative application's installation CD.

NOTE Depending on the creative application, the installation package may be provided as a Backburner.exe file instead. In this case, no extraction of contents is necessary — simply copy the executable to the workstation.

- 3 Extract the contents of the .zip file to a temporary folder on the workstation, making sure to preserve the path names stored in the archive.
- 4 Open the folder and double-click the Backburner installation file, *backburner.exe*.
The Backburner installation wizard appears.

NOTE If there is an earlier version of Backburner on the system, you are prompted to uninstall it.

- 5 Follow the prompts to install Backburner on the workstation.

NOTE It is recommended that you install Backburner components in the same destination folders on all machines.

- 6 Upon completion, the following shortcuts appear in the Windows Start menu.

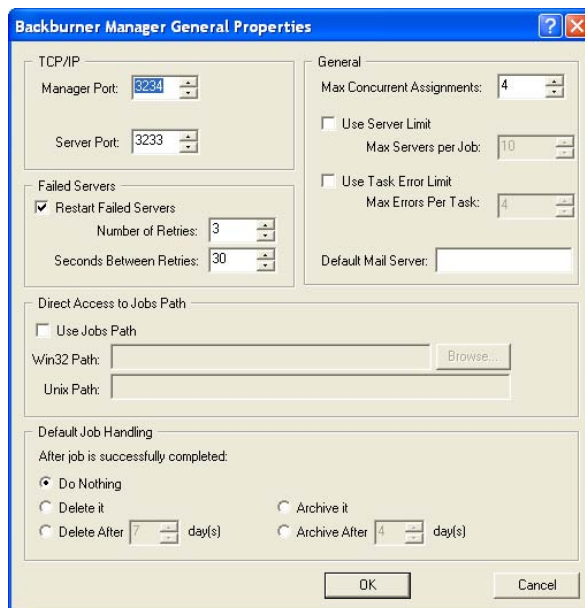
Menu Item	Application
Manager	Backburner Manager
Monitor	Backburner Monitor
Server	Backburner Server

Starting and Configuring the Backburner Manager

Once the Backburner components have been installed, you are ready to install and configure the Backburner Manager.

To start and configure the Backburner Manager:

- 1 From the Start menu, choose Programs, Autodesk, Backburner, and then Manager.
The first time you start the application, the General Properties dialog appears.



The dialog can be viewed at any time. From the Edit menu, choose General Settings, in the Backburner Manager UI.

2 Configure the manager as desired.

For most purposes the default settings should suffice. If not, see [Backburner Manager Configuration Settings](#) on page 81.

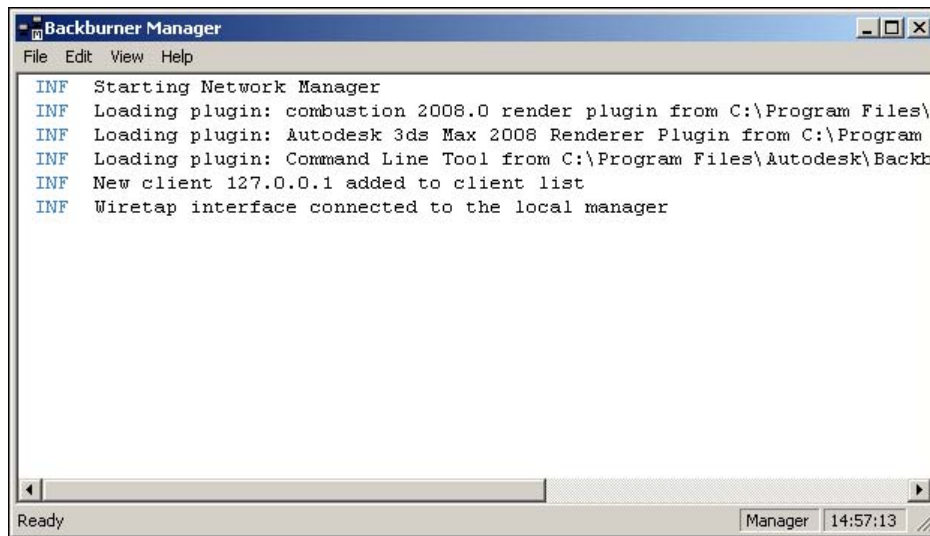
3 Click OK to start the application.

The configuration settings are written to the Backburner configuration file, *backburner.xml*, in a */backburner* subdirectory of the per-user application data directory. For example, in Windows XP:

C:\Documents and Settings\<user>\Local Settings\Application Data\backburner\backburner.xml

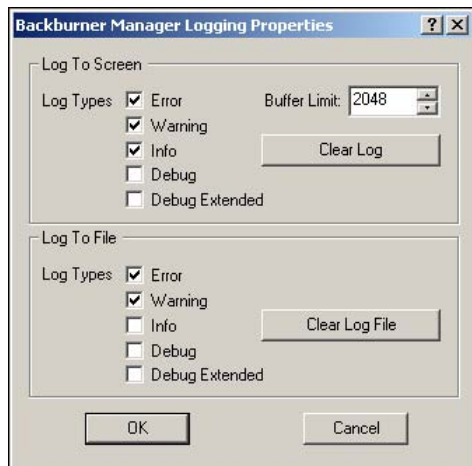
TIP To restore the default settings, simply delete the *backburner.xml* file. Note that this file contains the configuration settings for all Backburner components installed on the same machine.

The Backburner Manager GUI appears, displaying startup messages.



4 From the Edit menu, choose Log Settings to change the information that gets displayed in the GUI and/or written to the log file.

The Logging Properties dialog appears.



The settings in this dialog affect logging of the following events:

- Connection and registration between the Manager and Server(s)
- New job assignments
- The workstation that is the queue controller
- Frames assigned/rendered
- Acknowledge packets sent between the Manager and Server(s)
- Manager/Server(s) shutting down
- Rendering errors encountered

5 Click OK to save your changes, or Cancel to close the dialog without committing the changes.

The configuration settings are written to the Backburner configuration file, **backburner.xml**.

6 You must restart the Backburner Manager for the changes to take effect.

Running the Backburner Manager as a Windows Service

You can set up the Backburner Manager to run as a Windows service so that it starts with the workstation's operating system and runs in the background. When running as a service, no GUI is presented—events are logged to the log file only. A utility, **managersvc.exe**, has been provided to facilitate the setup process.

NOTE Depending on how Windows is configured, it may be necessary to log on to a Windows account for the service to load.

NOTE Running the Backburner Manager as a service changes the location of the *backburner.xml*, *wiretap.cfg*, and log files to */backburner* subdirectories of the per-user application data directory subdirectories belonging to the *Local Service* user. These directories are hidden to non-administrative level users. For example, on Windows XP: *C:\Documents and Settings\Local Service\Local Settings\Application Data\backburner\backburner.xml*

To set up Manager as a Windows service:

1 Open a Command Prompt (DOS) shell and navigate to the Backburner root directory:

C:\Program Files\Autodesk\Backburner

2 Register the Backburner Manager executable as a service by typing the following at the command prompt:

managersvc -i

The following message appears:

Backburner Manager

Service Installed

3 To remove the Backburner Manager, type the following:

managersvc -r

4 Start the service without rebooting. From the Start menu, choose Settings, Control Panel, Administrative Tools, and then Services.

The Services dialog appears.

5 Right-click the entry for the Backburner Manager, and choose Start.

Windows starts the manager, and updates its status in the dialog.

NOTE By right-clicking the entry for the Backburner Manager and choosing Properties, you can set a number of parameters, including the account under which the service logs on, and recovery options.

- 6 The next time you start the workstation and/or log on to Windows, the Backburner Manager is started automatically.

Starting and Configuring the Backburner Windows Monitor

The following procedure provides abbreviated instructions for launching the Windows version of the Backburner Monitor and connecting it to the Backburner Manager. For detailed instructions, see the *Autodesk Backburner User Guide*.

To launch the Backburner Windows Monitor:

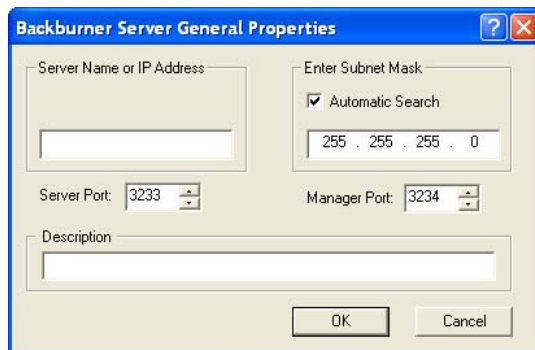
- 1 Start the Backburner Monitor. From the Start menu, choose Programs, Autodesk, Backburner, and then Monitor.
- 2 Connect the monitor to the manager. From the Manager menu, choose Connect. Then, deselect the Automatic Search checkbox and enter the manager name/IP address of the workstation running the Backburner Manager.
- 3 Verify the connection. From the Manager menu, choose Properties. Manager details appear in a dialog. If no connection was made, this option is greyed out.

Starting and Configuring the Backburner Server

Once the Backburner Manager is running and you have set up monitoring, you are ready to start and configure the Backburner Server. When this step is complete on each render node, the render farm is ready.

To start and configure the Backburner Server:

- 1 From the Start menu, choose Programs, Autodesk, Backburner, and then Server. The first time you start the application, the General Properties dialog appears.

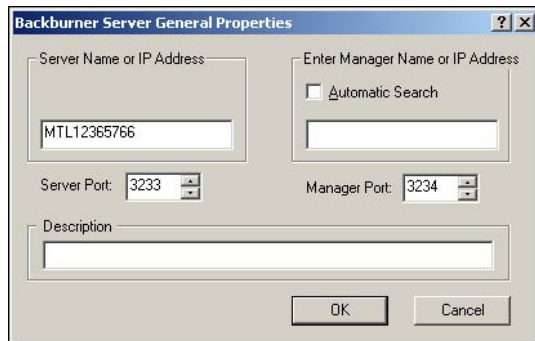


The dialog can be viewed at any time. From the Edit menu, choose General Settings, in the Backburner Server UI.

- 2 Enter the host name, server name, or IP address of the Backburner Server in the *Server Name or IP Address* field. This is the address that is used by the manager to communicate with the server.

- 3 Uncheck the *Automatic Search* checkbox, and enter the host name, server name, or IP address of the Backburner Manager in the *Enter Manager Name or IP Address* field.

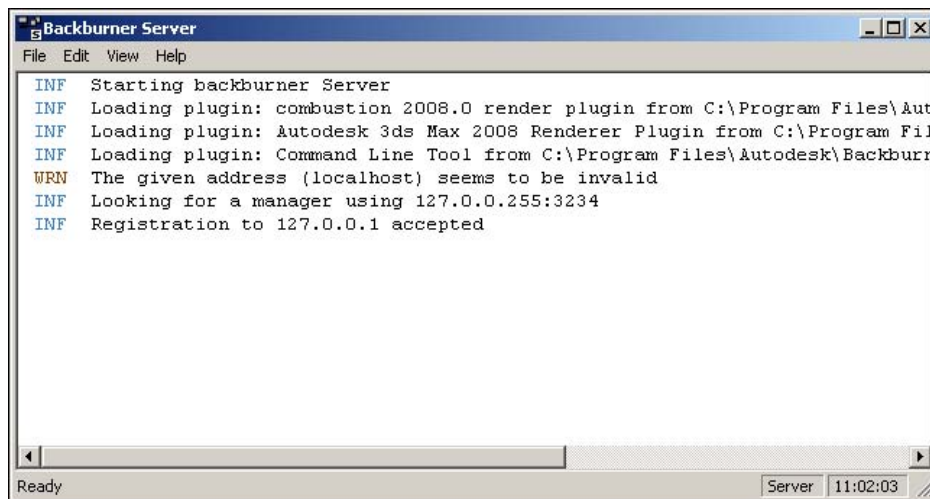
Employing automatic searching is not recommended, since it makes extensive use of network resources and can take some time.



- 4 Modify other configuration settings for the manager, as desired.
For settings details see [Configuring the Backburner Server](#) on page 62.

- 5 Click OK to start the application.

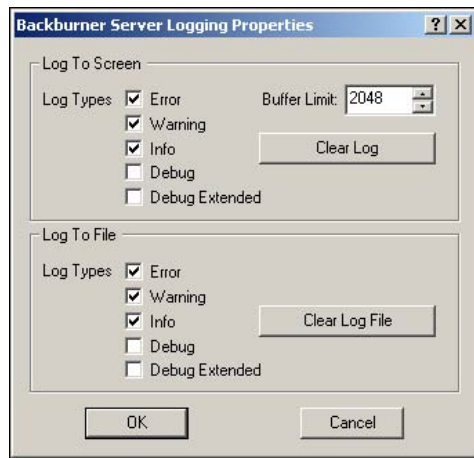
The Backburner Server GUI appears, displaying startup messages as it loads plug-ins and connects to the Backburner Manager.



Ensure a connection is established with the Backburner Manager before continuing.

- 6 To change the information that gets displayed in the GUI and/or written to the log file, from the Edit menu, choose Log Settings.

The Logging Properties dialog appears.



The settings in this dialog affect logging of the following events:

- Connection and registration between the Manager and Server(s)
- New job assignments
- Frames assigned/rendered
- Acknowledge packets sent between the Manager and Server(s)
- Manager/Server(s) shutting down
- Rendering errors encountered

- 7 Click OK to save your changes, or Cancel to close the dialog without committing the changes.

The configuration settings are written to the Backburner configuration file, **backburner.xml**, in a **/backburner** subdirectory of the per-user application data directory. For example, in Windows XP:
C:\Documents and Settings\<user>\Local Settings\Application Data\backburner\backburner.xml

TIP To restore the default settings, simply delete the *backburner.xml* file. Note that this file contains the configuration settings for all Backburner components installed on the same machine.

- 8 You must restart the Backburner Server for your changes to take effect.

Configuring the Backburner Server

The following describes each of the elements in the Backburner Server General Properties dialog.

Server Name or IP Address The host name or IP address the Backburner Server uses to identify itself to the Backburner Manager. It is helpful when a render node has more than one network interface card, hence more than one IP address.

Manager Port The port number used by the Backburner Manager. Changing this port number from its default value is not recommended. When using a different port, be sure to change the value on each Backburner component.

Server Port The port used by the render node. Changing this port number from its default value is not recommended. When using a different port, be sure to change the value on each Backburner component.

NOTE Make sure to configure any firewall to allow access to the manager and server ports. Otherwise, Backburner components will not be able to communicate.

Automatic Search Enable to allow the Backburner Server to search for a Backburner Manager using the TCP/IP subnet mask shown. It connects to the first manager found.

Disabling this option reduces Backburner network traffic, and is the recommended option. You must then specify the Backburner Manager explicitly in the *Enter Manager Name or IP Address* field.

Enter Manager Name or IP Address The name of the workstation on which the Backburner Manager is running. Alternately, its IP address or Domain Name System (DNS) name can be used.

This option can be used to address issues arising from running multiple Backburner Managers on the same TCP/IP subnet. For example, you can enter the Manager system's IP address to avoid any problems or conflicts caused by improper implementation of DNS. You can also use this option to segment your render farm by specifying the Backburner Manager to which each Render Node connects.

Setting Up Backburner Server as a Windows Service

Similarly to the Backburner Manager, it can be convenient to set up Backburner Server to run as a Windows service so that it starts with the workstation's operating system and runs in the background. When running as a service, no GUI is presented—events are logged to the log file only. A utility, **serversvc.exe**, has been provided to facilitate the setup process.

Recall that the server requires access to network mountpoints in order to obtain texture maps, media, etc., and to write results to output directories. By default, Windows services operate as local system accounts, which do not have the necessary remote access privileges. As a result, you must create a user account, assign it to the Backburner service, and give the account read/write access to the needed network mountpoints. These steps are outlined below.

To set up Backburner Server:

- 1 Create a 'privileged' user account to give the Backburner Server access to the network mountpoints containing the needed frames, textures, scenes, storage, etc.
See [Creating a Privileged User Account](#) on page 63.
- 2 Configure Backburner Server as a Windows service using the **serversvc.exe** utility.
See [Configuring Backburner Server as a Windows Service](#) on page 64.
- 3 Assign the privileged user account to the service so it starts as that user.
See [Assigning the Privileged User Account to the Service](#) on page 65.
- 4 Grant the user account read/write access to the needed mountpoints.
See the appropriate Windows documentation.

NOTE The procedures in this section are based on Windows XP Professional displaying a Classic Windows interface.

Creating a Privileged User Account

You create a user account for use by the Backburner Server service using the functionality of the Windows Control Panel. It is important to create the identical account on all workstations serving as render nodes. Then, when you give that account read/write access to network mountpoints, all render nodes will have the same access.

To create a special user for the Backburner Server service:

- 1 Log in to the workstation with administrator privileges.
- 2 From the Start menu, choose Settings, and then Control Panel.

- 3 In the Control Panel, navigate to Administrative Tools, Computer Management.
- 4 In the Computer Management dialog, navigate to System Tools, Local Users and Groups, and then Users.
- 5 From the Action menu, choose New User.
The New User dialog appears.
- 6 Create the new user as described in the following table.

Item	Requirement
User name	Use the same name for all render nodes.
Full name	not needed
Description	not needed
Password / Confirm password	Must be the same for all render nodes.
User must change password at next logon	disable
Password never expires	enable

- 7 Set up read/write access for the user on all needed network mountpoints, as described in Windows documentation.

Configuring Backburner Server as a Windows Service

The following procedure explains how to set up Backburner Server as a Windows service. A utility, **serversvc.exe**, has been provided to facilitate the setup process.

NOTE Depending on how Windows is configured, it may be necessary to log on to a Windows account for the service to load.

NOTE Running the Backburner Server as a service changes the location of the *backburner.xml* and log files to */backburner* subdirectories of the per-user application data directory subdirectories belonging to the *Local Service* user. These directories are hidden to non-administrative level users. For example, on Windows XP: *C:\Documents and Settings\Local Service\Local Settings\Application Data\backburner\backburner.xml*.

To install and register Backburner Server as a service:

- 1 Open a Command Prompt (DOS) shell and navigate to the Backburner root directory:
C:\Program Files\Autodesk\Backburner
- 2 Register the Backburner Server executable as a service by typing the following at the command prompt:
serversvc -i
The following message appears:
Backburner Server
Service Installed
- 3 To remove the Backburner Server, type the following:
serversvc -r
- 4 Start the service without rebooting. From the Start menu, choose Settings, Control Panel, Administrative Tools, and then Services.

The Services dialog appears.

- 5 Right-click the entry for the Backburner Server, and choose Start.
Windows starts the server, and updates its status in the dialog.

NOTE By right-clicking the entry for Backburner Server and choosing Properties, you can set a number of parameters, including the account under which the service logs on, and recovery options.

- 6 The next time you start the workstation, Backburner Server is started automatically.

Assigning the Privileged User Account to the Service

Once you have created the user account and configured the Backburner Server to run as a service, you are ready to assign the user account to the service. Assigning the account enables the service to assume the privileges associated with that account. Perform this procedure on each render node.

To assign a user to the Backburner Server service:

- 1 From the Start menu, choose Start, Settings, and then Control Panel.
- 2 In the Control Panel, double-click Administrative Tools, then Services.
- 3 In the Services dialog, right-click the Backburner Server item and choose Properties.
- 4 On the Log On tab, choose This Account and enter the name of the privileged user you created earlier.
If a user account was created on the domain, enter [domain name]\[user name] as This Account, or browse the domain for the user.
- 5 In the Password and Confirm Password fields, enter the password for the user account.
- 6 Click OK to exit the Properties dialog.
- 7 If the service is started, stop it by right-clicking the item and choosing Stop.
- 8 Right-click the item and choose Start to restart the service with the newly assigned user.
- 9 Close the Services dialog.
- 10 Give this user access privileges to the network mountpoints containing frames, textures, etc.

Backburner Manager – Linux

5

Topics in this chapter:

- [Overview](#) on page 67
- [Installing the Backburner Components on Linux](#) on page 68
- [Starting and Configuring Backburner Manager](#) on page 68

Overview

The Linux installation script can be used to install all three components: the Backburner Manager, Backburner Monitor, and Backburner Server. This chapter's primary aim is to assist in installing and setting up the Backburner Manager on a Linux workstation that is part of a render farm. It begins with instructions on installing the Backburner components. Next, it covers setting up the Backburner Manager, Backburner Monitor, and Backburner Server. It also contains instructions on setting up the Backburner Manager and Backburner Server as Windows services.

Set up the Backburner Manager on the designated Windows or Linux workstation. Because Backburner Manager is the hub of the network, it must be running before you set up Render Clients or Render Nodes. The following procedure shows the general workflow for setting up Backburner Manager.

To set up Backburner Manager:

- 1 Install the Backburner components on the Linux workstation.
- 2 Start and configure Backburner Manager.

Installing the Backburner Components on Linux

Use the following procedure to install Backburner on a Linux workstation from your Autodesk application DVD. It includes instructions on checking whether they are configured to start with the operating system, and configuring the Backburner components.

To install Backburner on a Linux workstation:

- 1 Log in to the workstation as *root* and open a terminal.
- 2 If installing from the application DVD, mount the DVD, navigate to the *dist* subdirectory of the application directory, and install the Backburner packages by typing:
rpm -Uvh backburner*386.rpm

NOTE If the application DVD includes an installation script for Backburner, use that instead.

- 3 If installing from the downloaded package, navigate to the sub-directory for the installed Linux distribution and run the installation script by typing:
./INSTALL
- 4 Follow the prompts to install the Backburner Manager and/or Backburner Server, per your configuration requirements.
Once completed, you are returned to the command prompt.
- 5 If this is an upgrade, installation is complete. Otherwise, you may need to run additional installation scripts. First, change directories:
cd /usr/discreet/backburner/
- 6 Run the Backburner configuration script to set up the Backburner Manager and Backburner Server:
./backburnerConfig
- 7 Use the Linux **chkconfig** command to determine if the manager and server are configured to start automatically:
chkconfig --list | grep -i backburner
Output similar to the following should appear:
backburner_manager 0:off 1:off 2:on 3:on 4:on 5:on 6:off
backburner_server 0:off 1:off 2:on 3:on 4:on 5:on 6:off
Levels 3, 4, and 5 must be set to “on” for the daemons to start automatically. If no output appears, or if the output is different from the above, please contact Customer Support.

Starting and Configuring Backburner Manager

Under normal circumstances, there should be no need to configure the Backburner Manager. However, should the need arise, this section provides instructions for doing so. The most common changes—such as specifying the default mail server through which Backburner sends job-related email specifications—can also be made via the Backburner Web Monitor.

To start and configure Backburner Manager:

- 1 Log in to the workstation as *root* and open a terminal.
- 2 Stop the Backburner Manager service. Type:
/etc/init.d/backburner stop

The Backburner Manager service on the workstation is stopped, if it was running previously. If it was not running previously, a “Failed” error message is displayed. If you are starting Backburner Manager for the first time, do steps 3 to 6 to configure the Manager before starting it. Otherwise, go to step 7 to start the Manager.

- 3 Navigate to the *Network* directory of the Backburner application and view its contents by typing:

```
cd /usr/discreet/backburner/Network; ls
```

The *backburner.xml* file in this directory contains the configuration settings for the Backburner Manager. You must edit this file to configure the Manager for your facility. Before editing it, make a backup copy of this file.

- 4 Make a backup copy of the *backburner.xml* file. Type:

```
cp backburner.xml backburner_backup_<date>.xml
```

where <date> is today's date.

The file is created in the *Network* directory. You can refer to and/or restore this file in case of problems.

- 5 If necessary, edit the *backburner.xml* file in a text editor such as *vi*, or an XML editor, to configure Backburner Manager for your facility. See [Backburner Manager Configuration Settings](#) on page 81.
- 6 Save and close the edited *backburner.xml* file.

- 7 Start Backburner Manager by typing:

```
/etc/init.d/backburner start
```

The Backburner Manager service starts, using the configuration information from the *backburner.xml* file. Once Backburner Manager is running properly, you are ready to set up the Backburner Web Monitor and the Render Nodes.

TIP All Backburner Manager events are recorded in log files in the directory */usr/discreet/backburner/Network/backburner.log*. You can use the *tail* command to view the contents of this log for a Linux-based Backburner Manager.

Backburner Web Monitor

– Linux

6

Topics in this chapter:

- [Overview](#) on page 71
- [Installing Backburner Web Monitor](#) on page 72
- [Installing the Apache HTTP Server for Linux](#) on page 72
- [Installing the Apache HTTP Server Software](#) on page 72
- [Setting Up Access to the Backburner Web Monitor](#) on page 73
- [Assigning Backburner Web Monitor Administrator Privileges](#) on page 73

Overview

The Backburner Monitor is the user interface to the Backburner network. It is a Windows application that is installed automatically when you install Backburner Manager on a Windows system. If you install Backburner Manager on a Linux system, this application is not available.

The Backburner Web Monitor is the Web-based version of the Backburner Monitor. It can run on any Linux or Mac workstation. It runs in a Web browser and is accessible from any host on the network.

Although the interfaces are not identical, both the Backburner Monitor and the Backburner Web Monitor provide the same functionality. Both also assume the workstation on which they are running has access to the Backburner network.

Installing Backburner Web Monitor

Backburner Web Monitor relies on Web server software which you must install and configure. After you install and configure this software, users can open a browser and connect to Backburner Web Monitor. The software you must install and configure is:

- The following Web server application:
 - The Apache HTTP server, which is a popular open-source Web server application available for many platforms including Windows, Linux, and Mac OS.
- If necessary, consult your system administrator to determine the Web server application used by your facility and for assistance with installing and configuring this application. Installing the Web server application requires an understanding of HTML or XML, an understanding of Web server configuration, and a basic understanding of Backburner's architecture.
- Backburner Web Server (installed automatically with Backburner components)

Installing the Apache HTTP Server for Linux

If your facility is running Backburner for Autodesk Visual Effects and Finishing applications on Linux, you can also run the Backburner Web server from a Linux workstation or Render Node with the Apache HTTP server for Linux.

The following procedure shows the general workflow for installing and configuring the Apache HTTP server on a Linux workstation.

NOTE It is not necessary to install Backburner Web Server on Linux.

To install and configure the Apache HTTP servers:

- 1 Install the Apache HTTP server on the Linux workstation where Backburner Manager is installed. See [Installing the Apache HTTP Server for Linux](#) on page 72.
- 2 Set up access to the Backburner Web Monitor users so network users can manage jobs and Render Nodes through the Web Monitor. See [Setting Up Access to the Backburner Web Monitor](#) on page 73.
- 3 Specify which users have Backburner Web Monitor administrator privileges. See [Assigning Backburner Web Monitor Administrator Privileges](#) on page 73.

Installing the Apache HTTP Server Software

Install the Apache HTTP server for a Linux workstation using the following procedure. This procedure assumes you are installing the Apache HTTP server that is included with your Linux distribution.

NOTE You must have root access to perform the following procedure.

To install the Apache HTTP server:

- 1 Log in as root and open a terminal.
- 2 Check if the Apache HTTP server is installed. Type:
chkconfig --list | grep -i httpd

If the Apache HTTP server's `httpd` daemon does not appear in the output, continue to the next step. Otherwise, go to step 4.

- 3 Install the Apache HTTP server package (`httpd`) from your Linux distribution on the Linux workstation where Backburner Manager is installed. Refer to your Linux distribution's documentation or help system for installation instructions.

NOTE It is recommended that you install the Apache HTTP server on the same workstation as the Backburner Manager. Do not install these Web servers on multiple workstations on the Backburner network.

- 4 Make sure that the Apache HTTP server is set to start with the workstation. Type:
`chkconfig httpd on`
- 5 Start the Apache HTTP server by typing:
`/etc/init.d/httpd start`

Setting Up Access to the Backburner Web Monitor

Setting up access to the Backburner Web Monitor requires that you create Backburner Web Monitor user accounts. Backburner Web Server requires all users to provide a login name and password to access the Backburner Web Monitor.

NOTE The default user account *backburner* is created during the installation of Backburner Manager. The password associated with this account is *backburner*. For security, you may wish to change it.

To create a Backburner Web Monitor user account:

- 1 Log in as root and open a terminal.
- 2 Check if the *backburner.auth* file is in place. Type:
`ls /etc/httpd/auth`
If the following output appears, proceed to step 4:
`backburner.auth`
- 3 If the *backburner.auth* is not in place, copy it from the Backburner folder. Type:
`cd /usr/discreet/backburner/WebMonitor`
`cp backburner.auth /etc/httpd/auth`
- 4 Add the account. Type:
`htpasswd /etc/httpd/auth/backburner.auth <username>`
- 5 Enter a password when prompted.
The Backburner Web Monitor can now be accessed with the account information you have entered.

To delete a Backburner Web Monitor user account:

- 1 Log in as root and open a terminal.
- 2 Delete the account. Type:
`htpasswd -D /etc/httpd/auth/backburner.auth <username>`

Assigning Backburner Web Monitor Administrator Privileges

Users without administrator privileges can modify and control their own jobs, and monitor the status of all Backburner jobs in the Backburner Web Monitor. Users with administrator privileges can actively manage

all jobs and render nodes. For a complete list of tasks that require administrator privileges, refer to the *Autodesk Backburner User Guide*.

NOTE The default user account *backburner* created during the installation of Backburner Manager has administrator privileges by default. If you are creating new user accounts, you may wish to remove administrator privileges from the default account, for security. Alternately, change the password.

To assign administrator privileges to a Backburner Web Monitor user account:

- 1 Log in as root and open a terminal.
- 2 Navigate to the folder containing the file that defines administrator privileges:
cd /usr/discreet/backburner/Network
- 3 Open the file *wiretap.cfg* in a text editor and scroll down to the [SECURITY] section. This section contains the Backburner Administrators keyword, which specifies the user accounts with administrator privileges. For example, the following assigns administrator privileges to the user account *backburner*:
BackburnerAdministrators=backburner
- 4 Edit the Backburner Administrators keyword, separating account names with a comma. For example, the following assigns administrator privileges to the user accounts *backburner*, *loic*, and *solemn*:
BackburnerAdministrators=backburner,loic,solemn
- 5 Save and exit the file.

When users *backburner*, *loic*, and *solemn* log in to the Backburner Web Monitor, the system management tools appear in the user interface, indicating the user account has administrator privileges. For the location of the system management tools in the user interface, refer to the section “Accessing System Management Tools from the Backburner Web Monitor” in the *Autodesk Backburner User Guide*.

Backburner Components – Mac

7

Topics in this chapter:

- [Overview](#) on page 75
- [Backburner Manager Installation Overview](#) on page 76
- [Installing the Backburner Components on Mac OS X](#) on page 76
- [Configuring the Backburner Manager](#) on page 77
- [Setting Up Access to the Backburner Web Monitor](#) on page 78
- [Assigning Backburner Web Monitor Administrator Privileges](#) on page 78
- [Configuring the Backburner Server](#) on page 79

Overview

The first step in setting up a render farm is installing the Backburner Manager, which can include relocating it from a workstation it shares with the creative application to a dedicated workstation. This must be the first step, since the Backburner Manager must be running before you can properly establish monitoring or configure render nodes. The second step is setting up monitoring of the Backburner network. The third step is setting up the render nodes.

This chapter presents generalized methods for installing Backburner components needed for setting up a Mac-based render farm. Generally speaking, you should use the application CD/DVD that came with your creative application to install all Backburner components. In most cases, an installation wizard appears automatically upon insertion of the CD/DVD, in which case that can be used instead of the instructions provided here. In addition, the installation wizard may provide the ability to install Backburner components independently of the creative application itself.

For example, the Smoke for Mac OS X installation DVD (or downloaded image file) can be used to install all three Backburner components, independently of Smoke:

- Backburner Manager
- Backburner Monitor
- Backburner Server

This chapter's primary aim is to assist in installing and setting up the Backburner Manager on a Mac workstation that is part of a render farm. It begins with instructions on installing the Backburner components. It also covers setting up the Backburner Web Monitor and Backburner Server.

NOTE For Smoke for Mac OS X, the Backburner Server is installed with Burn. See the *Burn Installation and User Guide*.

Backburner Manager Installation Overview

Because the Backburner Manager is the hub of the network, it must be running before you set up the creative application to take advantage of network rendering, or the render nodes themselves. The following procedure shows the general workflow for setting up the Backburner Manager in a render farm.

NOTE There should be just one instance of the Backburner Manager running on the network at a time.

To set up Backburner Manager:

- 1 Install the Backburner Manager on the Mac workstation.
See [Installing the Backburner Components on Mac OS X](#) on page 76.
- 2 Start and configure the Backburner Manager.
See [Configuring the Backburner Manager](#) on page 77.
- 3 Ensure end-users have access to the manager via the Backburner Web Monitor.
See [Setting Up Access to the Backburner Web Monitor](#) on page 78.
- 4 Assign some Backburner Web Monitor users administrator privileges.
See [Assigning Backburner Web Monitor Administrator Privileges](#) on page 78.

Installing the Backburner Components on Mac OS X

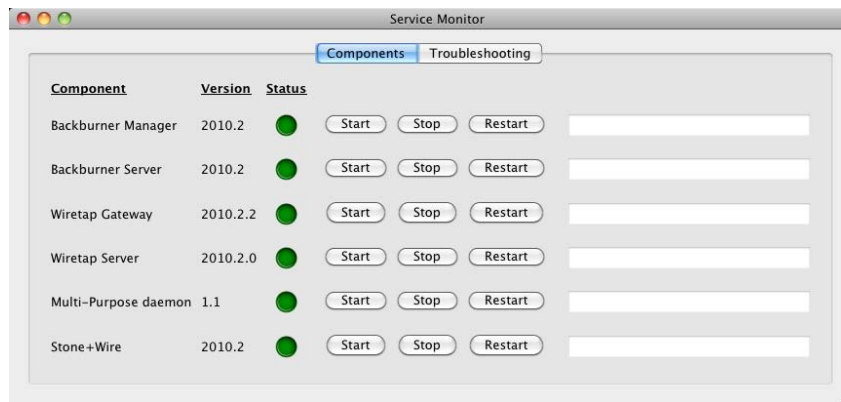
Use the following procedure to install Backburner on a Mac workstation from your Autodesk application CD/DVD or downloaded disk image (.dmg) file. It includes instructions on checking whether they are configured to start with the operating system, and configuring the Backburner components.

NOTE The following procedure applies to Smoke for Mac OS X only. For Maya, use the application DVD to install all needed Backburner components.

To install Backburner on a Mac workstation:

- 1 Insert the Smoke for Mac OS X application DVD and double-click the DVD icon that appears on the desktop.
If installing from a downloaded disk image (.dmg) file, double-click that file first.
- 2 Make use of the installation wizard options to install the Backburner Manager and/or Backburner Server, per your configuration requirements.

- 3 Once completed, verify that the needed components have been properly installed and started. Go to *Applications / Autodesk* in Finder, and open the *Autodesk Systems Monitor* application.



Configuring the Backburner Manager

Under normal circumstances, there should be no need to configure the Backburner Manager. However, should the need arise, this section provides instructions for doing so. The most common changes—such as specifying the default mail server through which Backburner sends job-related email specifications—can also be made via the Backburner Web Monitor.

To configure Backburner Manager:

- 1 Launch the Terminal application.
The Terminal application can be launched by selecting *Go > Application > Utilities > Terminal*.
- 2 Stop the Backburner Manager daemon. Type:
`/usr/discreet/backburner/backburner stop`
The Backburner Manager daemon on the workstation is stopped, if it was running previously.
- 3 Navigate to the *Network* directory of the Backburner application and view its contents by typing:
`cd /usr/discreet/backburner/Network; ls`
The *backburner.xml* file in this directory contains the configuration settings for the Backburner Manager. It is this file that must be edited to configure the Manager for your facility.
- 4 Make a backup copy of the *backburner.xml* file. Type:
`cp backburner.xml backburner_backup_<date>.xml`
where *<date>* is today's date.
The file is created in the *Network* directory. You can refer to and/or restore this file in case you have problems.
- 5 Open the *backburner.xml* file in a text editor such as *vi*, or an XML editor.
- 6 Configure the Backburner Manager settings as desired. See [Backburner Manager Configuration Settings](#) on page 81.
- 7 Save and close the edited *backburner.xml* file.
- 8 Start the Backburner Manager by typing:
`/usr/discreet/backburner/backburner start`

The Backburner Manager service starts, using the configuration information from the *backburner.xml* file.

Once the Backburner Manager is running properly, you are ready to set up the Backburner Web Monitor and the Render Nodes.

TIP All Backburner Manager events are recorded in log files in the directory */usr/discreet/backburner/Log*. You can use the *tail* command to view the contents of this log for a Mac-based Backburner Manager.

Setting Up Access to the Backburner Web Monitor

Setting up access to the Backburner Web Monitor requires that you create Backburner Web Monitor user accounts. The Backburner web server requires all users to provide a login name and password to access the Backburner Web Monitor.

NOTE The default user account *backburner* is created during the installation of the Backburner Manager. The password associated with this account is *backburner*. For security reasons, you may wish to change it.

To create a Backburner Web Monitor user account:

- 1 Launch the Terminal application.
 - 2 Check if the *backburner.auth* file is in place. Type:

```
ls /private/etc/apache2/auth
```

If the following output appears, proceed to step 4:

```
backburner.auth
```
 - 3 If the *backburner.auth* is not in place, copy it from the Backburner folder. Type:

```
cd /usr/discreet/backburner/WebMonitor
```

```
cp backburner.auth /private/etc/apache2/auth
```
 - 4 Add the account. Type:

```
htpasswd /private/etc/apache2/auth/backburner.auth <username>
```
 - 5 Enter a password when prompted.
- The Backburner Web Monitor can now be accessed with the account information you have entered.

To delete a Backburner Web Monitor user account:

- 1 Log in as root and open a terminal.
- 2 Delete the account. Type:

```
htpasswd -D /private/etc/apache2/auth/backburner.auth <username>
```

Assigning Backburner Web Monitor Administrator Privileges

Users without administrator privileges can only monitor the status of Backburner jobs in the Backburner Web Monitor. Users with administrator privileges can actively manage all jobs and render nodes. For a complete list of tasks that require administrator privileges, refer to the *Autodesk Backburner User Guide*.

NOTE The default user account *backburner* created during the installation of the Backburner Manager has administrator privileges by default. If you are creating new user accounts, you may wish to remove administrator privileges from the default account, for security. Alternately, change the password.

To assign administrator privileges to a Backburner Web Monitor user account:

- 1 Launch the Terminal application.
- 2 Navigate to the folder containing the file that defines administrator privileges:
cd /usr/discreet/backburner/Network
- 3 Open the file *wiretap.cfg* in a text editor and scroll down to the [SECURITY] section. This section contains the Backburner Administrators keyword, which specifies the user accounts with administrator privileges. For example, the following assigns administrator privileges to the user account *backburner*:
BackburnerAdministrators=backburner
- 4 Edit the Backburner Administrators keyword, separating account names with a comma. For example, the following assigns administrator privileges to the user accounts *backburner*, *loic*, and *solenn*:
BackburnerAdministrators=backburner,loic,solenn
- 5 Save and exit the file.
When users *backburner*, *loic*, and *solenn* log in to the Backburner Web Monitor, the system management tools appear in the user interface, indicating the user account has administrator privileges. For the location of the system management tools in the user interface, refer to the section “Accessing System Management Tools from the Backburner Web Monitor” in the *Autodesk Backburner User Guide*.

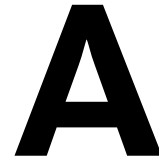
Configuring the Backburner Server

The following procedure illustrates how to configure the Backburner Server so it can locate a Backburner Manager.

To configure the Backburner Server:

- 1 Launch the Terminal application.
The Terminal application can be launched by selecting Go > Application > Utilities > Terminal.
- 2 Ensure the Backburner Server is installed and running by typing the following in the Terminal window.
ps -ef | grep backburner
Output similar to the following should appear:
0 37312 ... /bin/sh /usr/discreet/backburner/backburner_server run
Ellipses (...) in the above code output indicate material omitted from the sample output.
- 3 Ensure the host name of the Backburner Manager appears in the following file:
/usr/discreet/backburner/cfg/manager.host
The **manager.host** file is used by the Backburner Server to locate the manager.
- 4 Restart the Backburner Server so that it picks up the new settings:
/usr/discreet/backburner backburner restart

Backburner Manager Configuration Settings



Topics in this chapter:

- [Overview](#) on page 81
- [TCP/IP Settings](#) on page 82
- [General Settings](#) on page 82
- [Failed Node Behavior](#) on page 83
- [Job Path Settings](#) on page 83
- [Default Job Handling Behavior](#) on page 84

Overview

The Backburner Manager can be configured in its General Properties dialog (Windows) or by editing the *backburner.xml* file (Windows/Linux).

NOTE The Backburner Manager should be stopped before configuring.

While the default settings are appropriate for most cases, you may have to adjust these settings to enable communication between the Backburner Manager and other components.

NOTE The *backburner.xml* file provides many elements that can be used to customize Backburner Manager. Take care when modifying this file directly unless you are familiar with XML and are comfortable with editing *.xml* files. Do not modify the *backburner.xml* file beyond the scope of the elements covered in the following sections, unless directed to do so by Customer Support. Otherwise, Backburner stability may be affected.

TCP/IP Settings

Configure general TCP/IP settings for the Backburner Manager by modifying either fields in the TCP/IP group of the Backburner Manager General Properties dialog, or elements in the *backburner.xml* file. The fields/elements specify the TCP/IP port numbers and subnet mask to be used by Backburner.

Port numbers are like extensions for different users of the same telephone number. They represent two channels of communication between the Render Node and the Manager. Only a trained network administrator should change these settings.

Changing these port numbers from their default values is not recommended. If you decide to use other port numbers, these values must be changed in the TCP/IP settings of each Backburner component.

Field	XML Element	Description
Manager Port	<ManagerPort>	Specifies the port number used by the Backburner Manager. The default value is 3234.
Server Port	<ServerPort>	Specifies the port number used by the Render Node(s). The default value is 3233.
N/A	<NetworkMask>	Sets the default subnet mask used for the network. The default value is 255.255.255.0. This value can be overridden using the Windows-based Backburner Monitor; when you launch the Monitor, from the Manager menu, choose Connect. Enter the subnet mask in the dialog that appears.

NOTE If you are using a firewall in your facility, make sure you configure it to allow access to the Manager and Server ports. Otherwise, Backburner components will not be able to communicate, receive footage, or return rendered material.

General Settings

Configure general Backburner Manager behavior by modifying either fields in the General group of the Backburner Manager General Properties dialog, or elements of the *ManagerSettings* sub-element of the *AppDetails* element of the *backburner.xml* file.

Field	XML Element(s)	Description
Max Concurrent Assignments	<MaxConcurrentAssignments>	<p>Specifies the number of jobs the Manager sends out simultaneously. This number depends on the processor speed of the Manager workstation, general job size, and overall network speed.</p> <p>Generally, the default value of 4 is adequate. You can change this value in the following situations:</p> <ul style="list-style-type: none">■ Decrease this value when rendering jobs are large and/or Backburner is struggling to keep up with jobs.■ Increase this value when rendering jobs are small and/or Backburner has no trouble keeping up with jobs. <p>Avoid increasing this value substantially; too high a value may cause an increased number of node timeouts because the jobs are sent faster than the nodes can handle them. In such cases, decrease the value.</p>

Field	XML Element(s)	Description
Use Server Limit	<UseServerLimit> and <ServerLimit>	Sets the maximum number of Render Nodes that will be allocated for a specific job. This feature can override the server limit settings in some applications. For information, see the application's Advanced Settings Dialog.
Use Task Error Limit	<UseTaskErrorLimit> and <TaskErrorLimit>	Defines the number of times a Render Node retries a task before suspending it.
Default Mail Server	<DefaultMailServer>	Defines the default mail server used for email notification. This mail server is used only for jobs that do not already have a mail server associated with them.

Failed Node Behavior

Configure the Backburner Manager to automatically restart Render Nodes with failed jobs by modifying either fields in the Failed Server group of the Backburner Manager General Properties dialog, or elements in the `ManagerSettings` sub-element of the `AppDetails` element of the *backburner.xml* file.

Field	XML Element(s)	Description
Restart Failed Servers	<RetryFailedServers>	Activate to enable automatic Render Node restarting. If this option is disabled, the Render Node does not attempt to render the job again after the first failure. This option is enabled by default.
Number of Retries	<RetryCount>	Specifies the number of times the Manager attempts to restart a failed Render Node. The number of retries is set to 3 by default. The range is between 1 and 1024.
Seconds Between Retries	<TimeBetweenRetries>	The time between each retry in milliseconds. This value is set to 30000 milliseconds by default. In Windows, this value is adjusted to be displayed as 30 seconds.

The state of a Render Node is recorded on a per-job basis. The Backburner Manager regularly goes through the list of Render Nodes for a job, checking for failures. If Restarts Failed Servers is enabled (set as 1 in the *backburner.xml* file), the Manager keeps track of the following:

- The time a Render Node fails on a particular job.
- The time elapsed since the node failed on a particular job.

If the time elapsed is greater than the specified Seconds Between Retries, the Backburner Manager decreases the Number of Retries by one and resets the Failed flag for the Render Node. Once the failure count reaches the specified Number of Retries, the Manager stops trying to restart the Render Node for that particular job.

Once a restarted Render Node completes a frame, it is flagged as active and resumes processing until the job is complete.

Job Path Settings

Configure the Backburner Manager to access jobs on specified system or network drives by modifying either fields in the Direct Access to Job Path group of the Backburner Manager General Properties dialog, or elements of the `ManagerSettings` sub-element of the `AppDetails` element of the *backburner.xml* file.

You use these fields/elements when you do not want to have jobs kept on the drive or workstation where the Backburner Manager is installed. Examples of such situations are:

- You have very little drive space on the system drive where the Backburner Manager is installed. Because another system drive has plenty of space, you share a folder on this drive called *MyJobs*, where jobs are placed when submitted. In this case, you would direct jobs to `\\computername\MyJobs`.
- You are using your workstation simultaneously as a Backburner Manager and creative workstation. To reduce the overhead on the workstation from the Backburner Manager, you set up a shared job folder on a network file server, called *backburnerJobs*. The Win32 job path would be set to `\\filesrvr\backburnerJobs` and jobs you submit placed on the file server.

Field	XML Element(s)	Description
Use Jobs Path	<EnableJobPath>	When enabled, defines job location using the Win32 or UNIX paths. This tells the Render Nodes to get the job files from this location, minimizing the file I/O traffic on the Manager workstation.
Win32 Path	<Win32Path>	The Windows file path where jobs are located. You can click the Browse button in the General Properties dialog to search your system for the job location.
Unix Path	<UnixPath>	The Unix file path where jobs are located. You can click the Browse button in the General Properties dialog to search your system for the job location.

Default Job Handling Behavior

Configure how the Backburner Manager handles completed jobs by modifying either fields in the Default Job Handling group of the Backburner Manager General Properties dialog, or elements of the *ManagerSettings* sub-element of the *AppDetails* element of the *backburner.xml* file.

You use these settings to better manage the job queue in the Manager. For example, use these settings to delete a completed job after a certain number of days or leave the job indefinitely in the queue to be archived. Archiving is useful when you submit a final version of the project and you know that no additional changes are required.

NOTE These settings can be overridden from the Advanced Settings dialog accessed from the Network Job Assignment dialog.

<DefJobHandling> Setting	State	XML Element(s)	Description
0	Do Nothing		When enabled in the General Properties dialog, a completed job is left in the queue.
1	Delete it		When enabled in the General Properties dialog, the job is deleted from the queue upon completion.
2	Delete After	<DeleteDays>	When enabled in the General Properties dialog, the job is kept in the queue for the specified number of days upon completion. Once the number of days has been exceeded, the job is deleted from the queue.

<DefJobHandling> Setting	State	XML Element(s)	Description
3	Archive it		When enabled in the General Properties dialog, the job is archived upon completion. Archive it is enabled by default for Windows Managers.
4	Archive After	<ArchiveDays>	When enabled in the General Properties dialog, the job is kept in the queue for the specified number of days upon completion. Once the number of days has been exceeded, the job is deleted from the queue.

Index

3ds Max
 data scalability 35
 render farm 37
 standalone model 11
3ds Max Composite
 standalone model 13

A

Apache HTTP server
 Linux installation 72

B

Backburner
 architecture 5
 defined 3
 deployment models 7
 installing on Linux 68
 installing on Mac 76
 installing on Windows 56
 recommendations 8
 supported applications 3
 system requirements 8
 terminology 4
 third-party development 8
Backburner Manager
 configuring 81
 defined 6
 isolating 29
 third-party development 8
Backburner Manager Linux
 configuring 81
 starting 68
Backburner Manager Windows
 configuring 57
 configuring as a service 59
 starting 57
Backburner Monitor
 defined 6
 Windows vs web 31
Backburner Server Windows
 configuring 62
 configuring as a service 63
 starting 60
Backburner Web Monitor
 defined 27
 user management 33
Backburner Web Server, defined 4

Backburner Windows Monitor
 configuring 60
bucket rendering 5
Burn
 Burn for Lustre 44
 Visual Effects and Finishing 18, 24

C

Cleaner XL
 Smoke 2010 for Mac OS X 43
 Visual Effects and Finishing 17, 23, 41
conventions, notation 10
customer support, contacting 10

D

data scalability 35
documentation for this release 10

L

Lustre
 Burn for Lustre 44
 network transcoding with Backburner 45
 render farm 44
 standalone model 25

M

Maya
 render farm 46
 standalone model 26
Maya Composite
 standalone model 26

N

notation conventions 10

P

packets
 defined 5

R

- render farm
 - 3ds Max 37
 - dedicated render farm 28
 - deployment options 28
 - Lustre 44
 - shared render farm 29
 - Smoke 2010 for Mac OS X 42
 - steps to set up 27
 - Visual Effects and Finishing 39
 - WiretapCentral and Smoke 2010 for Mac OS X 43
 - WiretapCentral and Visual Effect and Finishing 41
- render nodes
 - defined 6
 - setting up 34

S

- Smoke 2010 for Mac OS X
 - Cleaner XL 43
 - render farm 42
 - WiretapCentral 43
- support, contacting 10
- System architecture 5

T

- Toxik
 - render farm 51

V

- Visual Effects and Finishing
 - background I/O 13
 - Burn 18, 24
 - Cleaner XL 17, 23, 41
 - data scalability 36
 - render farm 39
 - standalone model 13, 19
 - wire transfers 13
 - WiretapCentral 18, 25, 41

W

- Windows service
 - setting up Backburner Manager 59
 - setting up Backburner Server 63
- Wiretap 8
- Wiretap Gateway 25
- WiretapCentral
 - render farm 39
 - Smoke 2010 for Mac OS X 43
 - Visual Effects and Finishing 18, 25, 41