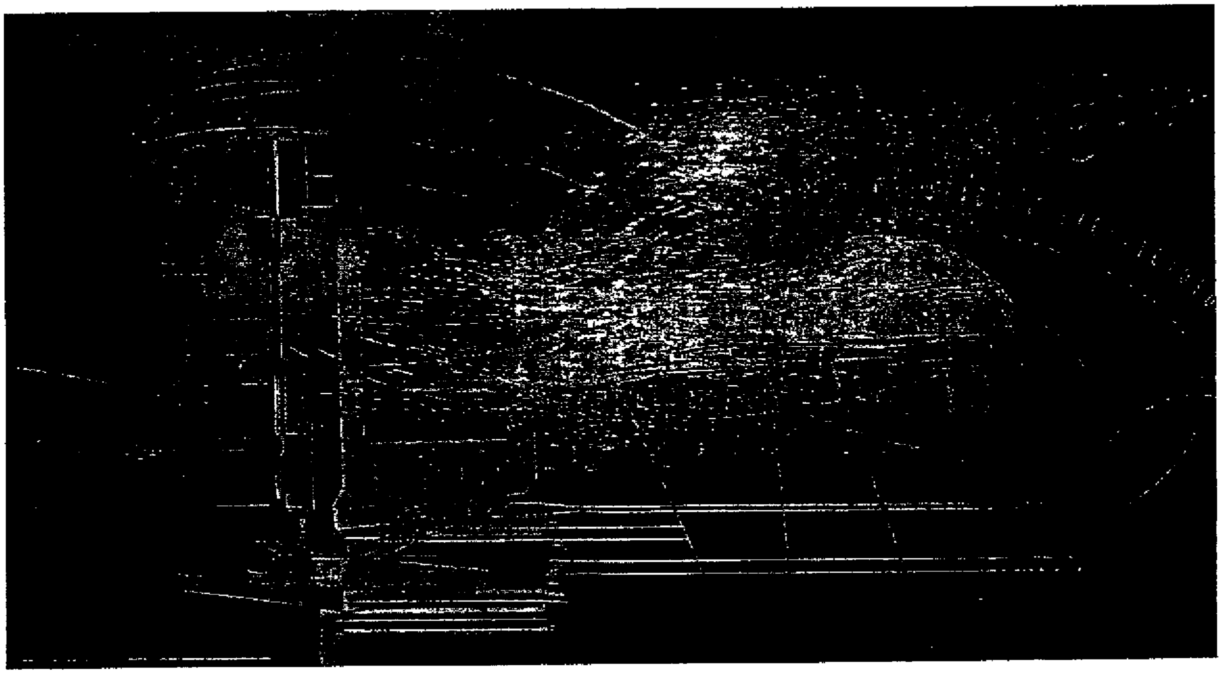
Burstware®

[User Guide]



burstware®

Burst.Com, Inc.
500 Sansome Street
San Francisco, CA 94111

Warranties

Great care has been taken to ensure that the information presented here is as accurate, comprehensive and timely as possible. Burst.Com, Inc. is not responsible for errors or omissions and in no event shall Burst.Com, Inc. be liable for any consequential or incidental damages, lost profits, or any indirect damages.

Contents

4 Deploying Burstware on Your Network

5 Installing and Starting Conductors and Servers

8 Optional Configuration of Burst-Enabled Players

9 The Burstware Bridge Monitor

10 Conductor and Server Runtime Management

11 Burstware Security

Index

CHAPTER 1

Introduction

Burstware User Guide is written for system administrators, network administrators, and web developers who plan, implement, and manage Burstware environments.

Guide to this Document

Chapter 2, Burstware Overview, highlights the business and technical problems that Burstware addresses, and provides a high level description of Burstware functionality and components.

Chapter 3, Planning a Burstware Deployment, discusses deployment considerations and alternative configurations for Burstware. System and network administrators responsible for planning, implementing, and supporting Burstware configurations should read this chapter before installing Burstware.

Chapter 4, Deploying Burstware on Your Network, presents and discusses several sample Burstware deployments.

Chapter 5, Installing and Starting Conductors and Servers, gives detailed instructions on installing and starting the components of your Burstware deployment.

Chapter 6, Configuring Conductors and Servers, discusses customizing conductor and server behavior by editing initialization files and describes the configuration parameters for Burstware Conductor and Burstware Server.

Chapter 7, Burst-Enabling Players, explains how to install the Burstware Bridge for Windows Media Player and the Burstware Bridge for Apple QuickTime Player, and how to play files. It also discusses error handling, supported file formats and CODECs, and the Slow Connection Warning dialog.

Chapter 8, Optional Configuration of Burst-Enabled Players explains how to customize the behavior of conductors and servers by setting configuration parameters.

Chapter 9, The Burstware Bridge Monitor, describes how to install the Burstware Bridge Monitor, and how to use it to obtain network and media file play information.

Chapter 10, Conductor and Server Runtime Management, explains how to monitor and interpret runtime information and how to manage log files.

Chapter 11, Burstware Security, describes the security features available with Burstware.

Typographical Conventions

Burstware User Guide uses the following typographical conventions:

Style	Meaning	Example
italics	Variables or items for which you must supply a value	Each entry has the form *hostname:port_number* or *IP_address:port_ number*.
[ ]	Variables or items for which supplying a value is optional	`\\burst://conductor_list/file_name[?params]` `metafile=.burst`
bold	Configuration parameters	The **BackBufferSize** parameter specifies the size, in kilobytes (KB), of the back buffer portion of the Burstware buffer.
	File names	
	Directory names	

Style	Meaning	Example
code	Commands you must type at an operating system prompt HTML code	Change the current directory to the mount directory: `% cd /cdrom/burst_11`

Contacting Burst.Com, Inc.

We value your comments on this guide. Please use the information below to contact us with your comments:

Burst.Com, Inc.
500 Sansome Street
San Francisco, CA 94111

Phone: (415) 391-4455

Fax: (415) 391-3392

email: doc@burst.com

CHAPTER 2

Burstware Overview

As companies across the business spectrum flock to implement Internet-based applications and services, the network itself is increasingly the victim of its popularity. Both public and private TCP/IP networks are now expected to shoulder a burden for which they were not designed.

The broad benefits provided by Internet-based applications have sparked a ubiquitous demand for rich content, delivered in high volume with fidelity and reliability. Multimedia is effective, engaging, and everywhere. Internet-based applications that embed multimedia content are changing the way that companies:

- Interact with customers
- Build brand identity
- Sell and deliver entertainment
- Provide news and sports coverage

Problematically, applications that include multimedia are the ones that demand the most from a network, and degrade the most visibly when network performance is poor. Adding insult to injury, the more successful a company is in generating traffic and transactions on its Web site, the more likely it is to experience an interruption of satisfactory Internet service.

TCIP/IP networks deliver many small packets of information. Each of the many packets that comprise a single transfer can take a unique route to its final destination. Each route is subject to different latencies and bandwidth constraints. Packet-based protocols were developed when content meant text and timing issues were a low priority. For static media, like text, the synchronization and timing realities of TCP/IP are acceptable. However, today's multimedia applications deliver audio and video content, for which delivery time is critical.

Burstware is designed to help companies create and maintain efficient and reliable media-rich applications.

The Burstware Environment

Burstware is a software solution that can be deployed in any network that uses the TCP/IP protocol. Burstware allows you to manage and deploy network bandwidth efficiently, thereby ensuring the highest quality viewing experience for your end users.

In a "burst-enabled" environment, the Burstware components manage the delivery of multimedia content. Burstware accepts requests from clients, monitors the status of the components that send and receive multimedia files, and "bursts" requested content into client-side buffers.

With Burstware, you can allocate bandwidth to your multimedia applications and control their impact on your network. You can minimize the effects of unpredictable network performance on the end user's viewing experience.

Burstware is compatible with the other tools, applications, and solutions that constitute your content management and deployment environment.

How Burstware Works

Figure 1 is a simple conceptual view of the Burstware components and how they cooperate and communicate with each other.

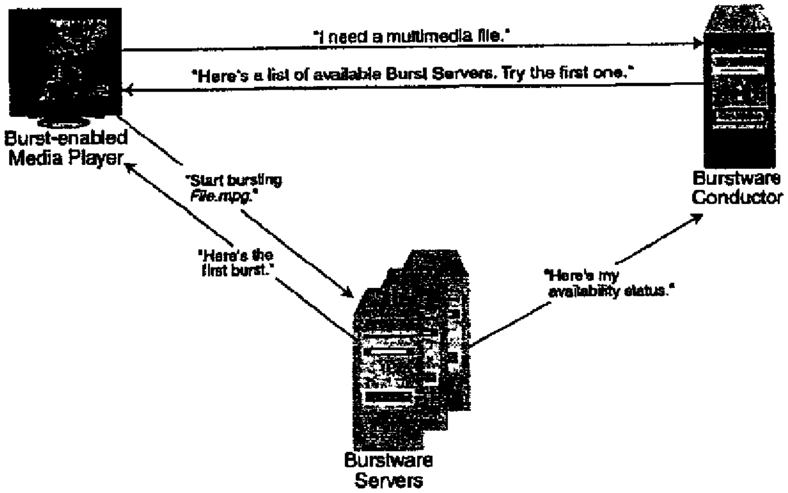

"I need a multimedia file."
"Here's a list of available Burst Servers. Try the first one."
Burst-enabled Media Player
"Start bursting *File.mpg*."
"Here's the first burst."
Burstware Conductor
"Here's my availability status."
Burstware Servers

Figure 1: Burstware conceptual overview

In a Burstware deployment, the client is a media player, running standalone, or embedded in an application. A Burstware Conductor monitors the availability of bandwidth for multimedia file transmission, and the load on Burstware Servers—the components that actually serve content to clients.

A Burstware Conductor intelligently allocates requests for content to Burstware Servers, which, in turn, intelligently serve a set of client machines. A Burstware Server keeps each client-side buffer provisioned with sufficient "inventory" of the requested content to provide a high-quality viewing experience for the person watching the video.

The end result is maximized throughput and bandwidth utilization, and a satisfying viewing experience for the end user.

The following table summarizes the key interactions among the components in a Burstware environment:

Burstware Conductor	• Receives requests for service from clients • Monitors the availability and load of Burstware Servers • Provides a list of available Burstware Servers to a requesting client
Burstware Servers	• Sends requested media to clients • Monitors their clients' local Burstware caches to determine when, and how much data to deliver to each client
Burstware Clients	• Notifies Burstware Conductor when a user requests a multimedia file • Connects to the Burstware Server(s) recommended by the conductor to get the file

The key Burstware components are described in the paragraphs that follow.

Burstware Conductor

Burstware Conductor manages the distribution of player requests over multiple Burstware Servers. When a player requests content, a conductor sends back a list of servers. The conductor orders the list based on which servers have the most bandwidth available. The list is composed of available and standby servers. See the section, Load Balancing and Failover with Available and Standby Servers, on page 38 for more information on the different states a server can be in.

Burstware Conductor manages the following functions:

- Load balancing—balances the multimedia transmission load among multiple servers (for example, network connections).

- Server management—monitors server activity and availability; allows users to add and remove servers dynamically.
- Centralized network control—monitoring and management of the number of clients and total managed bandwidth.
- Point of contact for burst-enabled media players—the conductor manages all communications with players, and insulates players from other deployment details.
- Failover—provides automatic failover to another Burstware Conductor and/or Burstware Server, when necessary. Failover is transparent to clients.

Burstware Server

A Burstware Server manages video and audio content and delivers it across the network to computers with burst-enabled players.

The server performs the following functions:

- Player monitoring—tracks client-side buffer levels and consumption rates, network latency, and other statistics in order to optimize the use of network bandwidth.
- Intelligent bandwidth allocation—bursts data to each of its clients' buffers, based on needs and resources.

  The server works to maximize utilization of bandwidth, while stocking the client side buffers with a backlog of data to ensure continued play, even in the event of network delays. The server determines the size and timing of "bursts" of data to client caches based on client buffer levels and network bandwidth conditions.
- Throttling—controls bandwidth consumption to a pre-defined level, specified by the **managedBandwidth** parameter, allowing network administrators to regulate the impact of audio and video on their network. Bandwidth consumption for individual requests for media files can also be controlled.

A typical Burstware implementation includes multiple Burstware Servers, which helps balance load among multiple connections, and enables failover in the event that a given server becomes unavailable.

Burst-enabled Media Players

A media player is "burst-enabled" with the addition of a Burstware software bridge. Burstware Bridges are available for the Microsoft Windows Media Player (WMP) and the Apple QuickTime Player for Windows. A burst-enabled player cooperates with a Burstware Conductor and a Burstware Server to:

- Request content—request multimedia content via the conductor
- Share client status—report the state of client buffers
- Coordinate delivery of multimedia content

The Burstware Bridges are non-invasive; they do not limit or affect a player's user interface or functionality. A burst-enabled player can also play content from non-Burstware environments. Applications that embed supported players can be easily burst-enabled.

A burst-enabled Windows Media Player plays a variety of industry-standard media formats, including ASF, AVI, MPEG1, MPEG2, MP3, WMA, and QuickTime. A burst-enabled Apple QuickTime Player plays the QuickTime media format and all media formats and CODECs that the QuickTime player supports.

One or more burst-enabled players are configured with the names of the active and standby conductors. In the event that the active conductor is unavailable, the players contact the standby conductor.

Burstware Buffer

The Burstware buffer is an intelligently managed memory or disk buffer on the player machine, in which data delivered by a Burstware Server is cached. Caching the data locally provides for uninterrupted, jitter-free viewing and frees up network resources, resulting in their more efficient use.

Burstware ensures that the player buffer always contains data and the player has immediate access to the data it needs. Burstware determines the size and type of the buffer, based on the player machine's resources. In addition, the type and size of buffers can be explicitly controlled by setting the values of appropriate Burstware configuration parameters.

The Advantage of Large Buffers

In general, the larger the Burstware buffer, the better the end user's viewing experience. A larger buffer improves the viewing experience because the more media data the Burstware system is able to cache on the player machine before the end user needs it, the less likely it is that network problems will disrupt the delivery of the media data.

Large buffers also enable the Burstware client to accept large bursts of data when network traffic is low, making the most efficient use of available bandwidth. The resulting bandwidth savings mean that the Burstware system can service more users with the same network and server hardware.

Multimedia Content

Multimedia content can be stored anywhere on your network. The directories that contain content are identified to the Burstware environment. For a "no single point of failure" environment, replication of content is required.

Functional Flow in a Sample Configuration

Figure 2 illustrates a simple configuration, in which one Burstware Conductor manages three Burstware Servers. (No standby conductor is shown, to simplify the diagram.)

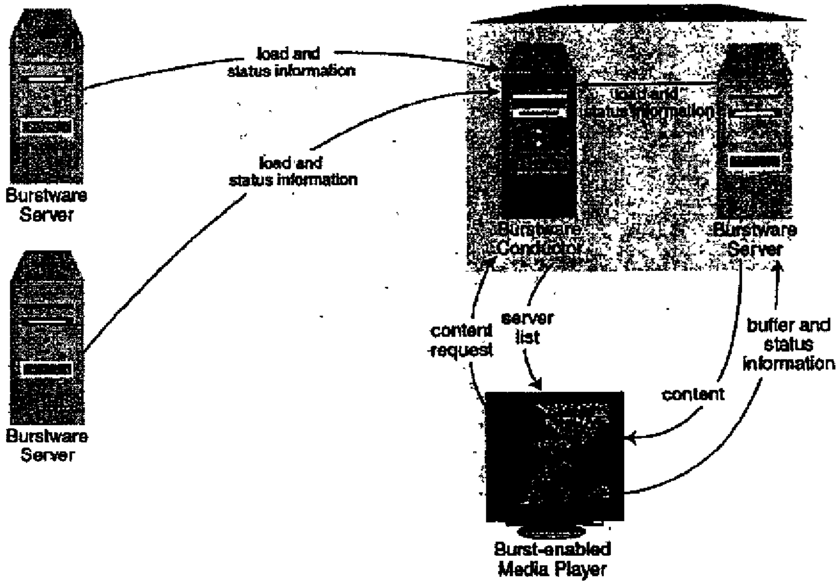

load and status information
load and status information
load and status information
Burstware Server
Burstware Server
Burstware Conductor
Burstware Server
content request
server list
buffer and status information
content
Burst-enabled Media Player

Figure 2: Burstware functional flow

The conductor periodically polls the three servers, to determine status and availability. Each server monitors the clients it is currently serving.

When a player requests a media file from a conductor, the following occurs:

1. The conductor sends the player a list of servers, ordered by availability based on recent polling results.
2. The player attempts to connect with each server on the list until successful.

3. The server starts bursting data into the Burstware buffer on the player machine.
4. The burst-enabled player starts playing data for the end user.
5. Bursting continues until the file has been completely transferred to the client.
6. Play continues until the file has been completely played or the user stops play.

Burstware Failover Strategy

Burstware provides failover for both conductor and servers. Failover means that if a conductor or server becomes unavailable due to machine, network, or other problems, another assumes its function. For example, if the machine a conductor is running on fails, a standby conductor will assume management of the first conductor's servers.

Burstware's failover architecture has a number of benefits:

- It enables deployment with no single point of failure
- It ensures uninterrupted viewing in the event of software or hardware failure
- It provides a cost-effective alternative to hardware-based redundancy solutions.

The following sections provide background information about conductor and server failover.

Conductor Failover

Burstware environments typically include two Burstware conductors: a primary conductor and a standby conductor. This strategy enables failover in the event the primary conductor becomes unavailable for any reason.

When an end user requests media, directly or via an application, the application accesses an ordered list of conductors. (A system or network administrator creates the list during the deployment planning phase.)

To request content, the player attempts to connect with its primary conductor. If the primary conductor does not respond, the player attempts to connect to the standby conductor. If the player is unable to connect to any of the conductors on the ordered list, Burstware reports an error.

Server Failover

Burstware Server's failover capabilities help to ensure an uninterrupted viewing experience for the end user. If the failure of a component interrupts delivery of content to the client, play from the local Burstware buffer continues while a connection between the client and another Burstware Server is established. The new server starts content delivery at the point at which service by the previous server was interrupted. As a result, play continues without interruption, and the end-user is unaware that either a failure or the resulting failover occurred.

If the connection between the player and the server is interrupted at any time, the player requests an updated list of available servers from the conductor. The player attempts to establish a connection to the first server on the list. If that fails, the player attempts to establish a connection to the second server, then with the third server, and so on, through the ordered list.

If the player cannot establish a connection to any of the servers on the list, the Burstware system sleeps for a set period of time (called the reconnect interval), and then repeats the process of getting a list of servers and attempting connection. If the player is unable to connect to a server after a set period of time (called the reconnect timeout), the Burstware system reports an error.

CHAPTER 3

Planning a Burstware Deployment

This chapter describes topics related to planning and deploying your Burstware environment. Information is presented in these sections:

- The Deployment Planning Process
- Burstware and the Rest of Your Environment
- Deployment Considerations
- Configuration Decisions

The Deployment Planning Process

Burstware is typically deployed as a part of an overall content publishing environment. Companies choose to integrate multimedia in web-based applications for a variety of reasons. Often, the corporate initiative related to such applications is strategic in nature.

In such cases, implementing Burstware is one aspect of a larger implementation process that might include some or all of the activities described below:

- Design and implementation of the overall application

  If you are planning to deploy multimedia using media players built into an application, your Burstware deployment is part of an overall application development process. This process may include, for example, the design and implementation of an application that either contains or creates the web pages presented to end users.

  In terms of the application architecture, Burstware is mostly transparent. For example, in a web application, developers can leverage Burstware's capabilities by including appropriate markup in the HTML to point to conductors and set media player parameters.

- Design and implementation of the Burstware configuration

The technical topics related to deploying Burstware are covered in this manual. From a project management point of view, it may be wise to take a phased approach to implementing the desired configuration. Tasks in the process could include:

- Planning the production configuration (a configuration that will support all of the content and users planned for your application)
- Implementing a simple test configuration—perhaps a subset of the configuration planned for production use—to verify correct communications among components and validate functionality

  You can use this pilot environment as a testbed to evaluate how Burstware and other elements of the overall environment work together.
- Based on the test configuration, refining plans for the production configuration, if needed
- Establishing the production configuration
- Introducing the production configuration to the end user community, in phased fashion if appropriate
- Refining the production configuration, if needed

- Monitor and maintain environment

  After launching the production application environment, ongoing support tasks may include user and security administration, system management, problem resolution, and user support.
- Expand the environment

  Over time, as additional content and bandwidth are introduced and your user community grows, you may need to revisit configuration planning and testing.

Burstware and the Rest of Your Environment

This section describes how Burstware fits with other applications and tools that comprise your multimedia management and delivery environment. It also helps prepare you for making Burstware configuration decisions.

Web Servers

Access to rich media is often provided to the end user by embedding a media player in web pages containing HTML (or DHTML, ASP, or JavaScript), which are served by a web server.

When users start to play multimedia files using a burst-enabled media player, the URL in the selected web pages is a request that specifies the media file name, the IP address or hostname of the Burstware Conductor, and other information that Burstware uses in the bursting process.

In essence, as far as your web application is concerned, Burstware is just like any other video server. The methods Burstware uses to provide a high quality viewing experience are transparent to your application and to end users.

Key reminders related to Burstware and your web server:

- Don't put Burstware on your web server machine unless you must

  TCP/IP port conflicts and performance degradation could result if Burstware and your web server are on the same machine.

- The web pages containing media players must reference Burstware

  The pages serving multimedia content must reference that content via Burstware. The HTML invokes a burst-enabled player, passing the Burstware URL as a parameter. See Chapter 7, Burst-Enabling Players, for information on how to embed a burst-enabled player in a web page.

Firewalls

Firewalls are security systems that enforce an access control policy between networks. Typically, firewalls sit between private networks and the Internet, regulating the types of traffic leaving and entering the private network.

Firewalls, TCP/IP Ports, and Burstware

Generally speaking, different types of data travel via different TCP/IP ports on a computer. For example, HTTP traffic—traffic flowing to and from web servers— usually flows through port 80. Firewalls allow network administrators to control access to the network by leaving certain ports open and others closed. Most firewalls are configured to leave port 80 open.

Burstware allows you to specify the exact port numbers that Burstware components use to transmit control information and to send and receive data, using configuration parameters stored in Burstware initialization files. These parameters are:

- **basePort**—the port used by Burstware Conductor and Burstware Server to listen for requests and send control information
- **dataPort**—the port used by Burstware Server for multimedia data transfers. The value of **dataPort**, if not specified, defaults to **basePort** plus 1.

NOTE: **There is no requirement that dataPort be set to the address one higher than basePort. This is the default number assigned to dataPort when the address is not specified.**

Chapter 6, Configuring Conductors and Servers, provides detailed information about Burstware port assignments.

Content Managers

Sites that deliver multimedia content sometimes use content management systems to store content files and to maintain related metadata, such as revision status, keywords, and attributes. Content management systems may store the multimedia data in a file system, and maintain the metadata in a relational database.

Burstware is fully compatible with such environments, if the multimedia content is stored in the file system. The **mediaPath** configuration parameter need only point to the relevant directory locations.

Content Distribution Networks

High volume content publishers who serve a broad geographical area may employ content distribution networks. Content distribution networks manage and replicate content in multiple locations, such as different Internet points-of-presence (POPs). Burstware integrates easily into such environments. Each Burstware Server in such an environment must point to the appropriate directory locations via the **mediaPath** server configuration parameter. See the section, Burstware Server Configuration Parameters, on page 107 for more information about the **mediaPath** parameter.

Web Site Generation Tools

Some web sites employ tools that generate the web pages on demand, often from database-resident information. This capability and the content management capabilities described in the section, Content Managers, on page 30, sometimes exist together in e-commerce application environments.

HTML-generating tools are valuable in sites:

- That present personalized pages based on end user or visitor preferences or roles
- Whose underlying content is subject to frequent change
- Where content generation tasks are distributed among many contributors

You can use a site-generation tool, or any other type of product that automatically constructs web pages, to create pages that invoke a burst-enabled media player.

Personalized pages can be constructed on demand, based on predefined rules and media attributes stored in the relevant database. However, just as in less dynamic web applications, to leverage Burstware, the pages need only set the player configuration parameters, including specifying the target content in the play request URL. While the site generation tool may be dynamically selecting appropriate content when building the page, this process is transparent to Burstware.

For more information on the mediaPath parameter, see the section, Burstware Server Configuration Parameters, on page 107.

Local Load Balancers

Companies with high-volume web sites often employ local load balancers to balance incoming traffic among multiple web servers. In planning a Burstware configuration, we recommend that you carefully distinguish between the load balancing capabilities of generic load balancing tools and Burstware's built-in multimedia load balancing.

Currently available load balancers and Burstware provide complementary capabilities and advantages. Generally speaking, local load balancers are designed to deal with flat, or static content, such as text, while Burstware provides load balancing designed to maximize sustained multimedia transfers.

Load balancers often use a round robin routing approach. This strategy works best for connectionless protocols and in situations where packets of data are generally of similar size.

In contrast, Burstware provides load balancing designed specifically for connection-based protocols and for traffic that involves media of varying sizes and encoding rates. Burstware's load balancing approach takes into account the connection speeds, the encoded bit rate of media files, and length of the media file.

Some of the more sophisticated local load balancers do take available bandwidth into consideration when allocating requests. However, Burstware is almost invariably better suited for load balancing multimedia traffic, because it has access to and uses information about client resources and status to enable optimal load balancing and bursting.

For the reasons stated above, we recommend that in environments that include local load balancers you do the following:

- Load balance your non-multimedia content with traditional load balancers
- Load balance multimedia with Burstware Conductor

Figure 3 illustrates a Burstware configuration that implements this recommendation. In this configuration, the load balancer directs clients to the most available web server. The web server serves web pages. Pages can contain embedded media players, which point to the conductor. In this configuration, multimedia load balancing is accomplished by Burstware.

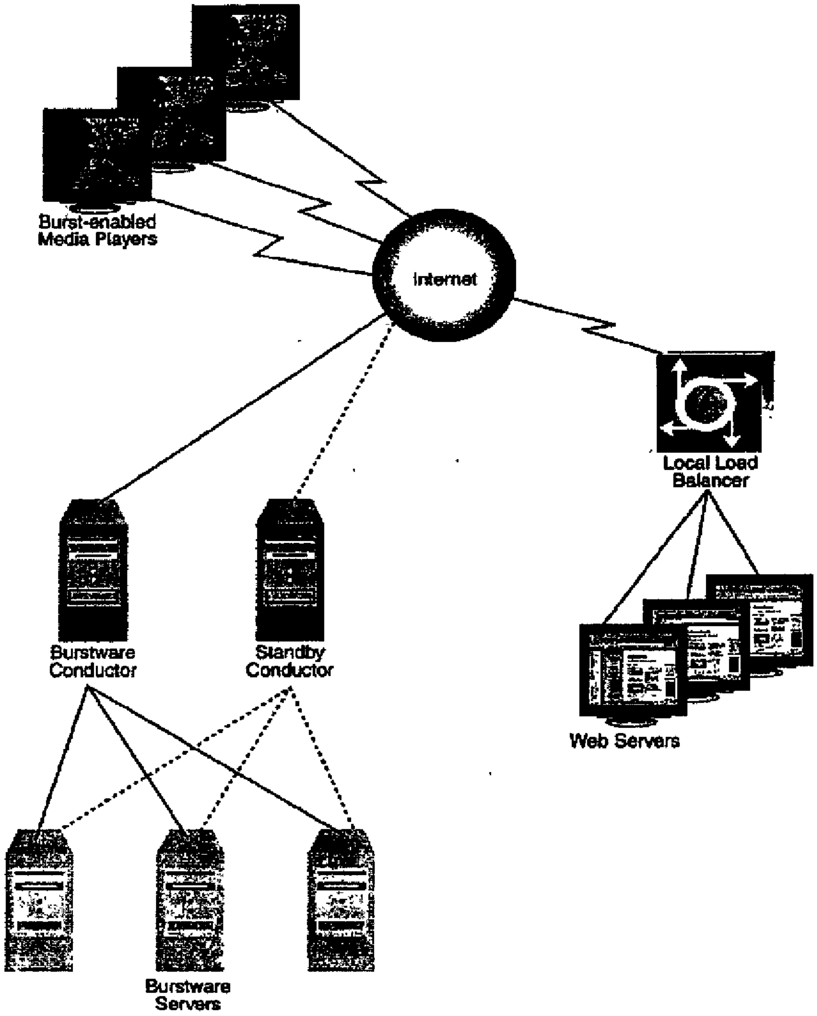


Figure 3: Burstware and a local load balancer

External Authentication Systems

Burstware provides an access control interface that enables you to implement a custom access control mechanism or to interface with external authentication systems. The access control interface is part of Burstware Conductor. See the section, Burstware Access Control Overview, on page 199 for details.

Deployment Considerations

This section describes important factors to consider as you plan your Burstware deployment and is intended to help you with the nuts and bolts of configuration decisions.

Bandwidth and Bottlenecks

Bandwidth is the most important factor in deciding how many Burstware Servers to deploy and where to put them. Understanding how much bandwidth you have includes considering your network topology, how your network is segmented, and the location of multimedia-seeking clients in relation to your topology and potential network bottlenecks.

Different segments of your network may support different transfer rates. To the extent possible, locate Burstware Servers close to clients to minimize multimedia transfers across slower network segments.

If your clients are within your network and are requesting multimedia from various workgroups in distributed locations, your goal is to understand your demand profile. Where are the most significant multimedia consumers located? If they are on the other side of a bottleneck from your Burstware deployment, your configuration will not be optimal. You may have a high bandwidth connection for your Burstware Servers, but clients on lower-speed legs will be limited by the bandwidth of their local network connection.

Demand

Demand for Multimedia

In analyzing the potential load, consider the number of concurrent multimedia viewers you want to support, the encoded bit rate of the media files, the client's location, and how demand for media might change. If media rates are higher, you can support fewer concurrent users with a given amount of bandwidth.

If you anticipate periodic spikes, you may want to establish alternative configuration strategies—typical, versus high-demand settings—and reconfigure Burstware for periods when demand spikes.

For more specific information on configuring Burstware for your unique demand profile, call Burstware Customer Support, or see the customer support section of www.burst.com.

Competing Demands

In planning your Burstware implementation, it is important to also consider the non-multimedia load on your network from the many other applications you use to run and support your business.

Burstware allows you to allocate a portion of available bandwidth to Burstware-managed multimedia traffic. This is referred to as your "managed bandwidth." Burstware allows you manage your bandwidth on a per domain, per server, and per client basis. (A Burstware domain is a conductor, its standby, and the set of servers they manage.) This ensures that a given component, at any level, can only consume the amount of bandwidth allocated to it.

Assuming that Burstware and your other applications will be sharing the same bandwidth, the amount of bandwidth you allocate to multimedia as managed bandwidth is consequently unavailable to other traffic-generating applications. If the other applications sharing bandwidth with Burstware are mission-critical, take a conservative—and perhaps iterative—approach to allocating bandwidth to Burstware.

Reliability Requirements

Understanding your reliability requirements is key to planning your Burstware configuration. Different organizations have different reliability requirements. Some companies may view service interruptions to internal users as more tolerable than service interruptions to external customers. Generally, the cost of downtime can be measured in either lost productivity or revenue.

If your Burstware environment will deliver self-paced training to employees, service interruptions to a client may be considered a nuisance, but tolerable. Alternatively, if your Burstware environment will serve video product demonstrations on your e-commerce site, service interruptions to clients may translate to lost revenue.

Burstware offers powerful reliability features. Burstware provides automatic failover if a component becomes unavailable for any reason. In addition, you can configure your environment to ensure no single point of failure.

While Burstware supports configurations that prevent single points of failure and enable failover, it is possible to configure leaner Burstware environments that do not provide these reliability features.

If your application requires no single point of failure, note the following:

- A Burstware deployment with no single point of failure requires a standby conductor and a standby server in each domain

  For more information on standby servers and failover, see the section, Load Balancing and Failover with Available and Standby Servers, on page 38.

- Try to locate a Burstware component—a conductor or a server—that provides failover for another component in a separate location from the component it backs up

  For example, when possible, locate a standby Burstware Conductor on a separate machine, network segment, or power grid from the active conductor.

Load Balancing and Failover with Available and Standby Servers

Burstware includes extensive load balancing and failover protection to optimize your application's efficiency and reliability, using a system of available and standby servers. Available servers participate in load balancing and failover. Standby servers participate in failover only.

A conductor routes client media file requests to the available servers in its domain, balancing client load among the available servers. When an available server fails, a standby server takes over and becomes available. Which of these two statuses a conductor confers on a server depends on a number of factors, including:

- The value of the conductor's **licensedBandwidth, licensedNumClients** and **licensedNumServers** parameters
- The server's position in the conductor's server list
- The value of each server's **managedBandwidth** parameter

To provide optimal load balancing and failover, Burst.Com recommends that you set **managedBandwidth** for each server in a conductor domain to the same value, and that the sum of the servers' **managedBandwidth** values somewhat exceed the conductor's **licensedBandwidth**. For example, if the **licensedBandwidth** for a conductor domain is 250 Mbps and the conductor's server list contains six servers, all six should have **managedBandwidth** values of 50 Mbps. In this configuration, the first five servers on the list will be available and the sixth server on the list will be a standby server. (Of course, there are other factors that help determine the best **managedBandwidth** setting for a server. See the section, Setting Managed Bandwidth, on page 45 for details.)

The technote called "Configuring Servers for Load Balancing and Failover", on Burst.Com's website, www.burst.com, provides more detailed information to help you configure servers for load balancing and failover.

Content Considerations

This section describes considerations related to content management in a Burstware environment.

Replication

For the current version of Burstware, Burst.Com recommends that each Burstware Server access an identical, replicated set of multimedia content. Full replication is required to ensure that all content is available in a failover scenario.

For this reason, you should ensure that when new or changed content is introduced, all Burstware Servers are updated. A variety of commercially available products can assist you with this process, ranging from shareware to global content distribution systems. Contact your Burst.Com sales representative for recommendations.

Play Rates and the Client's Available Bandwidth

As you plan your media library, take into account the range of client connection speeds you need to support.

Burstware's buffer management strategy is designed to ensure that the client buffer always contains sufficient data so that play can continue should network service degrade or be interrupted. However, if the client's average available bandwidth is less than the media play rate, interruptions in play may result.

What Happens When Play Rate Exceeds Average Available Bandwidth

Burstware tolerates variations in available bandwidth or connection speed, as long as the average available bandwidth during the connection session is greater than the encoded bit rate of the media file. While it is acceptable for the connection speed to fall below the play rate from time to time during a session, overall, the average connection speed should exceed the encoded bit rate of the media file.

If a client connection is, on average, *slower* than the play rate of the media, the media file will play from the buffer faster than the network connection allows the buffer to be provisioned, so the buffer level will decrease.

When the client buffer has been depleted, play will stop. Once the buffer has been refilled to the level specified by the **SecondsOfDataToBuffer** parameter, play will restart.

Prevent Problems with a Range of Encoding Rates

To avoid the problems that can result from serving high play rate files to low speed client connections, plan for a range of client connection speeds. Ensure that each of your media files is available in a range of encoded rates, including one that supports users with very slow connections. Users can select the appropriate version, based on their local connection speed, or you can use a bandwidth sniffer to choose for them.

For best results, an end user should be served media files with an encoding rate no higher than the average bandwidth between the client and the server. This allows you to successfully serve multimedia to users with a variety of connections speeds.

Re-Encode Media for Worst Case Connections

Re-encoding media for low speed connections enables you to meet the needs of clients with low bandwidth, while maintaining quality. Tools are available for taking an existing video and re-encoding it at a different bit rate.

Supporting low bandwidth viewers usually involves making some trade-offs. By re-encoding the media, you make the decisions, instead of requiring end users to suffer through unexpected events. If you re-encode a media file, you can decide on an appropriate compromise. For one file, it might be best to sacrifice sound quality by encoding in mono sound, thus allowing better video quality (more frames per second). For another file, the reverse might be preferable.

Play Rates and Content Naming Conventions

When a burst-enabled media player plays content, it supplies the encoded bit rate of the media file to the conductor. The encoded rate information is key to the conductor's multimedia load balancing function. It is also important for ensuring the highest quality viewing experience for the end user.

If the user browses and plays media interactively, the web page author must provide the media file's encoded bit rate in the HTML code that embeds the player in a web page, or supply it as part of the URL in the File > Open or File > Open URL dialog of the player.

For information on determining the encoded bit rate of your media file, refer to the section, Data Flow in Chapter 9, The Burstware Bridge Monitor.

Burstware and Security

Burstware's built-in security features are intended to prevent unauthorized users from modifying or monitoring a Burstware implementation.

Domain Security

Burstware provides password protection at the domain level. When you assign a password to a Burstware domain, you are restricting access to all servers and conductors in the domain.

By password-protecting a domain, you can:

- Prevent unauthorized conductors and servers from becoming part of your Burstware deployment
- Isolate Burstware domains from one another

  For example, you can prevent a conductor from one domain and a conductor from another domain from inadvertently managing the same server.
- Prevent unauthorized remote monitoring of your Burstware deployment (see the section, The Burstware Console, below)

See the section, Password-Protecting a Burstware Domain, on page 195 for details.

The Burstware Console

The Burstware Console, an application that provides remote monitoring of servers and conductors, requires a password before connecting to the components it monitors. A Burstware Console must use the same password as any conductor or server it attempts to monitor.

The Burstware Console is included as part of Burst.Com's Professional Services CD-ROM. For more information on the Burstware Console, contact your sales representative.

Access Control

Burstware Conductor's access control feature provides mechanisms for protecting content from unauthorized access. Requests for content can be fulfilled or denied, based upon predefined and/or dynamically defined rules or policies.

Burstware's default access control allows any user to have access to any content file. You can specify and manage access rules by using an access control module available from Burst.Com Professional Services, or you can link Burstware to an external authentication system, using the Burstware Conductor's access control interface. See the section, Burstware Access Control, on page 197 for more information.

Burstware and Virtual Private Networks

If your Burstware deployment requires stringent security, we recommend that you run all conductors, servers, and consoles as part of the same Virtual Private Network (VPN). This ensures encryption of all communication between Burstware components and prevents unauthorized users from obtaining the domain password.

Configuration Decisions

This section summarizes the major configuration decisions you must make before implementing Burstware.

Servers

How many Burstware Servers should I deploy and where do they go?

Bandwidth is Everything

Bandwidth availability and the location of clients determine the best location for Burstware Servers. If you have an existing multimedia distribution environment in place, you have already made some decisions about the optimal location for your multimedia content, based on bandwidth availability. Depending on the current performance of your environment, you may want to revisit these decisions.

Generally, where you put your active Burstware Servers should correspond to where you would put your content if you didn't have Burstware. However, Burstware enables a much simpler configuration than alternative streaming solutions, which are far more vulnerable to bandwidth limitations and latency issues. Streaming solutions often work by minimizing the distance information travels over the network—between server and client—but resulting in extensive content replication. If you are replacing an existing streaming solution, your overall content-serving environment may be significantly simplified with Burstware.

Burstware doesn't increase bandwidth—it makes the most of it—and uses it in innovative ways to provide high quality and maximize throughput. If your multimedia is optimally located for serving it without Burstware, it may already be optimally located for a Burstware deployment.

Deploy Separately from HTTP Server

To avoid resource contention, install Burstware Servers on a separate system from your web server. This prevents potential port conflicts and isolates multimedia transmission from HTML traffic on your site.

Separate Burstware Servers for Failover

To provide a failover strategy, locate identically configured servers on separate machines and, if possible, on separate network legs.

> NOTE: For this Burstware release, Burst.Com recommends that all servers manage identical multimedia files.

Locate Servers Close to Demand

Locate servers as close as possible to demand. For example, in an Intranet environment, if users on a particular subnet watch a set of training videos, place the server storing those training videos on that subnet.

You can easily and dynamically change the number and location of servers as the demands of your application change.

Conductors

How many conductors should I deploy and where do they go?

Choose Conductor Locations Carefully

Clients in a Burstware environment are unaware of where Burstware Servers are, so the impact of adding servers or changing their location is minimal to clients. However, clients do know the address of Burstware Conductors. The conductor is the clients' point of entry to the Burstware environment, so changing the address of a conductor affects a client's ability to connect and to receive multimedia data.

Changes to a conductor's location require changes to the HTML for an embedded player, to the multimedia file URL for a standalone player, and to the license key for that conductor. For this reason, it is best to plan your conductor locations carefully, and to minimize changes to the conductors' locations.

Run a Standby Conductor

To provide a failover strategy, locate two identically configured conductors on separate machines and, if possible, on separate network legs.

For global implementations, or geographically dispersed environments, consider implementing a Burstware domain in each Internet point of presence (PoP) where you will serve clients, and a global load balancer to route client requests to the appropriate PoP.

Setting Managed Bandwidth

How much bandwidth can I devote to multimedia?

Managing Bandwidth

The Burstware Server's **managedBandwidth** parameter provides a "throttle" that lets you control the impact of multimedia video and audio data on the network. The value you assign to **managedBandwidth** limits the total bandwidth available to the Burstware Server for delivering video and audio. The value you choose should be based on expected demand on the server.

If the server has insufficient bandwidth to accept a new connection (based on the value of **managedBandwidth**), it rejects a player's connection request. The rejected connection request is forwarded to the next server in the conductor's server list. The server then applies its own logic as to whether it can accept the request.

NOTE: Do not set managedBandwidth **arbitrarily high to prevent rejection of player requests; doing so can diminish the quality of the viewing experience.**

Choosing a Value for managedBandwidth

Burst.Com tests indicate that Burstware can easily manage 75 Mbps of bandwidth per 100 Mbps NIC card.

The default **managedBandwidth** setting is 50 Mbps, which is conservative for most networks with a capacity of 100 Mbps or more. You will likely need to adjust this value to meet your actual network capabilities. In many cases, the number might be 75 percent of the network card speed.

The value you choose for **managedBandwidth** depends on:

- How much bandwidth your application needs

- What your system can support
- Whether an HTTP server is running on the same machine as the Burstware server
- How much managed bandwidth is specified by your Burstware license agreement, and the number of servers that share this bandwidth

Meeting Your Application's Bandwidth Requirements

In the worst case, such as in standard streaming solutions, you would allocate bandwidth equal to the number of simultaneous users your application must serve, multiplied by the encoded bit rate (in Mbps) of the multimedia files the server is delivering. However, the efficiencies provided by Burstware allow you to serve the same number of users with significantly less bandwidth than in a conventional, streaming solution environment.

Burst.Com's benchmarks and simulation tools can help you determine a starting point for **managedBandwidth**. Contact Burst.Com technical support for assistance, if necessary.

Media Player Buffer Considerations

Increasing the size of the client-side Burstware buffer allows you to support more users with the same amount of managed bandwidth. Burstware's disk caching feature lets you cache to disk buffers on player machines, which generally can be much larger than memory buffers.

For more information on player buffers and disk caching, see the section, About the Burstware Buffer. on page 156.

The Limitations of Your Network and Disk Subsystem

The value you choose for **managedBandwidth** is also influenced by your network's bandwidth and equipment and your server's disk subsystem.

The value of **managedBandwidth** should never exceed the:

- Capacity of your network
- Sustained transfer rate of your server's disk subsystem

Either of these scenarios can result in degraded viewing experiences.

HTTP Server and Burstware Server on the Same Machine

If you co-locate a Burstware Server on the same machine as an HTTP server, they will contend for bandwidth on the machine's network interface card (NIC). As mentioned previously, Burst.Com advises against putting Burstware Servers and HTTP servers on the same machine.

If you must install a Burstware Server on the same machine as an HTTP server, it will influence how high you should set **managedBandwidth**. In such cases, plan on allocating a smaller percentage of the bandwidth available on the machine to multimedia transmission.

You may need to test repeatedly to determine the optimal **managedBandwidth** setting, especially if the HTTP server is heavily used. Because of the variety of activities and the range of traffic volumes a web server may support, Burst.Com is unable to provide general recommendations for **managedBandwidth** values if you put a Burstware Server on the same machine as a web server.

CHAPTER 4

Deploying Burstware on Your Network

This chapter presents and discusses several example Burstware deployments.

The information in this chapter will be more useful to you if you have already made some preliminary decisions about your Burstware configuration, such as the:

- Number of Burstware Servers and Burstware Conductors your deployment will have
- Optimal locations on your network for Burstware components
- Quantity of bandwidth allocated for multimedia delivery

Before adopting a deployment strategy for your own situation, you should read Chapter 3, Planning a Burstware Deployment.. That chapter contains information about the impact of your current environment and requirements on the deployment planning process, and helps you in making your deploying decisions.

Elements of Your Burstware Environment

The hardware in your Burstware environment consists of the machines on which the Burstware products run and the network equipment that enables them to communicate.

For a summary of machine requirements, see the section, Burstware System Requirements, on page 62.

The following table describes some network elements you can use in your Burstware environment.

Table 1: Network Equipment

Device	Characteristics	Examples
100Base T Hub	Star-wired fast Ethernet LAN segment providing 100 Mbps of shared bandwidth to all attached nodes. Some hubs support a mix of 10 and 100 Mbps links to attached nodes.	Cisco FastHub 300 3Com SS Hub 100 Intel 200 series Stackable Hub
Workgroup Switch	10/100 Mbps Ethernet/Fast Ethernet switch, providing either 10 or 100 Mbps connectivity into a packet switch with 5+ gigabits per second switching capacity.	Foundry FastIron II Switches Cisco 2900 Family 3-com SS II 3300 BayNet BayStack 350/450 Intel Express 500 series
Multilayer Switch	10/100/1000 Mbps intermediate switch/router that routes traffic between multiple workgroups. Can support Gigabit Ethernet links to backbone routers and high speed servers.	Cisco 3000, 5000 3-com SS 3000, SS 3300 BayNet BayStack 350T/450T Intel Express 550T/F switch Intel Gigabit Express switch
Router w/ Silicon	Enterprise class backbone router with large wire-speed gigabit routing capacity.	Foundry BigIron Switching Routers Cisco 5500, 7500, 8500, 12000 Baynet/Nortel BLC, BCN

 ATM Router	Variation of Router with Silicon; supports ATM WAN links and can interconnect a distributed corporate intranet.	Cisco 7500, 8500, 12000 with AIP Baynet System 5000
 Ethernet Hub	Star-wired Ethernet LAN segment providing 10 Mbps of shared bandwidth among all attached clients and servers. *Although this type of hub is commonly encountered in networks, Burst.Com recommends upgrading to a 100Base T hub.*	HP 10Base T Hub 16M 3Com SS Hub 10

Deployment Scenarios

Each of the following sections presents a scenario for deploying Burstware:

- Workgroup with moderate competing demand on the network from non-multimedia applications
- Workgroup with heavy competing demand
- Multiple workgroups with heavy competing demand

Each section proposes a network configuration and suggests alterations to the standard configuration for the server or the player, where appropriate. For examples of Internet deployments of Burstware, contact Burst.Com.

Workgroup with Moderate Competing Demand

In this scenario, one workgroup requires delivery of archived MPEG1 video. The local network infrastructure must also support light to moderate network traffic associated with email, database updates, and report generation. Examples of such environments include:

- An isolated call center
- A video production team

A sample Burstware configuration to support this environment is shown in Figure 4. Key aspects of this configuration are described below:

- A Burstware Conductor is running on the same machine as a single Burstware Server, which is linked to a 100Base T hub via a 100 Mbps link.

  If a 100Base T Hub supporting 10/100 Mbps links is used, only Burstware Server need be equipped with a 100 Mbps link.
- Managed bandwidth is 60 Mbps, allowing for moderate competition from other applications for network resources.
- Burst-enabled players can be linked to the hub via a 10 Mbps or 100 Mbps link according to the hub selection and the nature of the content. For MPEG1, a 10 Mbps link may suffice.

  If the content library also includes AVI content, a 100 Mbps link to the players is advisable.

- This configuration does not provide for failover, as it does not include a standby Burstware Conductor or a standby Burstware Server.

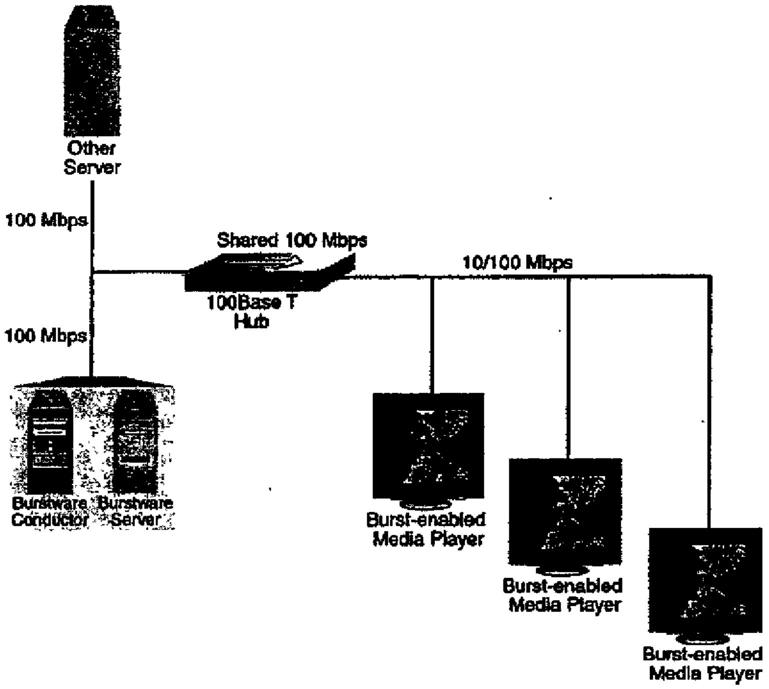


Figure 4: Workgroup with moderate competing demand

Workgroup with Heavy Competing Demand

This deployment scenario is suitable for a workgroup whose multimedia distribution needs are similar to those in the previous example, but whose network usage is heavy due to traffic generated by other application servers in the environment.

A sample Burstware configuration to support this environment is shown in Figure 5. Key aspects of this configuration are described below:

- The configuration shown in Figure 5 uses a switch, instead of a hub.

  The switch has a 500 Mbps switching capacity, which means it can handle significantly more traffic than the 100 Mbps that any single server or its network link can generate or absorb. Using a hub:

  - Increases scalability
  - Isolates video delivery from other network traffic
- There are two Burstware machines, one active and one standby, each running a Burstware Conductor and a Burstware Server.

  With a standby conductor and a standby server, this deployment supports failover.
- Managed bandwidth is 50 Mbps, leaving more bandwidth available for other applications competing for network resources.
- As in the first configuration, burst-enabled players can be linked to the hub via a 10 Mbps or 100 Mbps link according to the hub selection and the nature of the content.

  For MPEG1, a 10 Mbps link may suffice. If the content library also includes AVI content, a 100 Mbps link to the players is advisable.

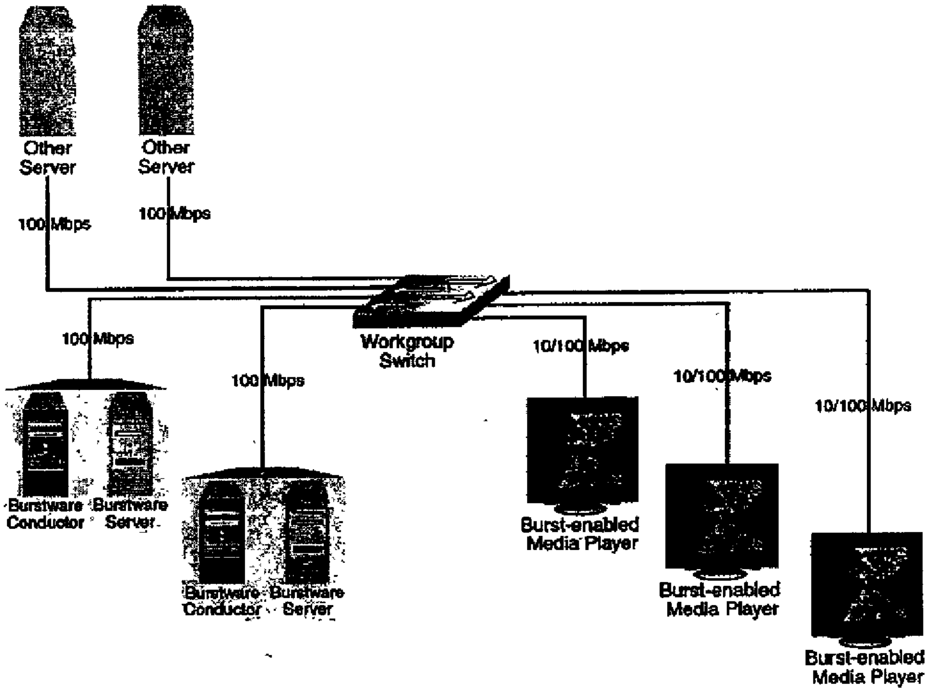


Figure 5: Workgroup with heavy competing demand

Multiple Workgroups with Heavy Competing Demand

In this scenario, multiple workgroups in a corporate intranet require access to multimedia. In addition, business applications generate significant network traffic. This demand profile might be found in:

- Networked call centers
- A video production environment with multiple teams
- Distance learning
- Global enterprise with scheduled corporate communications

A sample Burstware configuration to support this environment is shown in Figure 6. Key aspects of this configuration are described below:

- A high speed switch-router provides connectivity between workgroup switches and the backbone.

  The router could also serve as a natural point of attachment for gigabit servers—very powerful servers (comprising multiple SMPs, disk arrays)—with equally powerful network connections, such as Gigabit Ethernet.
- The primary conductor is running on a machine attached to the switch, removing it from any of the workgroups it serves.

  When contacted by a burst-enabled player, which can be located anywhere within the intranet, the conductor selects a Burstware Server based on the existing load on the servers.
- One machine, running a Burstware Server and a standby conductor, is attached to one of the workgroup switches.

  Two additional Burstware machines, one with an active Burstware Server, and one with a standby Burstware Server are attached to the other workgroup switch. With a standby conductor and a standby server, this deployment has no single point of failure and supports failover.
- Managed bandwidth is 75 Mbps.

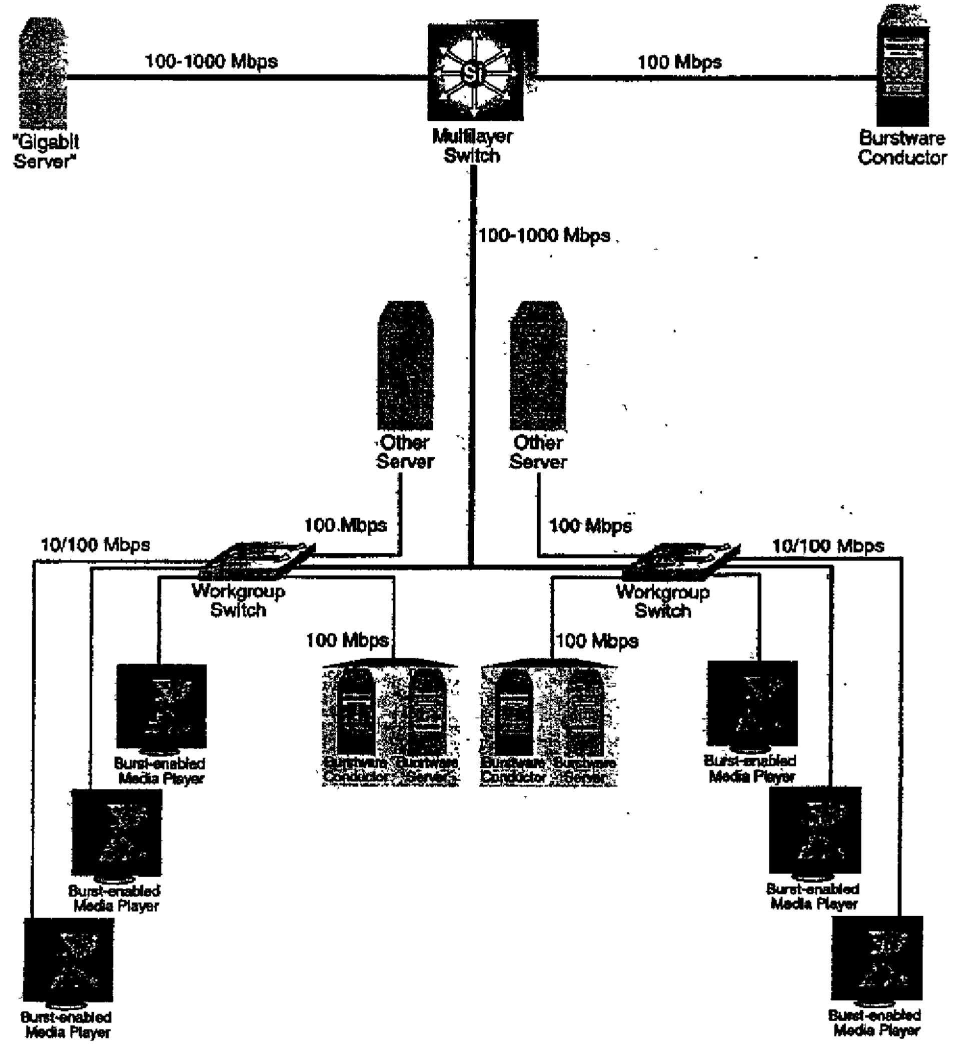


Figure 6: Multiple workgroups with heavy competing demand

Advanced Topics

Using a Burstware Component with Multiple Network Interface Cards

Burstware Conductors and Burstware Servers can be used on machines with multiple network interface cards (NICs).

There are two ways to configure a conductor or server for use with multiple NICs:

- One conductor or server listening on all NICs (default)
- One conductor or server per NIC

The first option requires no additional configuration. The second option requires you to edit the initialization file to specify the exact IP address at which the conductor or server listens.

Configuration Options for Multiple NICs

The following sections describe the two configuration options for multiple NICs.

One Conductor or Server Listening on All NICs

In this configuration, one conductor or server listens on all of the machine's IP addresses. The mapping of machine name to IP addresses is obtained from a local or Internet domain name server, or from a local host file. This is the default configuration; no action on the part of the administrator is required to accomplish this.

One Conductor or Server Listening on One NIC

In this configuration, one conductor or server listens on a specified IP address. You must add a **baseIPAddress** configuration parameter in the conductor or server initialization file to specify the IP address at which a conductor or server listens. Note that the IP address specified in the baseIPAddress parameter will override those specified in the host file or name service.

Here are the steps to configure an initialization file to listen on one IP address:

1. Open the initialization file using any text editor.

   The **burstserver.ini** file can be found in the **server** subdirectory of your Burstware installation directory. For example, if you installed Burstware in the default directory C:/Program Files/Burst/Burstware, the initialization file is C:/Program Files/Burst/Burstware/server/burstserver.ini. The **burstconductor.ini** file can be found in the **conductor** subdirectory of the Burstware installation directory in a similar fashion.
2. Add the following line to the initialization file:

   ```
   baseIPAdress = IP_address
   ```

   where IP_address is the IP address of the NIC on which you want the server or conductor to run.
3. Restart the server if it is already running.

CHAPTER 5

Installing and Starting Conductors and Servers

This chapter explains how to install and start Burstware Conductors and Burstware Servers on Windows, Solaris, and Linux. Information is presented in the following sections:

- Before You Install
- Burstware System Requirements
- Installing Burstware on Windows
- Starting Burstware on Windows
- Installing Burstware on Solaris and Linux
- Manually Starting Burstware on Solaris and Linux

See Chapter 7, Burst-Enabling Players. for instructions on installing the Burstware software to burst-enable Windows Media Player and Apple QuickTime Player.

Before You Install

Before you install Burstware, we recommend that you plan your Burstware environment, making such decisions as how many conductors and servers to install, how to distribute them on your network, and which machines to install them on.

NOTE: **As part of this planning process, fill out the installation worksheets included with this release. These worksheets are located in the docs directory on the Burstware CD-ROM.**

If you have not read the following chapters, we recommend you do so before continuing with the installation:

- Chapter 3, Planning a Burstware Deployment
- Chapter 4, Deploying Burstware on Your Network

Obtaining Burstware License Keys

Burstware employs license keys to ensure that Burstware products are used in a manner consistent with the terms of a particular license agreement. Typically, these terms are based on such factors as:

- Number of Burstware Servers the conductor can manage
- Total amount of combined bandwidth all the managed servers can use to deliver video
- Number of concurrent users
- Number of servers
- The expiration date of your Burstware license

Burstware Conductor enforces license keys. Each conductor requires a license key configured for the hostname of the machine on which the conductor has been installed.

To obtain your license key, contact your Burst.Com sales representative at sales@burst.com.

Burstware System Requirements

System requirements for Burstware components are discussed in the following sections:

- Burstware Bridge for Windows Media Player and Apple QuickTime Player Requirements
- Burstware Conductor Requirements
- Burstware Server Requirements

NOTE: Burstware Conductor and Burstware Server run on Windows NT, Linux, and Solaris. Burst-enabled players run on Windows NT, Windows 98, and Windows 95.

Burstware Bridge for Windows Media Player and Apple QuickTime Player Requirements

Platform	Windows NT 4.0 with Service Pack 3 or higher, or Windows 98, or Windows 95
Windows Media Player	Version 6.0
RAM	32 MB
Storage	300 KB More if using player disk caching
Browsers	Netscape 4.5 or higher Microsoft Internet Explorer 4 or higher
Processor	Required: Pentium-class processor (166 Mhz or greater) Recommended: Pentium II processor or better
Monitor	We recommend that you set monitors to display high quality color (either 16-bit or 32,000 colors)

Burstware Conductor Requirements

On Windows NT

Platform	Windows NT with Service Pack 3 or higher
RAM	128 MB
Storage	50 MB
Processor	Required: Pentium II processor or better (300 Mhz or greater)
Network Card	Ethernet 10Base T or faster

On Linux

Platform	Linux Red Hat 6.0
RAM	128 MB
Storage	50 MB
Processor	Required: Pentium II processor or better (300 Mhz or greater)
Network Card	Ethernet 10Base T or faster

On Solaris

Platform	Solaris 2.6 and 2.7
RAM	128 MB
Storage	50 MB
Processor	SPARC Ultra
Network Card	Ethernet 10Base T or faster

Burstware Server Requirements

Server system requirements are discussed in the following sections:

- Storage Requirements
- Processor and Memory Requirements
- Network Card and Operating System Requirements

Storage Requirements

Burst.Com recommends one of two storage solutions:

- A Redundant Array of Independent Disks (RAID) level 0 or 5 disk subsystem, supporting throughput of up to 100 Mbps
- A Storage Area Network (SAN) device with many machines clustered around it, supporting throughput of 1 gigabit per second (Gbps) or higher

It is possible to deploy and run a Burstware Server on a machine with slower disk drives, but your overall throughput will be limited by disk I/O speeds.

Processor and Memory Requirements

Processor and memory requirements depend on the value set in the managedBandwidth parameter, the number of simultaneous users you want to support, and the encoded bit rate of the multimedia files Burstware Server is delivering. Read the section, Setting Managed Bandwidth, on page 45 for help in choosing the best value for the **managedBandwidth** parameter.

These are a few guidelines:

- The greater the number of users serviced, the faster the processor and the greater the amount of memory required
- To configure conservatively, **managedBandwidth** should equal the number of simultaneous users multiplied by the encoded bit rate of the multimedia files the server is delivering

  When disk caching to large player buffers is enabled—where large means large enough to fit most or all of the multimedia file at once—the number of concurrent users can be as much as 50 percent higher.

The following processor and memory requirements are recommended when managedBandwidth is set to 75 Mbps:

For Windows NT and Linux

Encoded Bit Rate of Files	Simultaneous Users	Processor	Memory
1.5 Mbps	50 Up to 75 with large player buffers	Pentium II 400	256 MB
1.0 Mbps	75 Up to 112 with large player buffers	Pentium III 500	256 MB
250 Kbps	300 Up to 450 with large player buffers	Dual Pentium III 500 Processors	512 MB

For Solaris

Encoded Bit Rate of Files	Simultaneous Users	Processor	Memory
1.5 Mbps	50 Up to 75 with large player buffers	300 MHz Sparc	256 MB

1.0 Mbps	75 Up to 112 with large player buffers	400 MHz Sparc	256 MB
250 Kbps	300 Up to 450 with large player buffers	Dual 400 MHz Sparc	512 MB

NOTE: When a conductor and server are running on the same machine, combine their memory requirements.

Network Card and Operating System Requirements

For all platforms, use an Ethernet 100Base T network card.

The following table lists operating systems versions Burstware Server runs on, by platform:

Platform	Operating System Version
Windows NT	Windows NT with Service Pack 3 or higher
Linux	Linux RedHat 6.0
Solaris	Solaris 2.6 and 2.7

Burst.Com Can Help Determine Your Requirements

Given information about your application, hardware, and network, Burst.Com can help you determine how much memory and what type of processor your application needs, as well as the best value for managedBandwidth. Contact your Burst.Com sales representative at sales@burst.com.

Installing Burstware on Windows

This section explains how to install Burstware Conductor and Burstware Server on the Windows platform.

Preparing to Install

Burstware Conductors need to know the location of the Burstware Servers they manage.

A server's location is specified by the:

- Hostname or IP address of the machine running it
- Port number where it listens for requests

Before installing a conductor, note this server information.

In addition, before you install:

- Obtain a license key for each Burstware Conductor you are installing, as discussed in the section, Obtaining Burstware License Keys, on page 62

  Fill out and have available the *Installation Worksheet for NT*—worksheet_windows.pdf—included with this release in the following location:

  Windows/docs/worksheet_windows.pdf
- Exit all running applications, especially any existing versions of the Burstware components you plan to install
- If you are running either the Burstware Conductor or the Burstware Server as an NT Service, stop the Burstware Conductor Service or the Burstware Server Service as described in Stopping a Burstware Service, on page 76
- Note the name of the directory or directories of existing media files you want to access with Burstware
- On Windows NT, if you want to install Burstware components as NT Services, log in as Administrator or another account with administrator privileges

Installing Burstware Components

Accessing the Burstware CD-ROM

1. Place the Burstware CD-ROM in the caddy.

   If the installer does not start automatically, double-click the setup.exe icon in the CD-ROM's windows subdirectory.
2. Follow the instructions in the installer.

Providing General Information to the Installer

The installer first prompts you for general information about installing Burstware, including:

- Which Burstware components to install
- Whether you accept the terms of the Burst.Com license agreement
- The destination path—the directory where Burstware files are to be installed. The default path is C:\Program Files\Burst\Burstware.

  The installer creates a separate directory in the destination path for each component. For example, if you choose to install a Burstware Server, Burstware creates a server subdirectory.

The installer then prompts you for information specific to the components you are installing.

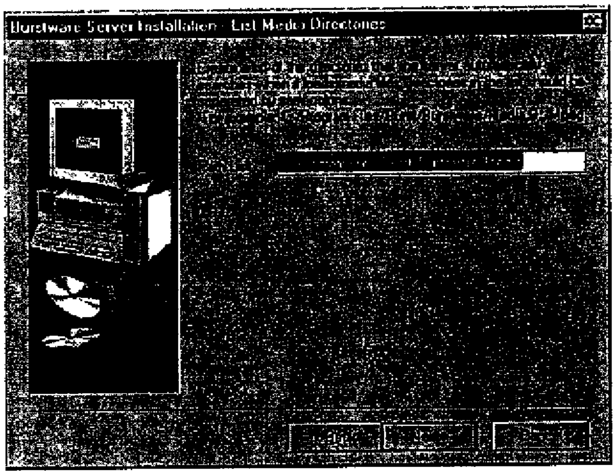
Burstware Server

When installing Burstware Server, the installer prompts you for the following information:

Media Directory Path

1. The media directory path is the directory where Burstware Server looks for media files at runtime.

   To store media files in multiple directories, specify multiple paths separated by semicolons. Be sure to specify the full path on the server machine, as in this example:

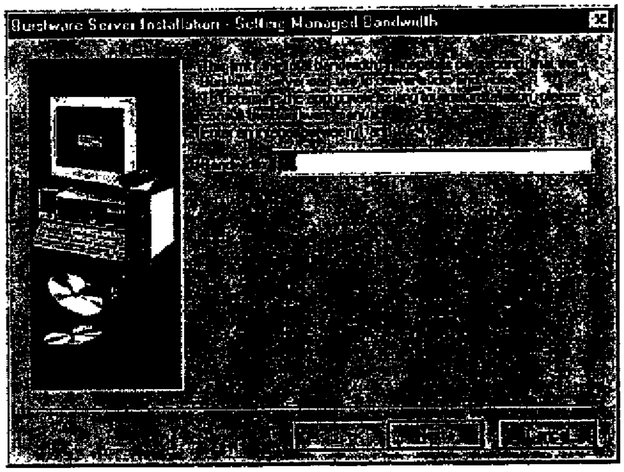


After entering a media directory path, click Next.

NOTE: For this Burstware release, you should place identical content on all server machines to ensure consistent video delivery in the event of a Burstware Server failure.

Server's Managed Bandwidth

2. If you are installing a Burstware Server, the installer prompts you for the server's managed bandwidth, in megabits per second (Mbps). The default value is 50. For help determining the proper managed bandwidth, see the section Setting Managed Bandwidth, on page 45.

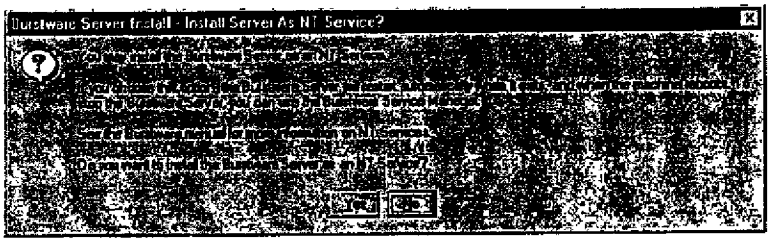


After entering a value for managed bandwidth, click Next.

Installing a Burstware Server as an NT Service

3. Installing a Burstware Server as an NT Service improves its availability. An NT Service starts automatically when the server machine boots up and restarts automatically if the server is terminated. If you choose this option, the service starts automatically when the Burstware installation is complete.

   To install a Burstware Server as an NT Service, click Yes.



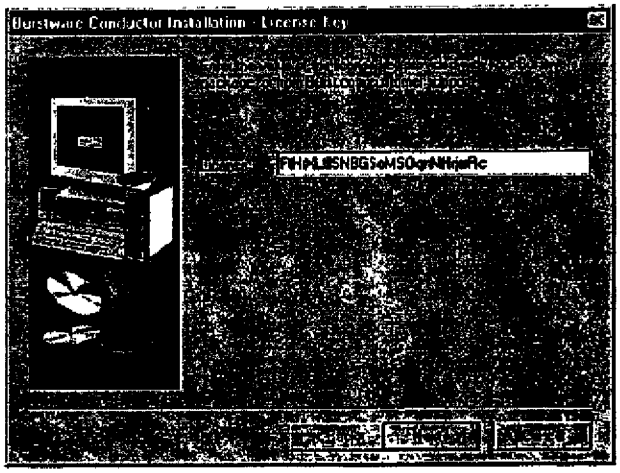
Burstware Conductor

When installing Burstware Conductor, the installer prompts you for the following information:

License Key

1. The conductor uses the license key to ensure that you use Burstware components according to the terms of your license agreement with Burst.Com. These terms include:
   - Licensed bandwidth
   - Name of the machine on which the conductor runs
   - Number of concurrent users
   - Number of Burstware Servers
   - Expiration date of your Burstware license

   Be sure to enter the license key exactly as provided to you by Burst.Com. The installer does not accept non-alphanumeric characters.



FtHpL8SNBGSeMS0qrNHjeRc

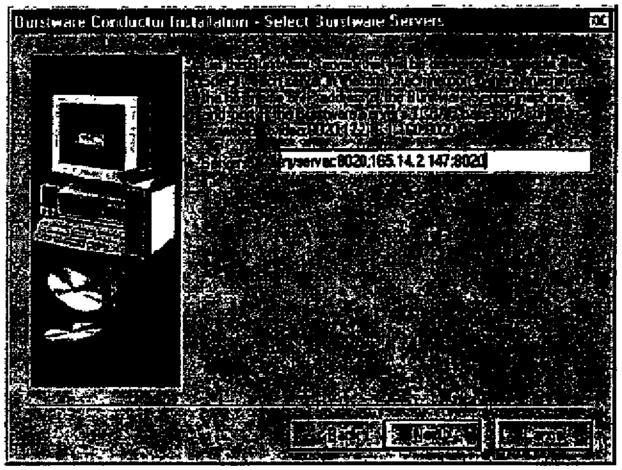
To obtain your license key, contact your Burst.Com Sales representative at sales@burst.com.

After entering a license key, click Next.

Server List

2. The Burstware Conductor sends the Burstware Server list to a burst-enabled player, which connects to one of these servers to play requested media files.

   The server list can contain one or more Burstware Server entries. Each entry has the form *hostname:port_number* or *IP_address:port_number*. The hostname or IP address is separated from the port number by a colon. Use semicolons to separate entries in the list, as in the following example:



yserver:8020;165.14.2 147:8020

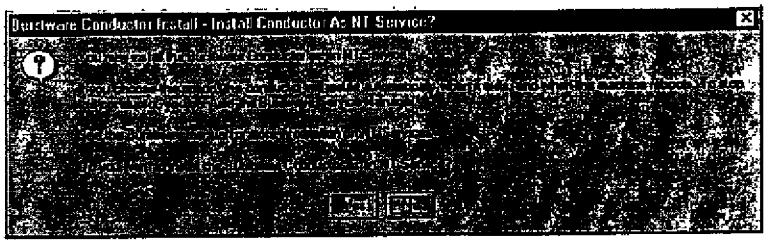
It is to your advantage to list multiple Burstware Servers. If a network problem occurs, a server machine is disconnected or shut down, or a burst-enabled player cannot reach a server, the player tries to connect to the next server on the list.

After entering a server list, click Next.

Installing a Burstware Conductor as an NT Service

3. Installing the conductor as an NT Service improves its availability. An NT Service starts automatically when the conductor machine boots up and restarts automatically if the conductor is terminated. If you choose

this option, the service starts automatically when the Burstware installation is complete.


Burstware Conductor Install - Install Conductor As NT Service?

To install a Burstware Conductor as an NT Service, click Yes.

Documentation and Supporting Files

The Burstware installation includes the documentation and supporting files listed below.

The following files are found in the docs subdirectory of your Burstware installation directory:

- burstware.pdf—Burstware documentation
- worksheet_windows.pdf—an installation worksheet
- readme.pdf—release notes
- BurstLicense.txt—company license agreement

NOTE: Use Acrobat Reader to open .pdf files.

If you installed the server, the following file is found in the media subdirectory of your Burstware installation directory:

- burstware1.mpg—a short sample media file

To access the documentation online, click on the Burstware Guide shortcut in the Programs/Burstware folder in your Start menu.

Starting Burstware on Windows

How you start a Burstware Server or Burstware Conductor depends on whether the server or conductor was specified as an NT Service during installation.

If the server or conductor was specified as an NT Service during installation, follow the instructions in the section, Using Burstware Service Manager, on page 75.

If the server or conductor was not specified as an NT Service, follow the instructions in the section, Starting Burstware Components That Are Not NT Services, on page 78.

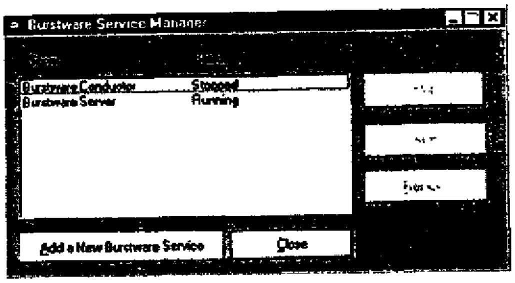
Using Burstware Service Manager

Use the Burstware Service Manager to add, remove, start, and stop Burstware Services on a particular machine.

If you did not install Burstware Conductor and Burstware Server as NT Services and want to add them as a Burstware Service or want to reconfigure an existing Burstware Service (to change its initialization file or memory usage), you can use the Burstware Service Manager.

NOTE: If you want to reconfigure an existing Burstware Service, you must first remove it from the Burstware Service Manager, then add it, then reconfigure it.

To launch the Burstware Service Manager, select the Burstware Service Manager entry in the Burstware Programs Folder shortcut in the Start menu. The following dialog box pops up:



Burstware Conductor Stopped
Burstware Server Running

The dialog box displays a list of the Burstware Services already added to the NT Services Registry and whether they are running or stopped.

From this dialog box, you can start, stop, remove, or add a service.

Starting a Burstware Service

To start a service, highlight the service and click Start.

Stopping a Burstware Service

To stop a service, highlight the service and click Stop.

Removing a Burstware Service

To remove a service, highlight the service and click Remove. The Burstware Service Manager prompts you for confirmation.

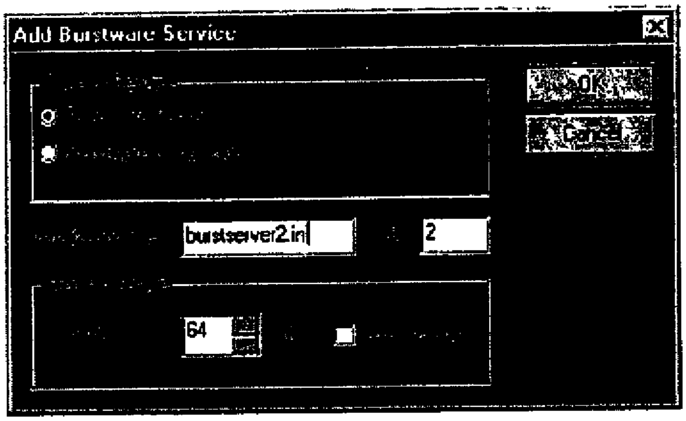
NOTE: You must stop a Burstware Service before removing it.

Adding a New Burstware Service

To add a new Burstware service, follow these steps:

1. Click Add a New Burstware Service in the dialog box.

The following dialog box pops up:



burstserver2.in

2

64

2. Select the type of service—Burstware Server or Burstware Conductor.
3. If you do not already have a unique initialization file for the new server or conductor, create one now. This can be done by copying an existing initialization file and changing the relevant parameters, such as basePort and logFile.
4. In the "Initialization File" field, type the name of the initialization file to be used by the Burstware Conductor or Burstware Server for which you are adding this service.

   Each Burstware Conductor and Burstware Server requires a unique initialization file. For more information, see the section, About Initialization Files, on page 95.
5. If other Burstware services of the same type are already running on this machine, enter a number in the ID field to differentiate the new service from services already running.

   For example, if a service called "Burstware Server 1" is running, enter the number 2 in the ID field. When the new service starts, it will have the unique name of "Burstware Server 2".

   If no similar services are installed, leave the ID field blank.
6. If you want to use a memory size other than the default, in the Memory Usage area, uncheck the "Use Default" box and choose a memory size.

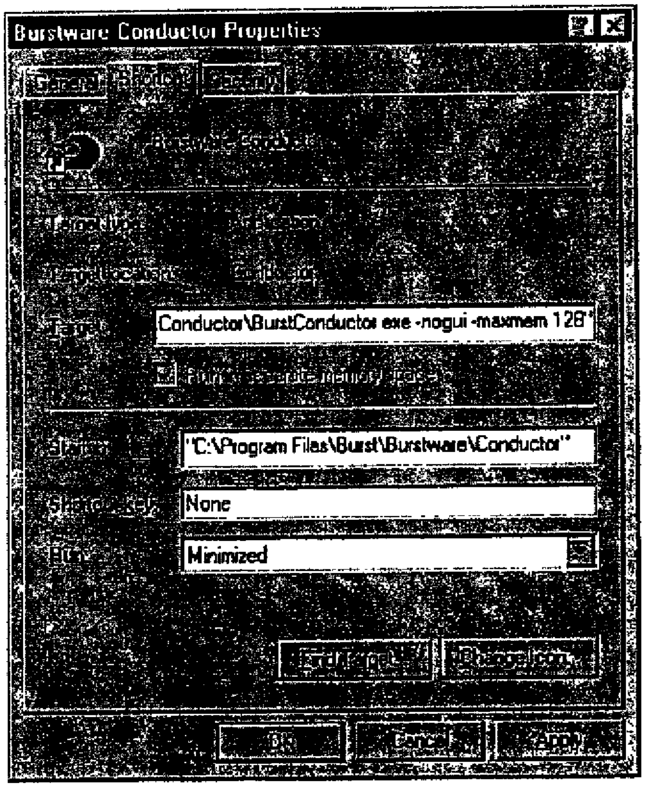
7. Click OK.

Starting Burstware Components That Are Not NT Services

The installer places a shortcut on the desktop for Burstware Conductor and Burstware Server. The installer also adds a Burstware Programs Folder shortcut to the Start menu and places component shortcuts in this folder.

To start Burstware Conductor or Burstware Server manually, double-click the desktop shortcuts or select the shortcuts via the Start menu.

You can also include optional configuration parameters by adding them to the Target field of the Properties dialog.



Burstware Conductor Properties

Target: Conductor\BurstConductor exe -nogui -maxmem 128"

Start in: "C:\Program Files\Burst\Burstware\Conductor"

Shortcut key: None

Run: Minimized

The syntax for the command line is the following:

```
burstserver.exe | burstconductor.exe [-help] [-ini
<iniFile>] [-minmem <memSpec>] [-maxmem <memSpec>]
[-nogui]
```

For example, to set the minmem parameter to 48, and maxmem parameter to 128 MB:

```
burstserver.exe -minmem 48 -maxmem 128
```

Command Line Parameters

help

The help parameter prints out the full command-line syntax and a brief explanation of each parameter and then exits with a non-zero status.

ini

The ini parameter directs the startup script to an initialization file other than the default.

nogui

The nogui parameter starts up a Burstware Conductor or Burstware Server without displaying its GUI. For example, you may be starting up a conductor or server on a Linux or Solaris machine not running XWindows. Or, conductors and servers may be running on machines without monitors.

You can remotely monitor conductors and servers—including those running with the nogui parameter—with the Burstware Console application, which is included on Burst.Com's Professional Services CD-ROM. For more information, contact your sales representative.

minmem

The minmem parameter specifies the initial amount of working memory the Burstware Server or Burstware Conductor executable uses, in MB. The default values are:

- Conductor- 16 MB
- Server- 48 MB

maxmem

The maxmem parameter specifies the highest amount of working memory that the Burstware Server or Burstware Conductor executable is allowed to use, in MB. The default values are:

- Conductor- 16 MB
- Server- 48 MB

About Working Memory

The amount of working memory needed is related to the number of concurrent connections that can be supported. Server and conductor memory usage tends to grow and shrink based on the number of concurrent connections the conductor or server is supporting.

There are a number of advantages to setting **maxmem** and **minmem**:

- Setting maximum memory usage can prevent excessive swapping by the virtual memory manager and thereby prevent serious performance degradation
- When serving hundreds of users, setting the **maxmem** parameter to a higher value allows the server or conductor to better handle the load
- When serving fewer users, setting the **maxmem** parameter lower than the default allows the server or conductor to conserve memory

Generally, the server needs more working memory than the conductor since the server performs more memory-intensive processing, and its connections last longer.

In addition to the default values for **minmem** and **maxmem**, the conductor and server use a certain amount of additional memory to carry out their processes. The conductor uses approximately 4MB in addition to the working memory; the server uses approximately 7MB.

Installing Burstware on Solaris and Linux

This section explains how to install Burstware Conductor and Burstware Server on the Solaris and Linux platforms.

Preparing to Install

Burstware Conductor needs to know the location of the Burstware Servers it manages. A server's location is specified by the:

- Hostname or IP address of the machine running it
- Port number where it listens for requests

Before installing Burstware Conductor, note the location of each Burstware Server.

In addition, before you install:

- Obtain a license key for each Burstware Conductor you are installing. See the section, Obtaining Burstware License Keys, on page 62.
- If you are installing a Burstware Server on a Unix machine, set the number of files that a process can have open to 1024 or higher, based on the number of concurrent connections you want to support. See the section, Setting Number of Files a Unix Process Can Have Open, on page 82 for details.
- Fill out and have available either the *Installation Worksheet for Solaris*—included in this release in the following location on the Burstware CD-ROM:

  /Burstware/solaris/worksheet_solaris.pdf

  or the *Installation Worksheet for Linux*—included in this release in the following location on the Burstware CD-ROM:

  /Burstware/linux/worksheet_linux.pdf
- Exit all running applications, especially any existing versions of the Burstware components you plan to install.
- Choose an installation directory and verify that it meets the storage requirements for the conductor and server. See the section, Burstware System Requirements, on page 62.
- If you want the Burstware components to run as daemons, log in as "root."
- Note the name of the directory or directories of existing media files you want to access with Burstware.

Setting Number of Files a Unix Process Can Have Open

Burstware Servers use 3 file descriptors for each client connection. If you set the file descriptor hard limit to 1024, you can expect about 330 concurrent connections. On Linux, to set the hard file descriptor limit above 1024, you typically must rebuild the kernel.

If this parameter is set too low, a heavily loaded server may not be able to open a media file for a new client, and the following error will appear in the server log: "ioRead: Error on read from file." See the Solaris documentation on the ulimit -n command for help setting this parameter.

Installing Burstware Components

Accessing the Burstware CD-ROM

1. Access the Burstware CD-ROM in the appropriate manner for your operating system.

Launching the Installation Script

1. On the Burstware CD-ROM, change your working directory to the directory corresponding to your platform type, such as Burstware/Linux for the Linux platform.
2. Launch the installer with the following script:

```
% ./install.sh
```

The Burstware installer starts.

NOTE: **If the operating system type is invalid, the installer aborts.**

Providing General Information to the Installer

The installer is a command-line interface. To answer installer questions, you can:

- Accept the default answer provided by pressing Enter
- Type an answer and press Enter
- Accept the default in a yes/no question by noting which letter in the answer is capitalized, and pressing Enter. For example, in an answer where the choices are yes or No, No is the default, because the initial "n" is capitalized.

The installer prompts you for general information about installing Burstware, as discussed in the next sections.

License Agreement

1. Confirm that you agree to the terms of the license agreement. Use the space key to page through to the end of the license agreement.

   Type agree to accept the terms of the license agreement, or type refuse to abort the installation. Press Enter.

Root User Verification

2. If you are not logged in as "root", Burstware warns you that:
   - Burstware must be placed in a directory for which the current user has write permission
   - Burstware cannot configure the installed Burstware processes as daemons. See the section, Make Burstware Run as a Daemon, on page 87.

   After displaying the warning, the installer asks whether to proceed. Type yes to proceed, or no to abort the installation. Press Enter.

Selecting the Burstware Components to Install

3. Decide whether to install a Burstware Conductor.

   Type Yes to install a conductor or No to proceed without installing a conductor. Press Enter.

4. Decide whether to install a Burstware Server.

   Type Yes to install the server or No to proceed without installing the server. Press Enter.

NOTE: If you elect to install neither the conductor nor the server, the installer aborts.

Installation Directory Name

5. Choose a name for the Burstware installation directory.

   The default installation directory is /usr/local/burst/burstware. Either accept this default by pressing Enter, or type another installation directory location and then press Enter.

NOTE: If the Burstware installation directory you choose already exists, the installer asks whether you would like to keep your existing Burstware installation. If you choose to keep it, you will be prompted for a directory location for the new installation. The installer will not delete your existing installation.

Providing Conductor and Server Information

The installer then prompts you for information about the Burstware Conductor and Burstware Server:

Conductor License Key

1. If you are installing a conductor, the installer prompts you for a license key. The conductor uses the license key to ensure that you use Burstware components according to the terms of your license agreement with Burst.Com. These terms include:
   - Licensed bandwidth
   - Name of the machine on which the conductor runs
   - Number of concurrent users
   - Number of Burstware Servers
   - Expiration date of your Burstware license

   Be sure to type the license key exactly as provided to you by Burst.Com. The installer does not accept non-alphanumeric characters.

   To obtain your license key, contact your Burst.Com Sales representative at sales@burst.com.

NOTE: If you do not know the license key and want to complete an installation, you can type any combination of letters and numbers for the license key. Remember to add the license key to the conductor initialization file later, as the conductor will not start without a valid license key.

After entering the license key, press Enter.

Server List

2. If you are installing a conductor, the installer prompts you for a list of Burstware Servers. The conductor sends this list to a burst-enabled

player, which connects to one of these servers to play the requested media files.

The list can contain one or more server entries. Each entry has the form *hostname:port_number* or *IP_address:port_number*. The hostname or IP address is separated from the port number by a colon. Use semicolons to separate entries in the list, as in the following example:

```
server1:8086;linux01:8018
```

NOTE: If you are installing a new Burstware Server during this installation and include it in this server list by its hostname, the port number you specify for the new server becomes the value of the server's basePort parameter in the server initialization file.

After entering the server list, press Enter.

Media Path

3. If you are installing a Burstware Server, the installer prompts you for the media directory or directories where the server looks for media files. The default media path is the name of the installation directory, followed by "/media". Separate multiple directories with semicolons.

   Be sure to specify the full path on the server machine, as in this example:

   ```
   % /usr/local/burst/burstware/media/mpeg
   ```

   Since media files can be very large, the directories you specify should be on devices with large amounts of free space.

   After entering the media path, press Enter.

Managed Bandwidth

4. If you are installing a Burstware Server, the installer prompts you for the server's managed bandwidth, in megabits per second. The default value is 50.

   For help determining the proper managed bandwidth, see the section, Setting Managed Bandwidth, on page 45. After entering a value for managed bandwidth, press Enter.

Make Burstware Run as a Daemon

5. If you are logged in as root, the installer asks whether all installed Burstware components should be run as daemons.

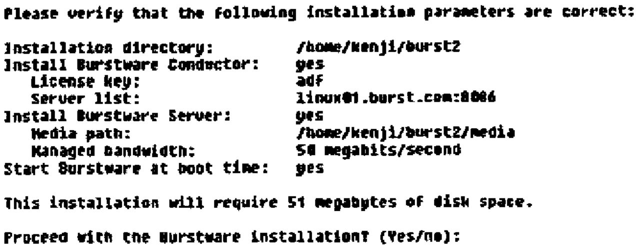
   Installing a server or conductor as a daemon improves its availability. A server or conductor running as a daemon starts automatically when the host machine boots up. Burstware components installed as daemons run without a GUI. Type either yes or no. Press Enter.

NOTE: If you choose Yes, all Burstware components will run as daemons.

Completing the Installation

The installer displays a summary of the installation configuration constructed from answers provided during installation, including how many megabytes of disk space the installation requires.

I



```
Please verify that the following installation parameters are correct:

Installation directory:         /home/kenji/burst2
Install Burstware Conductor:    yes
   License key:                 adf
   Server list:                 linux01.burst.com:8086
Install Burstware Server:       yes
   Media path:                  /home/kenji/burst2/media
   Managed bandwidth:           50 megabits/second
Start Burstware at boot time:   yes

This installation will require 51 megabytes of disk space.

Proceed with the Burstware installation? (Yes/no):
```

NOTE: If there is insufficient space to install Burstware, the installer displays a warning message. At this time, you can free the required disk space before continuing with the installation.

1. Type Yes to confirm the installation configuration and install the conductor or server, or type No to abort the installation if you want to change the installation configuration details or cannot free up enough disk space.

   If you typed Yes, the installer installs the appropriate files, and displays the message "Burstware Release 2.0 has been installed successfully."

Additional Files Created by Installation

In addition to the files for the Burstware components, the installer creates two log files in the installation directory after a successful installation:

- **install.log**—contains the description of the installation configuration for the current installation
- **install.rsp**—contains the responses you gave during installation

Using the Responses File for an Identical Installation

If you want to perform an identical installation in the future, you can use the **install.rsp** file to provide the same responses to the installer questions. To use the **install.rsp** file in this way follow these instructions:

1. Copy **install.rsp** from the installation folder and place it in another location, for example **/temp/install.rsp**.
2. On the same machine or another machine, access the Burstware installation CD-ROM, and change the working directory to the directory corresponding to your platform type, such as Burstware/Linux for the Linux platform.
3. Invoke the **install.rsp** file as in the following example:

   ```
   % ./install.sh < /temp/install.rsp
   ```

   The installer uses the **install.rsp** file to fill in responses to install questions and perform an identical installation.

Documentation and Supporting Files

The Burstware installation includes the documentation and supporting files listed below.

The docs directory of your Burstware installation directory contains:

- **burstware.pdf**—Burstware documentation
- **worksheet_solaris.pdf**—an installation worksheet (only if installing under Solaris)
- **worksheet_linux.pdf**—an installation worksheet (only if installing under Linux)

- readme.pdf—Release Notes

NOTE: Use Acrobat Reader to open .pdf files.

If you installed a server, the media subdirectory of your Burstware installation directory contains:

- burstware1.mpg—a short sample media file

To access the documentation online, start Adobe Acrobat Reader and open the documentation in the docs subdirectory of the installation directory.

Manually Starting Burstware on Solaris and Linux

How you manually start a Burstware Server or Burstware Conductor depends on whether the server or conductor was specified as a daemon during installation. You start a server or conductor by invoking its startup script. However, if the server or conductor was specified as a daemon, the executable is invoked from a different location than that of a server or conductor not specified as a daemon.

NOTE: Conductors and servers running as daemons do not display GUIs.

Starting a Conductor or Server Not Being Run as a Daemon

Starting up a Burstware Server or Burstware Conductor that is not being run as a daemon on Solaris and Linux machines is performed through a command line interface. Configuration parameters can be optionally included with the startup command. To start up a server or conductor that is not being run as a daemon:

1. Change the working directory to your Burstware installation directory.
2. Issue this command:

```
% burstserver | burstconductor [-help] [-ini
<iniFile>] [-minmem <memSpec>] [-maxmem <memSpec>]
[-nogui]
```

For example:

```
% ./burstserver -ini burstserver1.ini -minmem 48
-maxmem 128
```

Command Line Parameters

help

The help parameter prints the usage and a brief explanation of each parameter, then exits with a non-zero status.

ini

The ini parameter directs the startup script to an initialization file other than the default.

nogui

The nogui parameter starts a Burstware Conductor or Burstware Server without displaying its GUI. For example, you may be starting a conductor or server on a Linux or Solaris machine not running X Windows. Or, conductors and servers may be running on machines without monitors.

You can remotely monitor conductors and servers—including those running with the nogui parameter—with the Burstware Console application, which is included on Burst.Com's Professional Services CD-ROM. For more information, contact your sales representative.

minmem

The minmem parameter specifies the initial amount of working memory Burstware Server or Burstware Conductor executable uses, in MB. The default values are:

- Conductor- 16 MB
- Server- 48 MB

maxmem

The maxmem parameter specifies the highest amount of working memory the Burstware Server or Burstware Conductor executable is allowed to use, in MB. The default values are:

- Conductor- 16 MB

- Server- 48 MB

About Working Memory

The amount of working memory needed is related to the number of concurrent connections that can be supported. Server and conductor memory usage tends to grow and shrink based on the number of concurrent connections the server or conductor is supporting.

There are a number of advantages to setting **maxmem** and **minmem**:

- Setting maximum memory usage can prevent excessive swapping by the virtual memory manager and thereby prevent serious performance degradation
- When serving hundreds of users, setting the **maxmem** parameter to a higher value allows the server or conductor to better handle the load
- When serving fewer users, setting the **maxmem** parameter lower than the default allows the server or conductor to conserve memory

Generally, the server needs more working memory than the conductor since the server performs more memory-intensive processing, and its connections last longer.

In addition to the default values for **minmem** and **maxmem**, the conductor and server use a certain amount of additional memory to carry out their processes. The conductor uses approximately 4MB; the server uses approximately 7MB.

Restarting a Daemon Conductor or Server

When a Burstware Server or Burstware Conductor is specified as a daemon during installation, scripts are placed in the **init.d** directory, which contains all the scripts for daemons on a host machine. To restart a daemon server or conductor, invoke the script through the runlevel directories, which contain links back to the **init.d** directory.

To restart a daemon server or conductor follow these steps:

- On Solaris:

Change to the rc3.d directory and invoke the appropriate script to restart the server or conductor (either **s99burstconductor1** or **s99burstserver1**), as in the following example:

```
% cd /etc/rc3.d
% ./S99burstconductor1 start
```

- On Linux:

  Change to either the **rc3.d** or **rc5.d** directory and invoke the appropriate script (either **s99burstconductor1** or **s99burstserver1**) to restart the server or conductor, as in the following example:

```
% cd /etc/rc.d/rc3.d
% ./S99burstconductor1 start
```

Specifying Working Memory Size for Burstware Daemons

You can change the working memory size of a Burstware Conductor or Burstware Server that is specified as a daemon using the following steps:

1. Use a text editor to open the file for either the conductor or server (either **burstconductor1** or **burstserver1**) in the following location:

   On Solaris:

   ```
   % cd /etc/init.d
   ```

   On Linux:

   ```
   % cd /etc/rc.d/init.d
   ```

2. Find the line containing the script path at the end of the script. For example:

   ```
   % /home/yip/burst2/burst_control.sh conductor \
   1 16 16 $1
   ```

3. To change the memory settings, edit the initial or maximum heap size parameters. The syntax is:

   ```
   % burst_control.sh {conductor|server} process_index\
   initial_heap_size maximum_heap_size {start|stop}
   ```

NOTE: The initial_heap_size parameter refers to minmem parameter, and maximum_heap_size parameter refers to maxmem parameter.

For example, to set the maximum heap size for the conductor to 32 MB:

```
% /home/yip/burst2/burst_control.sh conductor \ 1 \
16 32 $1
```

For more information on working memory, see the section, About Working Memory, on page 91.

CHAPTER 6

Configuring Conductors and Servers

You can customize the behavior of conductors and servers and tune their performance by setting the configuration parameters listed in their respective initialization files. While the parameter settings in the default initialization files shipped with Burstware support a fully operational conductor and server, you may choose to change the settings for your particular application.

Customizing begins during installation. Burstware edits the initialization files to the values of certain parameters—such as **serverList** and **licenseKey** for conductors, and **managedBandwidth** for servers—based on answers you provide when you install Burstware.

This chapter discusses Burstware Conductor and Burstware Server initialization files and describes conductor and server configuration parameters. Information is presented in the following sections:

- About Initialization Files
- Burstware Conductor Configuration Parameters
- Burstware Server Configuration Parameters
- Dynamic Reconfiguring

About Initialization Files

The Burstware Conductor and Burstware Server initialization files shipped with Burstware are called **burstconductor.ini** and **burstserver.ini**, respectively. At startup, a conductor looks for a file called **burstconductor.ini** and a server looks for a file called **burstserver.ini** in the current working directory.

To direct a conductor or server to use a different initialization file, specify the filename on the command line, with the **ini** parameter, as in these examples:

On Windows NT:

```
burstconductor.exe -ini burstconductor1.ini
```

```
burstserver.exe -ini burstserver1.ini
```

On Solaris and Linux:

```
% ./burstconductor -ini burstconductor1.ini
```

```
% ./burstserver -ini burstserver1.ini
```

By default, the Burstware Conductor shortcut on your desktop uses the standard **burstconductor.ini** file and the Burstware Server shortcut on your desktop uses the standard **burstserver.ini** file.

For more information on passing conductor and server command line parameters at startup and on command line parameters in general, see page 80 (Windows NT) and page 90 (Solaris and Linux).

When to Use Multiple Initialization Files

You must use multiple initialization files on one machine if you have multiple conductors or servers running on that machine. An initialization file specifies a unique TCP/IP port number and log file name for a conductor. If you are running more than one conductor or server on a single machine, you must assign each conductor its own initialization file to prevent port or log file conflicts between conductors and between servers.

Initialization File Conventions

The following conventions apply to initialization files:

- Use any text editor to edit the initialization file, but save the file as text only (ASCII)
- Comment lines and blank lines are allowed

Where to Find Initialization Files

The burstconductor.ini file can be found in the conductor subdirectory of your Burstware installation directory. For example, if you installed Burstware in the default directory, C:/Program Files/Burst/Burstware, the initialization file is C:/Program Files/Burst/Burstware/Conductor/burstconductor.ini.

The burstserver.ini file can be found in the server subdirectory of your Burstware installation directory. For example, if you installed Burstware in the default directory, C:/Program Files/Burst/Burstware, the initialization file is C:/Program Files/Burst/Burstware/Server/burstserver.ini.

Burstware Conductor Configuration Parameters

The following table summarizes Burstware Conductor configuration parameter values:

licenseKey

Specifies the license key value that Burstware Conductor uses to ensure that you are using Burstware according to the terms of your license agreement with Burst.Com. The license key value defines the:

- Number of burst-enabled players that can be concurrently connected to the Burstware Servers managed by the conductor
- Number of servers the conductor can manage
- Total amount of combined bandwidth all the managed servers can use to deliver video
- Expiration date of your Burstware license
- Name of the machine on which the conductor runs

To obtain your license key, contact your Burst.Com sales representative at sales@burst.com. Do not change this parameter value unless you contact Burst.Com for a new one.

If you want to run Burstware with less capacity than your license permits—for example, for testing purposes—you can reduce the limits defined by licenseKey with the licensedNumClients, licensedNumServers, and licensedBandwidth parameters, described later in this chapter.

serverList

Lists the Burstware Servers the conductor manages. The format is
serverList = *hostname1: [baseport1[.dataport1]]; hostname2: [baseport2[.dataport2]]...*

This string specifies the hostname or IP address of the server machine and, optionally, the values of **basePort** and **dataPort** on the machine.

If the conductor manages multiple servers, the information is included for each server, and the entries in the server list are separated by a semicolon. The order of the list determines the order in which a client will try to connect to servers.

The following examples illustrate valid serverList assignments:

serverList = server1:8080.80;server2:8080.80

The above example lists two servers and explicitly assigns **basePort** to 8080, and **dataPort** to 80 for each server.

serverList = server1:8080;server2:8080

The above example lists two servers and explicitly assigns basePort to 8080 for each. Since **dataPort** is not specified, it defaults to **basePort** plus 1, or 8081.

serverList
(continued)

serverList = server1;server2

The above example lists two servers, and does not explicitly assign either basePort or dataPort. So, basePort defaults to 8020, and dataPort defaults to basePort plus 1, or 8021.

NOTE: DataPort cannot be explicitly specified if basePort is not also explicitly specified.

A server's position on its list helps determine whether it participates in load balancing, or acts only as a standby server. See the section, Load Balancing and Failover with Available and Standby Servers, on page 38 for details.

basePort

Specifies the control port where Burstware Conductor listens for new requests.

The default port assignment for **basePort**, made during the Burstware Conductor installation process, is 8018. This value is intended to support a limited test implementation of Burstware—a single conductor and a single server on the same machine. This setting allows you to run a conductor and server on the same machine, and to avoid port conflicts if you have a web server on the same machine as Burstware.

When you set up your production Burstware environment, you may need to modify the default **basePort** value assigned by the installation process.

For most Internet firewall deployments, Burst.Com recommends setting the value of the conductor's **basePort** parameter to 8080.

This recommendation assumes that:

- Your Burstware Conductor and Burstware Server do not reside on the same machine

  Port conflicts could result if both Burstware components use the same port.
- There is no web server on the same machine as either your conductor or server

  Web servers use port 80, and often use port 8080.

Other port assignments may work in your environment, or be necessary for the configuration you prefer.

See the section, Firewalls, TCP/IP Ports, and Burstware. on page 30 for more information on port assignments and firewalls.

traceLevel

Sets the level of detail in the conductor log. Valid values are 0 through 6.

The default is **6**.

At level 0, the log contains only:

- Errors
- Initial configuration values
- Changes to configuration values

At levels 1 and 2, the log contains everything at level 0, plus:

- Sign on messages, when a player requests a connection
- Open messages, when a player has requested a multimedia file
- An exit message, when a player signs off

At level 3, the log contains everything at levels 1 and 2, plus:

- Forwarding messages, when the conductor sends the server list to a player

At levels 4 and 5, the log contains everything at level 3, plus:

- A confirmation message, when the conductor accepts a player connection

At level 6, the log contains everything at level 5 plus summary lines showing the:

- Number of configured and available servers
- Number of connected players
- Total combined available bandwidth across all servers

traceToScreen

Specifies whether the Burstware Conductor GUI's Logging screen displays log entries as they are generated. If set to **true**, the Burstware Conductor logging screen displays log entries as they are generated.

The default is **false**.

Log entries are always written to the log file, regardless of the value of **traceToScreen**.

logFileName

Specifies the file where Burstware Conductor writes log entries. You can qualify the file with either a full path or a path relative to the directory where the conductor is started. Here is an example of a full pathname:

logFileName = C:/tmp/burstconductor.log

Here is an example of a relative pathname:

logFileName = burstconductor.log

The default is **burstconductor.log** in the directory where the conductor is started.

logActionOnStart

Tells Burstware Conductor what to do with the previous conductor session's log file when you restart the conductor.

The default is **append**.

Values are:

- **append**—Burstware Conductor appends the new session's log entries to the previous session's log
- **delete**—the conductor deletes the previous log file and starts a new log file by the same name for the new session
- **save**—the conductor converts the previous session's log file to a named and numbered segment and starts a new log file for the new session

  The segment name is determined by the log file naming conventions described in the section, Log File Naming Conventions, on page 192.

logMaxFileSize

Limits the maximum size, in megabytes (MB), of the Burstware Conductor log file. When the log file grows to this size, the conductor converts it to a segment, names the segment according to Burstware Conductor log file naming conventions, and starts a new log file.

The minimum is 1. The default is 50.

logMaxSegments

Sets the maximum number of log segments Burstware Conductor can keep at once. Once the conductor creates a new segment that puts the number of log segments at this limit, the conductor deletes the oldest log segment.

The default is 1.

If you set **logMaxSegments** to 0, the conductor does not keep log segments, and overwrites the log file when it reaches **logMaxFileSize**.

consolePassword

Restricts a Burstware Console's access to a conductor or server and a Burstware Conductor's access to a server.

The default is **burstware.**

A Burstware Console's **consolePassword** value must match the **consolePassword** value for a conductor or server in order for the Burstware Console to remotely monitor the conductor or server. A Burstware Conductor's **consolePassword** value must match the **consolePassword** value for a server in order for the Burstware Conductor to manage the server.

See the section, Password-Protecting a Burstware Domain. on page 195 for details.

Other Conductor Parameters

The **burstconductor.ini** file can also be amended to contain parameters related to the limits established by the Burstware Conductor license key. By default, **burstconductor.ini** does not contain these parameters. The three optional parameters are:

- **licensedNumClients**
- **licensedNumServers**
- **licensedBandwidth**

Each optional parameter corresponds to one of the limits established by the licenseKey parameter described on page 98. Their defaults are established by licenseKey and are on the Burstware Conductor Configuration screen.

If you want to run Burstware with less capacity than your license permits—to limit the impact of video on the network or for testing purposes—you can reduce these parameter values either by modifying them on the screen or by adding them to **burstconductor.ini**. You cannot increase them beyond the limit specified by your license agreement.

NOTE: The three optional parameters must appear *after* the licenseKey parameter in the conductor's initialization file.

licensedNumClients Limits how many burst-enabled players can be concurrently connected to all the Burstware Servers managed by Burstware Conductor.

licensedNumServers Limits how many Burstware Servers the Burstware Conductor can manage. The conductor manages as many servers as this parameter permits, starting with the first server in serverList and continuing to the last. The conductor uses the remaining servers on the list as standby servers for failover.

The value of **licensedNumServers** helps determine which servers in the conductor's server list participate in load balancing, and which act only as a standby server. See the section, Load Balancing and Failover with Available and Standby Servers, on page 38 for details.

licensedBandwidth Limits how much combined bandwidth, in megabits per second (Mbps), all the Burstware Servers managed by this Burstware Conductor can use to deliver video and audio. To enforce this limit, the conductor adds the managedBandwidth values for the servers in serverList beginning with the first. If the total reaches **licensedBandwidth** and there are more servers on the list, the conductor does not route client requests to the remaining servers. Instead, the conductor uses them as standby servers for failover. See the section, Load Balancing and Failover with Available and Standby Servers, on page 38 for details.

Burstware Server Configuration Parameters

The following table summarizes Burstware Server configuration parameter values:

mediaPath

Specifies where Burstware Server looks for multimedia files requested by a burst-enabled player. The mediaPath parameter consists of one or more directories separated by semicolons. The server searches the directories in sequence for the file requested by the player.

Example:

mediaPath = D:/chapter/content;C:/burst/content

In this example, the server first searches D:/chapter/content for the requested file. If there is no match, the server searches C:/burst/content.

The default is the single directory ../media relative to the directory where the server is started.

The path separator must be a forward slash, not a backslash.

managedBandwidth

Limits the total bandwidth, in megabits per second (Mbps), this server can use to deliver video and audio. For information on how the server uses the **managedBandwidth** value, see the section, Setting Managed Bandwidth, on page 45.

The maximum permitted value is 1024.

The default is 50 megabits per second (Mbps), which is conservative for most 100 Mbps networks. You might need to adjust this value to meet your actual network capabilities. In many cases, the number might be 75 percent of the network card speed.

The value of **managedBandwidth** should never exceed the sustained transfer rate of your disk subsystem. Doing so might lead to degraded viewing experiences.

The terms of your license key agreement also influence how you set this value.

Do not set this parameter arbitrarily high to prevent rejection of player requests; doing so can diminish the quality of the viewing experience.

See the section, Setting Managed Bandwidth, on page 45 for more information on managed bandwidth.

basePort

Specifies the control port where Burstware Server listens for new requests.

The default port assignment for basePort, made during the Burstware Server installation process, is 8020. This value is intended to support a limited test implementation of Burstware—a single conductor and a single server on the same machine. This setting, along with the default dataPort setting, described below, allows you to run a conductor and server on the same machine, and to avoid port conflicts if you also have a web server on that machine.

When setting up your production Burstware environment, you may need to modify the default basePort value assigned by the installation process.

For most Internet firewall deployments Burst.Com recommends setting basePort to 8080. This recommendation assumes that:

- Your Burstware Conductor and Burstware Server do not reside on the same machine.

  Port conflicts could result if both Burstware components use the same port.
- There is no web server on the same machine as either your conductor or server.

  Web servers use port 80, and often use port 8080.

Other port assignments may work in your environment or be necessary for the configuration you prefer. See the section, Firewalls, TCP/IP Ports, and Burstware, on page 30 for more information on port assignments and firewalls.

dataPort

Specifies the data port the Burstware Server uses for data transfers. If not specified, **dataPort** defaults to **basePort** plus 1.

If this default assignment is inappropriate for your environment, you can explicitly specify a value for **dataPort**.

NOTE: If dataPort is explicitly specified, basePort must also be explicitly specified.

There is no requirement that **dataPort** be set to the address **basePort** plus 1.

When you install Burstware in your production environment, you may need to modify the default port assignments values assigned by the installation process.

For most Internet firewall deployments Burst.Com recommends setting **dataPort** for the Burstware Server to 80. This recommendation assumes that:

- Your Burstware Conductor and Burstware Server do not reside on the same machine.

  Port conflicts could result if both Burstware components use the same port.
- There is no web server on the same machine as either your conductor or server.

  Web servers use port 80, and often use port 8080.

See the section, Firewalls, TCP/IP Ports, and Burstware, on page 30 for more information on port assignments and firewalls.

traceLevel

Sets the level of detail in the server log. Valid values are 0 through 5. The default, which is 3, ensures the log contains all user-level events of interest.

At level 0, the log contains only:

- Errors
- Initial configuration values
- Changes to configuration values

At levels 1 and 2, the log contains everything in level 0 plus:

- New connections, including the name of the player machine connected
- The multimedia file the player has opened
- An exit message, when players sign off

At Level 3, the log contains everything in levels 1 and 2 plus exit statistics, such as total connection time.

At Level 4, the log contains everything in level 3, plus other information about seeks, reads, closes, pauses, and plays.

At Level 5 and above, the log is extremely verbose. Primarily useful for helping Burst.Com Technical Support diagnose problems.

NOTE: Each successive logging level places a greater load on the server, so setting this parameter arbitrarily high may degrade performance.

traceToScreen

Specifies whether the Burstware Server logging screen displays log entries as they are generated. If set to true, the Burstware Server logging screen displays log entries as they are generated.

The default is false.

Log entries are always written to the log file, regardless of the value of traceToScreen.

logFileName

Specifies the file where Burstware Server writes log entries. You can qualify the file with either a full path or a path relative to the directory where the server is started.

The default is burstserver.log in the directory where the server is started.

For more information, see the section, Log File Naming Conventions, on page 192.

logActionOnStart

Tells Burstware Server what to do with the previous server session's log file when you restart the server.

The default is append.

Values are:

- append—Burstware Server appends the new session's log entries to the previous session's log.
- delete—the server deletes the previous log file and creates a new log file by the same name for the new session.
- save—the server converts the previous session's log file to a named and numbered segment and creates a new log file for the new session. The segment name is determined by the log file naming conventions described in the section, Log File Naming Conventions, on page 192.

logMaxFileSize

Limits the maximum size, in megabytes (MB), of the Burstware Server log file. When the log file grows to this size, the server converts it to a segment, names the segment according to Burstware Server log file naming conventions, and starts a new log file.

The minimum is 1. The default is 50.

logMaxSegments

Limits the maximum number of log segments Burstware Server can keep at once. Once the server creates a new segment that puts the number of log segments at this limit, the server deletes the oldest log segment.

The default is 1.

If you set **logMaxSegments** to 0, the server does not keep log segments, and overwrites the log file when it reaches **logMaxFileSize**.

consolePassword

Restricts a Burstware Console's access to a conductor or server and a Burstware Conductor's access to a server.

The default is **burstware**.

A Burstware Console's **consolePassword** value must match the **consolePassword** value for a conductor or server in order for the Burstware Console to remotely monitor the conductor or server. A Burstware Conductor's **consolePassword** value must match the **consolePassword** value for a server in order for the Burstware Conductor to manage the server.

See the section, Password-Protecting a Burstware Domain, on page 195 for details.

Dynamic Reconfiguring

You can reconfigure some of the conductor and server configuration parameters at runtime. However, the new values do not make a persistent change in initialization files. They affect only the current session. To make persistent changes, edit the initialization file.

For more information on runtime reconfiguring, see the section, Runtime Configuration, on page 176.

CHAPTER 7

Burst-Enabling Players

The Burstware Bridge for Windows Media Player enables a Windows Media Player to play video and audio files delivered by a Burstware Server. The Burstware Bridge for Apple QuickTime Player accomplishes the same for the Apple QuickTime Player for Windows. The resulting "burst-enabled" players work together with other Burstware components to manage, deliver, and play media files, and offer all the benefits of bursting through familiar easy-to-use player interfaces.

In addition, both Burstware Bridges include:

- Burstware Bridge Settings—a configuration application that lets you set configuration parameter values on a per-machine basis. This application is described in Chapter 8, Optional Configuration of Burst-Enabled Players.
- The Burstware Bridge Monitor—a separately installed application that lets you monitor, on client machines, delivery of data to burst-enabled players. This application is described in Chapter 9, The Burstware Bridge Monitor.

Like an ordinary Windows Media Player or Apple QuickTime Player, a burst-enabled player can run standalone or embedded in a web page. Standard player features, such as Windows Media Player's player scripting, are fully supported. In fact, a burst-enabled Windows Media Player or Apple QuickTime Player works exactly as the ordinary player works, with these additions:

- All the advantages of Burstware, such as superior video quality, more efficient bandwidth usage, and unprecedented control over video delivery's effect on network performance
- A special syntax to specify a file to play from a Burstware Server
- The option for web page authors and end users to set configuration parameters that pertain to delivery of bursted data

This chapter explains how to install the Burstware Bridge for Windows Media Player and the Burstware Bridge for Apple QuickTime Player, and how to play files. It also discusses error handling, supported file formats, CODECs and the "Slow Connection Warning" dialog.

Information is presented in these sections:

- Installing the Burstware Bridges
- . Specifying Files to Be Played
- Error Handling
- Media File Formats and CODECs Supported
- Slow Connection Warning Dialog

Installing the Burstware Bridges

You can install a Burstware Bridge individually on each player machine using a self-extracting executable file or you can set up a web page to install a Burstware Bridge automatically when the web page is visited.

Installing From a Self-Extracting Executable File

The Burstware Bridge self-extracting executables are in the following locations on the Burstware installation CD-ROM:

- For the Windows Media Player bridge:
  /windows/webinstall/wmp_download/BurstWMP.exe
- For the Apple Quicktime Player bridge:
  /windows/webinstall/quicktime_download/BurstQTP.exe

To install from an executable file:

1. Locate the appropriate executable file for your installation.
2. Double-click its icon to begin the installation.
3. Click OK to complete the installation.

The next section explains how to set up a web page to install the bridge automatically.

Burstware Bridge Auto-Install Feature

You can set up a web page to install the Burstware Bridge on the end user's machine automatically, if it is not already installed. The next section provides instructions for setting up a web page to auto-install for Windows Media Player and Apple Quicktime Player under Internet Explorer. The section that follows provides the same information for Netscape Navigator.

Instructions are the same for both Windows Media Player and Apple Quicktime Player, except for the names of the files referenced (for example, BurstWMP.cab or BurstQTP.cab).

Auto-Install for Internet Explorer

In an Internet Explorer component download, the components you want the end user to download are packaged in a cabinet (.cab) file so that the browser can download them dynamically. Burstware provides two cabinet files (called **BurstWMP.cab** and **BurstQTP.cab**) on the Burstware CD, which you can use to create a web page that performs an Internet component download of the Burstware Bridge.

In the web page that performs the Internet component download, you use the OBJECT tag to reference **BurstWMP.cab** or **BurstQTP.cab**. When an end user opens the web page, the browser reads the OBJECT tag and retrieves the files necessary for the download.

To set up your web page to perform an Internet component download:

1. Copy the .cab file into the same directory containing the HTML page that performs the download. The .cab files are in the following locations on the Burstware installation CD-ROM:
   - For the Windows Media Player bridge:

     /windows/webinstall/wmp_download/BurstWMP.cab
   - For the Apple Quicktime Player bridge:

     /windows/webinstall/quicktime_download/BurstQTP.cab
2. Include the relevant HTML code sample in the HTML page performing the download. (See the code samples on page 120.)

Code sample for Windows Media Player

```
<OBJECT ID="BurstSourceFilter"
CODEBASE="BurstWMP.cab#version=2,0,0,0"
classid="CLSID:3E149130-1B20-11D3-97A8-00A0CC2274C2"
WIDTH=1
HEIGHT=1>
</OBJECT>
```

Code sample for Apple QuickTime Player

```
<OBJECT ID="BurstSourceFilter"
CODEBASE="BurstQTP.cab#version=2,0,0,0"
classid="CLSID:3E149130-1B20-11D3-97A8-00A0CC2274C2"
WIDTH=1
HEIGHT=1>
</OBJECT>
```

The CODEBASE parameter points to a location from which to download the .cab file and specifies the version number of the control contained in the .cab file.

When the end user loads a page with this code in it, Internet Explorer checks to see whether the player machine already has the Burstware Bridge installed, and if so, whether the version is greater than or equal to the version specified in the CODEBASE parameter. If the bridge is not installed, or the version of the bridge currently installed is older than the version specified in the CODEBASE parameter on the web page, Internet Explorer:

1. Prompts the user to approve the download and installation of the bridge.
2. Downloads the Burstware Bridge software from the .cab file and installs it.

NOTE: Web developers must be aware that if they update the Burstware Bridge version available on the page for download, they must also change the version number in the CODEBASE parameter in the web page code to match it.

Auto-Install for Netscape Navigator

In a Netscape Navigator install, the installation you want the end user to run is packaged in a Java archive (.jar) file so that the browser can run it automatically. Burstware provides two .jar files (BurstWMP.jar and BurstQTP.jar) on the Burstware CD that you can use to create a web page that installs the Burstware Bridge automatically.

In the web page that performs the automatic install, you reference the .jar file and include a JavaScript function that verifies that the player machine has either the same version of the Burstware Bridge referenced in the auto-install code or a more recent version. When the end user opens this web page, the JavaScript function is called and, if necessary, the latest version of the bridge is automatically installed from the .jar file.

To set up your web page to perform an automatic install:

1. Copy the .jar file into the same directory containing the HTML page that performs the download. The .jar files are located in following location on the Burstware installation CD-ROM:
   - For the Windows Media Player bridge:

     /windows/webinstall/wmp_download/BurstWMP.jar
   - For the Apple Quicktime Player bridge:

     /windows/webinstall/quicktime_download/BurstQTP.jar
2. Include the appropriate HTML code and JavaScript function (for either Windows Media Player or Apple Quicktime Player) in the HTML page performing the download. (See pages 122 and 124.)

   The Windows Media Player and Apple QuickTime Player code samples are in the following locations on the Burstware installation CD-ROM:
   - For the Windows Media Player bridge:

     /windows/webinstall/wmp_download/BurstWMPInstall.html
   - For the Apple Quicktime Player bridge:

     /windows/webinstall/quicktime_download/BurstQTPInstall.html

Code Sample for Windows Media Player
Netscape Navigator Auto-Install

```
<HTML>
<HEAD>
<TITLE>WMP-Burstware bridge install page</TITLE>
</HEAD>
<BODY onload=downloadBurstwareWMP() >

<SCRIPT LANGUAGE="JavaScript">
function downloadBurstwareWMP()
{
 // If this is not Netscape, return. This only
 // works under Netscape.
 if ( navigator.appName != "Netscape")
 {
  return;
 }
 // If the browser doesn't have Java, then return. The
 // browser must have Java in order to do the
 // automatic download.
 if ( navigator.javaEnabled() )
 {
  trigger = netscape.softupdate.Trigger;
  // If the browser must have SmartUpdate Enabled,
  // then return. Smart Update must be enabled in
  // order for the automatic download to work.
   if ( trigger.UpdateEnabled()
    {
    // If the browser isn't running on Windows, then
    // return. The browser must be on Windows in order
    // for the automatic download to work.
 if (navigator.platform != "Win32")
 {
 alert("This browser isn't running on
 Windows 95/98/NT/2000. It must be running
 on Windows 95/98/NT/2000 in order for the
 automatic download of the
 Burstware-WMP bridge to work.")
 return;
 }

// The parameter to VersionInfo() must match the
// version of the bridge.
vi = new
netscape.softupdate.VersionInfo(2, 0, 0, 0);


// The first parameter to
// ConditionalSoftwareUpdate()
// must be a full path to where the jar file can
// be found.


// The second parameter must be
// /Burst/WMPBridge.
// It cannot be changed.
trigger.ConditionalSoftwareUpdate(
"http://your_web_server/BurstWMP.jar",
"/Burst/WMPBridge",
vi, trigger.DEFAULT_MODE);
}
else
{
     alert("This browser doesn't have Smart Update
     enabled.\nIt must have Smart Update enabled in
     order for the automatic download of the
     Burstware-WMP bridge to work.");
     }
   }
     else
     {
      alert("This browser doesn't have Java
      enabled.\nJava must be enabled for the automatic
      download of the Burstware-WMP bridge to work.");
     }
}
</SCRIPT>
</BODY>
</HTML>
```

Code Sample for Apple Quicktime Player Netscape Navigator Auto-Install

```
<HTML>
<HEAD>
<TITLE>QTP-Burstware bridge install page</TITLE>
</HEAD>
<BODY onload=downloadBurstwareQTP() >

<SCRIPT LANGUAGE="JavaScript">
function downloadBurstwareQTP()
{
 // If this is not Netscape, return. This only
 // works under Netscape.
 if ( navigator.appName != "Netscape")
```

```
{
 return;
}
// If the browser doesn't have Java, then return. The
// browser must have Java in order to do the
// automatic download.
if ( navigator.javaEnabled() )
{
 trigger = netscape.softupdate.Trigger;
 // If the browser doesn't have SmartUpdate Enabled,
 // then return. Smart Update must be enabled in
 // order for the automatic download to work.
   if ( trigger.UpdateEnabled()
    {
    // If the browser isn't running on
    // Windows 95/98/NT/2000, then return.
    // The browser must be running on
    // Windows 95/98/NT/2000 in order
    // for the automatic download to work.


      if (navigator.platform != "Win32")
      {
      alert("This browser isn't running on
      Windows 95/98/NT/2000. It must be running on
      Windows 95/98/NT/2000 in order for the automatic
      download of the Burstware Bridge in order for
      Apple QuickTime Player to work.")
      return;
      }

    // The parameter to VersionInfo() must match the
    // version of the bridge.
    vi = new
    netscape.softupdate.VersionInfo(2, 0, 0, 0);
```

```
            // The first parameter to
            // ConditionalSoftwareUpdate()
            // must be a full path to where the jar file can
            // be found.


            // The second parameter must be
            // /Burst/QTPBridge.
            // It cannot be changed.

            trigger.ConditionalSoftwareUpdate(
            "http://your_web_server/BurstQTP.jar",
            "/Burst/QTPBridge",
            vi, trigger.DEFAULT_MODE);

            }

            else

            {

            alert("This browser doesn't have Smart Update
            enabled.\nIt must have Smart Update enabled in
            order for the automatic download of the
            Burstware Bridge for Apple QuickTime Player
            to work.");

            }

         }

            else

            {
             alert("This browser doesn't have Java
             enabled.\nJava must be enabled in order for
             the automatic download of the Burstware Bridge
             for Apple QuickTime Player to work.");

            }

      }
```

```
</SCRIPT>
</BODY>
</HTML>
```

When the end user loads a page with this code in it, Netscape Navigator determines whether the Burstware Bridge is already installed on the player machine and, if so, which version of the bridge is installed. If the bridge is not installed or the version of the bridge currently installed is older than the version specified in the Netscape.softupdate.VersionInfo constructor Netscape Navigator:

1. Prompts the user to approve download and installation of the bridge.
2. Downloads the Burstware Bridge software from the BurstWMP.jar file and installs it.

NOTE: Web developers must be aware that if they update the Burstware Bridge version available on the page for download, they must also change the version number in the Netscape.softupdate.VersionInfo constructor to match it.

Specifying Files to Be Played

Once the Burstware Bridge is installed, the Windows Media Player or Apple Quicktime Player becomes burst-enabled; it is can play files delivered from a Burstware Server. A burst-enabled player retrieves a multimedia file using a URL for that file. Standalone players obtain the URL from a dialog box. Embedded Windows Media Players obtain the URL from the FileName parameter in the HTML code for the web page. Embedded Apple QuickTime Players obtain the URL through a reference movie created as a pointer to the actual multimedia file.

Information Provided in the URL

To play files stored on a Burstware server, you provide several pieces of information in the URL, including:

- The protocol specifier—"burst:"— which tells the player to retrieve a file from a Burstware server.
- A list of available conductors, which the player uses to establish its connection to a Burstware server and, if the connection to the server fails at any point, to reconnect to another server.
- The media file name, which tells the server which file to access.
- The media play rate, which helps ensure a high quality viewing and listening experience. See the section, Importance of Setting Media Play Rate, on page 135.
- Optional configuration parameters.

  There are certain scenarios in which setting parameters to values other than their defaults could optimize video delivery in your Burstware deployment. See Chapter 8, Optional Configuration of Burst-Enabled Players.

The URL syntax for both standalone and embedded burst-enabled players is explained in the next section.

URL Syntax

The syntax for the URL is the same for a standalone player and for an embedded player running under Internet Explorer. The syntax differs slightly for an embedded player running under Netscape Navigator. The section that follows gives a syntax example for Internet Explorer, for Netscape Navigator, and for a standalone player.

URL Syntax for a Standalone Player

For a standalone player, specify the URL for the play request in the following format:

```
burst://conductor_list/file_name[?params]
```

For example:

```
burst://conductor1:8018;conductor2:8018/
phantom.asf?mediaplayrate=300
```

URL Syntax for Internet Explorer

For an embedded player running under Internet Explorer, specify the URL for the play request as the value of the FileName parameter of the OBJECT tag:

```
<PARAM NAME="FileName" VALUE="burst://conductor_list/
file_name[?params]>
```

The following example illustrates the use of the FileName parameter:

```
<PARAM NAME="FileName" VALUE="burst://
conductor1:8018;conductor2:8018/
phantom.asf?mediaplayrate=300">
```

URL Syntax for Netscape Navigator

For an embedded player running under Netscape Navigator, specify the Burstware conductor list and media file name in the URL of the EMBED tag:

```
Filename= "\\burst://conductor_list/
file_name[?params]& metafile=.burst"
```

The following example illustrates the use of the FileName parameter:

```
FileName = "\\burst://conductor1:8018;conductor2:8018/
phantom.asf?mediaplayrate=300&metafile=.burst"
```

NOTE: The two backslashes preceding "burst", and the "metafile=.burst" string, are required because of the Netscape Plug-in architecture.

If the URL is included in JavaScript code, you must use four backslashes preceding "burst" and the "metafile=.burst", as in the following example:

```
document.MediaPlayer.SetFileName("\\\\burst://compare/
themill.mpg?MediaPlayRate=1500&metaFile=.burst")
```

In Netscape Navigator, the Windows Media Player uses the SetFileName function to set the file name for a player. In JavaScript, use this function to set the Burstware URL for the player.

Conductor List Syntax

For all burst-enabled players, the conductor list takes this form:

```
hostname:port_number;hostname:port_number…
```

The order of the conductors on the list determines which is the primary conductor for a particular delivery session and which are the standby conductors; the first conductor in the list is the primary conductor, the rest are standby conductors. Typically you obtain the names of the conductors for the list—and their order—from your system administrator.

Parameter List Syntax

For all burst-enabled players, the parameters that follow the media file name take this form:

?[*param=value*] [&*param2=value2*] [&*param3=value3*] ...

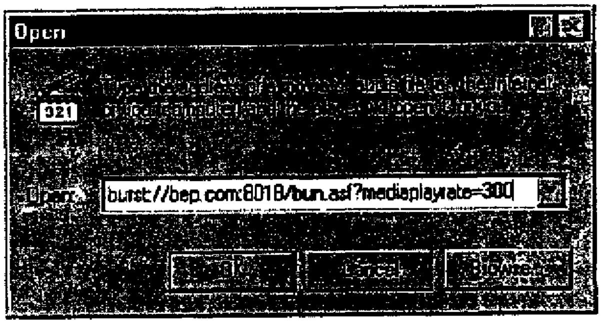
For example:

```
burst://conductor1:8018;conductor2:8018/
phantom.asf?mediaplayrate=300&backbuffersize=2000&buff
ersize=10
```

Specifying a File with a Burst-Enabled Standalone Player

To specify a media file to play with a standalone player, open a dialog box with the File > Open (Windows Media Player) or File > Open URL (Apple QuickTime Player) command and type the URL.

The following screenshot shows how to specify a URL for a standalone burst-enabled Windows Media Player:



Open: burst://bep.com:8018/bun.asf?mediaplayrate=300

Specifying a File with an Embedded Windows Media Player

In the Windows Media Player, you specify the media file to be played with the URL for that file. The URL is in the FileName parameter in the HTML code for the web page.

HTML Code Sample for Embedding a Windows Media Player in a Web Page

The HTML code sample below works for both Internet Explorer and Netscape Navigator. You can find this code in a file called BurstWMP1.html in the windows/webinstall/wmp_download directory on the Burstware installation CD-ROM.

```
<HTML>
<HEAD>
<TITLE> Burstware - Simple embedding of the Burst-
Enabled Windows Media Player</TITLE>
```

```
</HEAD>
<BODY
<P>
<OBJECT classid=CLSID:22d6f312-b0f6-11d0-94ab-
0080c74c7e95
codeBase=http://activex.microsoft.com/activex/
controls/mplayer/en/nsmp2inf.cab#Version=6,0,0,0
height=280 id=BurstPlay type=application/x-oleobject
width=360 VIEWASTEXT
standby="Loading Microsoft Windows Media Player
components...">
<PARAM NAME="AutoStart" VALUE="1">
<PARAM name="ShowStatusBar" value="1">
<PARAM NAME="FileName" VALUE="burst://
yadda.burst.com:8038/phantom.asf?MediaPlayRate=300">

<PARAM NAME="ShowControls" VALUE="1">
<EMBED type="application/x-mplayer2"
pluginspage="http://www.microsoft.com/windows95/
downloads/contents/wurecommended/s_wufeatured/
mediaplayer/default.asp"
name="BurstPlay"
width=352
Height=315
ShowControls = 1
ShowStatusBar = 1
AutoStart = 1
FileName = "\\burst://yadda.burst.com:8038/
phantom.asf?MediaPlayRate=300&metafile=.burst"

</EMBED>
</OBJECT>
</BODY>
</HTML>
```

Specifying a File with an Embedded Apple QuickTime Player

Embedding a burst-enabled Apple Quicktime Player in a web page requires the use of a reference movie in order to interpret the "burst://" portion of the URL syntax. A separate application is used to create a reference movie, which contains the URL for the multimedia file and any parameters passed. A link to the reference movie is then embedded in the web page, and points to the actual media file.

There are two ways to create a reference movie:

- Use the **BurstRefMovie.exe** included in the Burstware installation
- Use the Apple utility called **MakeRefMovie.exe**, which can be downloaded from Apple's website

Creating a Reference Movie Using BurstMakeRef.exe

BurstMakeRef.exe is a simple command line application that takes a URL as input and creates a reference movie.

To create a reference movie:

1. Locate the BurstMakeRef.exe in the windows/webinstall/ quicktime_download directory in the Burstware folder.
2. Open an MSDOS Command Prompt window.
3. Make the directory that contains **BurstRefMovie.exe** the current directory.
4. Issue the following command:

   ```
   burstmakeref <URL> <output filename>
   ```

   For example:

   ```
   burstmakeref burst://yadda.burst.com:8018/
   bunnies.mov?mediaplayrate=300 testmovbunn.mov
   ```
5. Press Enter.

   A reference movie file with the output name is created in the same directory as BurstMakeRef.exe, unless you specify a different drive or directory location in the output filename.

Creating a Reference Movie Using MakeRefMovie.exe

MakeRefMovie.exe is an Apple utility used to create reference movies. MakeRefMovie.exe has other capabilities which might be of interest. See the Apple website for more information.

To create a reference movie:

1. Double-click the MakeRefMovie.exe icon.
2. Type a name for the reference movie and save the file in the same directory as the media file it points to.
3. Pull down the Movie menu and select Add URL.
4. Delete the "http://", and type the URL of the media file, including any parameters desired.

For example:

```
burst://yadda.burst.com:8018/
bunnies.mov?mediaplayrate=300
```

5. Accept the defaults in the next screen.
6. Pull down the File menu and select Save, and close the window.

HTML Code Sample for Embedding an Apple QuickTime Player in a Web Page

Since the file's URL is contained within the reference movie, the only code that needs to be embedded in the web page is the location of the reference movie. The following code sample works for both Internet Explorer and Netscape Navigator.

```
<HTML>
<BODY>
<EMBED>

     src="burst_ref.mov"
     width="440" height="260"

</EMBED>
</BODY>
```

Optional Configuration Parameters

Burstware gives you the option of setting configuration parameters in the URL that specifies the file to play. If you do not set the parameters, the player behaves according to default parameter values Burstware assigns during installation.

A burst-enabled player is fully functional with the default values. *However, Burst.Com strongly recommends* that you set the **MediaPlayRate** parameter, for the reasons discussed in the next section. If you are using the Apple Quicktime Player, Burst.Com also recommends you set the **BackBufferSize** parameter in the URL of the specified file for reasons explained in the section, The Back Buffer, on page 159.

Chapter 8, Optional Configuration of Burst-Enabled Players, describes scenarios in which setting parameters to values other than their defaults could optimize video delivery in your Burstware deployment.

Importance of Setting Media Play Rate

The MediaPlayRate parameter specifies the rate, in kilobits per second (Kbps), at which a burst-enabled player plays a video or audio file. To help ensure a high quality viewing and listening experience, set **MediaPlayRate** to the average encoded rate—also known as "play rate"—of the video or audio file.

The default Media Play Rate value is 1500 Kbps. Although many MPEG1 files have an encoded rate of 1500 Kbps and would play properly at this rate, leaving this parameter at its default value has three possible consequences:

- If the encoded rate is less than 1500 Kbps, the Burstware Server may unnecessarily reject player connection requests, thinking the player needs more bandwidth than it actually does.
- If the actual encoded rate of your content is greater than 1500 Kbps, you may data starve your player at start-up, resulting in interruptions during play.
- If the actual encoded rate of your content is less than 1500 Kbps, there may be a substantial delay before the video or audio starts

playing, because the bridge will assume it needs to buffer more data than is actually necessary before beginning play.

How to Determine Media Play Rate

If you know the encoded bit rate of the media file, specify the media play rate to match it.

Otherwise, you can use the Burstware Bridge Monitor to help determine the proper media play rate. Follow these instructions:

1. Start the video whose encoded bit rate you want to determine.
2. Start the Burstware Bridge Monitor by clicking its shortcut in the Burstware Programs folder in your Start menu.
3. Let the video run for about ten seconds to let the buffer fill up.
4. On the "Recent" panel, observe the Data Flow area and watch the data in the "From" buffer graph. Take an average of the buffer rate over a few seconds and use that as the media play rate. See Chapter 9, The Burstware Bridge Monitor for more information.

Error Handling

If Burstware cannot deliver the requested file, the burst-enabled player displays an error pop-up window. This window reports errors occurring for a given media file request, such as the following:

- Failure to connect to a conductor or server
- Failure to open a media file
- Failure to reconnect to a server after being disconnected
- An invalid specification for the media file or conductor list

You may also want to use the Burstware Bridge Monitor to view information about network and Burstware's performance. See Chapter 9, The Burstware Bridge Monitor.

Media File Formats and CODECs Supported

Burst-enabled players can play media files stored on Burstware Servers in the following file formats and CODECs:

File Format or CODEC supported	Windows Media Player	QuickTime Player
MPEG1	*	
MP3	*	*
Quicktime - Sorenson CODEC		*
Quicktime - other	*	*
ASF	*	
AVI	*	
WMA	*	
WAV	*	

See the Burstware *Release Notes* for the current release for information about support for additional file formats and CODECs.

Slow Connection Waming Dialog

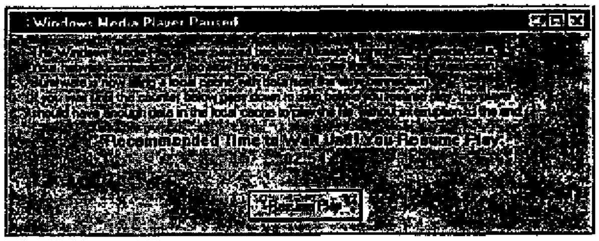
On the Internet, a user often selects a media file whose encoded bit rate is higher than the bandwidth available between the player machine and the server. In this situation, the player will experience interruptions during play, as the buffer on the player machine is refilled and emptied.

For Windows Media Player

The 2.0 release of the Burstware Bridge for Windows Media Player includes a feature that pauses play and displays a dialog when the network connection is too slow to maintain interruption-free play. The player is paused and the dialog is displayed after three "data-starves"—situations in which the Burstware buffer runs low on data.

Burstware calculates the amount of data the Burstware buffer needs to ensure that playing continues without interruption, and displays a countdown of the estimated amount of time it will take to fill the buffer with that much data. Burstware continuously monitors bandwidth, and adjusts the countdown as bandwidth conditions change. After the countdown times out, the user clicks Resume Play and play resumes. If the user clicks Resume Play before the recommended period of time has passed, the player may data-starve again.

The following screenshot illustrates the Slow Connection Warning dialog for Windows Media Player:



Windows Media Player Paused

[illegible]

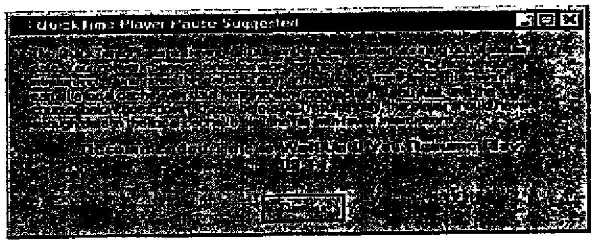
Recommended Time to Wait Until You Resume Play:

For Apple QuickTime Player

The Slow Connection Warning feature for the Apple QuickTime Player is similar to that for Windows Media Player, with these exceptions:

- The player is not paused; instead the end user must manually pause the player and then resume play once the countdown reaches zero
- The dialog is displayed after two "data starves" instead of three
- The end user clicks Dismiss instead of Resume Play

The following screenshot illustrates the Slow Connection Warning dialog for Apple QuickTime Player:



[illegible]

The WarnOnSlowConnection configuration parameter determines whether Burstware displays the dialog. By default, the parameter is set to true. To disable this feature, set WarnOnSlowConnection to false.

CHAPTER 8

Optional Configuration of Burst-Enabled Players

A burst-enabled player behaves according to Burstware-specific configuration parameters that pertain to data buffer configuration, failover, media play rate, the speed of the player's connection to the network, datawaits, and access control.

When you install the Burstware Bridge for either Apple QuickTime or Windows Media Player, Burstware assigns default values for these parameters. A burst-enabled player is fully functional with the default values. However, Burstware gives you the option of setting the configuration parameters to values other than the defaults.

This chapter summarizes each parameter and its function, and gives background information on the Burstware buffer—a key part of Burstware's delivery solution—to help you decide how to set buffer-related configuration parameters. It also describes scenarios in which setting parameters to values other than their defaults could optimize video delivery in your Burstware deployment, and explains how to set them.

Information is presented in these sections:

- Setting Bridge Configuration Parameters
- About the Bridge Configuration Parameters
- Parameter Precedence Rules
- About the Burstware Buffer
- Scenarios in Which to Set Parameters

Setting Bridge Configuration Parameters

With the exception of MediaPlayRate, which *Burst.Com strongly recommends you set* (see the section, Importance of Setting Media Play Rate, on page 135), and BackBufferSize, which Burst.Com recommends setting if you are using the Burstware Bridge for Apple QuickTime Player (see the section, The Back Buffer, on page 159), you do not need to set bridge configuration parameters.

If you do choose to set them, you can do so in the URL specifying the media file (by typing parameters in the player dialog box, or defining them in the HTML code in the web page) or in the Burstware Bridge Settings dialog on an individual player machine

NOTE: **Some parameters can only be set in the URL specifying the media file. There are also rules governing whether parameters are used from the player machine or the URL; these are discussed in the section,** Parameter Precedence Rules, on page 154.

Parameters are discussed in detail in the section, About the Bridge Configuration Parameters, on page 144. Default parameter values are described in the section, Summary of Precedences and Default Values, on page 155.

Syntax for Setting Values in the URL in an HTML File

For an embedded player, add the parameters you want to set in the URL in the HTML file, using the following syntax:

For Internet Explorer:

```
burst://<conductor_list>/<file_name>?<parameter_name>
=<parameter_value>&<parameter_name>=
<parameter_value>&…
```

For Netscape Navigator:

```
\\burst://<conductor_list>/
<file_name>?<parameter_name>
=<parameter_value>&<parameter_name>=<parameter_value>&
…metafile=.burst
```

NOTE: The two backslashes preceding "burst" and the "metafile=.burst" string are required because of the Netscape Plug-in architecture. The "metafile=.burst" string must appear at the end of the parameter list.

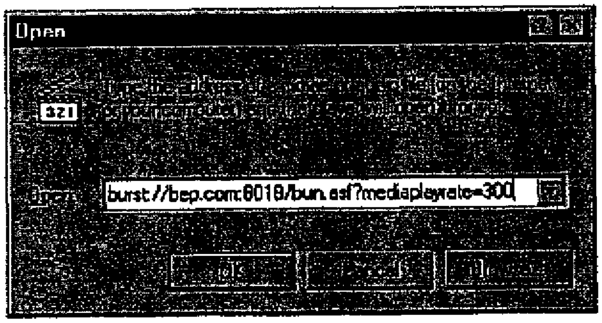
Syntax for Setting Values in the URL in a Standalone Player

For a standalone Windows Media Player, use the File > Open command to specify parameter values. For a standalone Apple QuickTime Player, use the File > Open URL command to specify parameter values.

Add the parameters you want to set in the URL, using the following syntax:

```
burst://<conductor_list>/<file_name>?<parameter_name>
=<parameter_value>&<parameter_name>=<parameter_value>&
...
```

The following screen shot illustrates setting parameters in the Windows Media Player with the File > Open command:



burst://bep.com:8018/bun.asf?mediaplayrate=300

About the Bridge Configuration Parameters

This section gives you detailed information about each of the following configuration parameters:

- BufferType Parameter (page 144)
- BackBufferSize Parameter (page 145)
- BufferDirectoryPath Parameter (page 146)
- BufferSize Parameter (page 145)
- ForbidDiskCache Parameter (page 147)
- MediaPlayRate Parameter (page 148)
- PauseWhenDataStarved Parameter (page 148)
- ReconnectInterval Parameter (page 149)
- ReconnectTimeout Parameter (page 149)
- UserNetworkBandwidth Parameter (page 150)
- WarnOnSlowConnection Parameter (page 151)
- SecondsOfDataToBuffer Parameter (page 151)
- AccessControlString Parameter (page 152)

BufferType Parameter

The BufferType parameter specifies the type of buffer to use for the Burstware buffer, which caches data on the player machine.

The values are:

- memory
- disk

The following examples set the buffer type to disk:

For Internet Explorer and standalone players:

```
burst://barney.burst.com;fred.burst.com/
phantom_300.asf?MediaPlayRate=300&BufferType=disk
```

For Netscape Navigator:

```
\\burst://barney.burst.com;fred.burst.com/
phantom_300.asf?MediaPlayRate=300&BufferType=disk&meta
file=.burst
```

See the section, Buffer Types and Sizes, on page 157 for a description of the Burstware buffers types and a discussion of the advantages and disadvantages of each type.

The default buffer type is memory.

NOTE: **If the ForbidDiskCache parameter is set to true and you specify a value of "disk" for the BufferType parameter, Burstware ignores the BufferType setting and creates a memory buffer.** See the section. ForbidDiskCache Parameter. on page 147.

BackBufferSize Parameter

The BackBufferSize parameter specifies the size, in kilobytes (KB), of the back buffer portion of the Burstware buffer. See the section, The Back Buffer, on page 159 for background information about the back buffer.

The BackBufferSize setting you specify for a given media file in the URL overrides the back buffer size set in the Burstware Bridge Settings on a player machine. This is because the format of a given media file determines how large the back buffer should be.

The minimum back buffer size is 256 KB, and the maximum back buffer size is half the total buffer size.

The default back buffer size is 512 KB.

See the next section for an example of how to set this parameter.

BufferSize Parameter

The BufferSize parameter specifies the size of the Burstware buffer, in megabytes (MB), used to cache media files on the player machine. The minimum buffer size is 2 MB. The maximum buffer size depends on the player machine's capacity.

For disk buffers, Burstware ensures a minimum of 90 MB available on the disk. For example if you request a 50 MB disk buffer and the disk has 120 MB, Burstware will create a 30 MB disk buffer, leaving 90 MB free.

For memory buffers, Burstware ensures that the buffer is not larger than available memory (the physical memory not being used by the system or other applications), unless available memory is less than 2 MB. Burstware also ensures the buffer is not larger than the amount of available virtual memory.

The examples below set buffer size to 10 MB and back buffer size to 1024 KB. (These two parameters are often set together.)

For Internet Explorer and standalone players:

```
burst://barney.burst.com;fred.burst.com/
phantom_190.mov?MediaPlayRate=190&BufferSize=10&BackBu
fferSize=1024
```

For Netscape Navigator:

```
\\burst://barney.burst.com;fred.burst.com/
phantom_190.mov?MediaPlayRate=190&BufferSize=10&BackBu
fferSize=1024&metafile=.burst
```

BufferDirectoryPath Parameter

The BufferDirectoryPath parameter specifies the directory for the disk buffer that will be created. When you use a buffer type of disk, the Burstware system creates the disk buffer in the directory specified by the BufferDirectoryPath parameter.

The default buffer directory path is the value of the client's TEMP environment variable—$TEMP.

The value of BufferDirectoryPath must be a fully qualified pathname, such as C:\Burstware\tmp. A relative pathname, such as tmp, will not work.

The examples below set the buffer directory path to C:\Burstware\tmp:

For Internet Explorer and standalone players:

```
burst://barney.burst.com;fred.burst.com/
phantom_40.asf?MediaPlayRate=40&BufferDirectoryPath=C:
\Burstware\tmp
```

For Netscape Navigator:

```
\\burst://barney.burst.com;fred.burst.com/
phantom_40.asf?MediaPlayRate=40&BufferDirectoryPath=C:
\Burstware\tmp&metafile=.burst
```

ForbidDiskCache Parameter

The ForbidDiskCache parameter specifies whether or not to allow individual player machines to specify disk buffer usage. By default, the ForbidDiskCache parameter is set to false, which means that individual players can set the buffer type to disk using the Burstware Bridge Settings dialog.

If you want to prevent individual players from requesting disk buffers, set the ForbidDiskCache parameter to true.

NOTE: This parameter can only be set in the URL specifying the media file.

The following examples set the ForbidDiskCache parameter to true:

For Internet Explorer and standalone players:

```
burst://barney.burst.com;fred.burst.com/
phantom_300.asf?MediaPlayRate=300&ForbidDiskCache=true
```

For Netscape Navigator:

```
\\burst://barney.burst.com;fred.burst.com/
phantom_300.asf?MediaPlayRate=300&ForbidDiskCache=true
&metafile=.burst
```

MediaPlayRate Parameter

The MediaPlayRate parameter specifies the rate, in kilobits per second (Kbps), at which the media file was encoded for play. Burst.Com strongly recommends providing this information when possible to aid in load balancing, prevent delays at start-up, and minimize interruptions in play.

The rate depends on how the media file is encoded. For example, many MPEG1 files have a play rate of 1500.

The default value is 1500.

NOTE: **This parameter can only be set in the URL specifying the media file.** See the section, Importance of Setting Media Play Rate, on page 135.

The following examples set the media play rate to 300 Kbps:

For Internet Explorer and standalone players:

```
burst://barney.burst.com;fred.burst.com/
phantom_300.asf?MediaPlayRate=300
```

For Netscape Navigator:

```
\\burst://barney.burst.com;fred.burst.com/
phantom_300.asf?MediaPlayRate=300&metafile=.burst
```

PauseWhenDataStarved Parameter

The PauseWhenDataStarved parameter specifies how the burst-enabled player handles playing if the Burstware buffer runs low on data.

When PauseWhenDataStarved is set to true, the burst-enabled player pauses playing when the Burstware buffer runs low on media data, and resumes playing when the buffer regains sufficient data. When the PauseWhenDataStarved parameter is set to false, the burst-enabled player behaves as it normally does when media data runs low (that is, it starts to skip frames and eventually comes to a stop).

The default value for the PauseWhenDataStarved parameter is true.

The following examples set the PauseWhenDataStarved parameter to false:

For Internet Explorer and standalone players:

```
burst://barney.burst.com;fred.burst.com/
phantom_300.asf?MediaPlayRate=300&PauseWhenDataStarved
=false
```

For Netscape Navigator:

```
\\burst://barney.burst.com;fred.burst.com/
phantom_300.asf?MediaPlayRate=300&PauseWhenDataStarved
=false&metafile=.burst
```

ReconnectInterval Parameter

The ReconnectInterval parameter is used when a player is unexpectedly disconnected from a server. The ReconnectInterval parameter specifies the period of time—in milliseconds—during which the Burstware system sleeps between attempts to reconnect to a server. During a reconnection attempt, the player requests an ordered list of available servers from the conductor and then attempts to connect to each of the servers on the list. If the player fails to connect to any of the servers on the list, the player rests for the period of time specified by ReconnectInterval before getting a new server list from the conductor.

See the section, Server Failover, on page 26 for background information on failover and the reconnect interval.

The default value is 500 (half a second).

See the next section for examples that set this parameter.

ReconnectTimeout Parameter

The ReconnectTimeout parameter is used when a player is unexpectedly disconnected from a server. It specifies the period of time, in seconds, during which the player attempts to reconnect to a server. If the player is unable to reconnect to a server before the time specified by the ReconnectTimeout parameter has elapsed, the burst-enabled player presents an error dialog to the user. This error indicates that network problems or server failures are disrupting the media data delivery.

See the section, Server Failover, on page 26 for background information on failover and the reconnect timeout.

The default value is 20.

The examples below set the ReconnectInterval parameter to 1000 milliseconds, and the ReconnectTimeout parameter to 30 seconds. (These parameters are often set together.)

For Internet Explorer and standalone players:

```
burst://barney.burst.com;fred.burst.com/
phantom_300.asf?MediaPlayRate=300&ReconnectTimeout=30&
ReconnectInterval=1000
```

For Netscape Navigator:

```
\\burst://barney.burst.com;fred.burst.com/
phantom_300.asf?MediaPlayRate=300&ReconnectTimeout=30&
ReconnectInterval=1000&metafile=.burst
```

UserNetworkBandwidth Parameter

The UserNetworkBandwidth parameter specifies the maximum bandwidth, in kilobits per second (Kbps), that Burstware can use to deliver data to a client.The scenario described under the section, "You want to restrict bandwidth on your video/audio-enabled site." on page 163, describes a situation in which you would set this parameter.

The default user network bandwidth is 100,000. (100 Mbps.)

The following examples set the UserNetworkBandwidth parameter to 350 Kbps.

For Internet Explorer and standalone players:

```
burst://barney.burst.com;fred.burst.com/
phantom_300.asf?MediaPlayRate=300&UserNetworkBandwidth
=350
```

For Netscape Navigator:

```
\\burst://barney.burst.com;fred.burst.com/
phantom_300.asf?MediaPlayRate=300&UserNetworkBandwidth
=350&metafile=.burst
```

WarnOnSlowConnection Parameter

The WarnOnSlowConnection parameter specifies whether, after three "data starves"—situations in which the Burstware buffer runs low on data because of a slow network connection—playing pauses and a dialog informs users the buffer is running low on data.

NOTE: The dialog is displayed on both QuickTime Players and Windows Media Players. Playing pauses only on Windows Media Players.

For a detailed explanation of the dialog, see the section, Slow Connection Warning Dialog, on page 138.

The default is true.

NOTE: In order for WarnOnSlowConnection to work, the PauseWhenDataStarved parameter must be set to true.

The following examples set the WarnOnSlowConnection parameter to false:

For Internet Explorer and standalone players:

```
burst://barney.burst.com;fred.burst.com/
phantom_40.asf?MediaPlayRate=40&WarnOnSlowConnection=
false
```

For Netscape Navigator:

```
\\burst://barney.burst.com;fred.burst.com/
phantom_40.asf?MediaPlayRate=40&WarnOnSlowConnection=
false&metafile=.burst
```

SecondsOfDataToBuffer Parameter

The SecondsOfDataToBuffer parameter specifies the number of seconds worth of data that must accumulate in the buffer before video begins playing, either at start-up or after a seek.

When a burst-enabled player starts up, it reads a large chunk of data from the Burstware buffer. By "cushioning" the buffer with sufficient data before play begins, you can prevent buffer depletion and the resulting interruptions in play. SecondsOfDataToBuffer specifies the amount of data in the cushion.

The minimum value is 0. The maximum value is the length of the video or audio file.

The default is 5 seconds.

The following examples set the SecondsOfDataToBuffer parameter to 10 seconds:

For Internet Explorer and standalone players:

```
burst://barney.burst.com;fred.burst.com/
phantom_190.mov?MediaPlayRate=190&SecondsOfDataToBuffe
r=10
```

For Netscape Navigator:

```
\\burst://barney.burst.com;fred.burst.com/
phantom_190.mov?MediaPlayRate=190&SecondsOfDataToBuffe
r=10&metafile=.burst
```

AccessControlString Parameter

The AccessControlString parameter can be used to allow or deny users' access to content. The access control string is passed unchanged to Burstware Conductor's access control module, and the conductor's access control module uses the value of the parameter, along with other parameters, to decide whether to fulfill or deny a play request. See Chapter 11, Burstware Security, for more information on Burstware's access control feature.

A Burstware application can make use of an access control string in a number of ways:

- As an encrypted representation of a user name and password

For example, when the end user requests a multimedia file, the application might query the end user for a user name and password, and encrypt this information prior to requesting the media file from Burstware. The encrypted name and password are then passed into Burstware as the AccessControlString parameter, so the conductor's access control module can perform user authentication and determine if the request should be fulfilled.

- To store any information useful in logging, since the access control string is written verbatim to the conductor log

The AccessControlString parameter is optionally specified in the play request URL. The string can be of any length and can contain any character allowed in base64 encoding (A-Z, a-z, 0-9,+,/,=).

The following examples set the access control string to an encrypted authentication key. Note that typically this string would be dynamically generated by the web page.

For Internet Explorer and standalone players:

```
burst://barney.burst.com;fred.burst.com/
phantom_190.mov?MediaPlayRate=190&AccessControlString=
cA9Fv2s/63+Q
```

For Netscape Navigator:

```
\\burst://barney.burst.com;fred.burst.com/
phantom_190.mov?MediaPlayRate=190&AccessControlString=
cA9Fv2s/63+Q&metafile=.burst
```

Parameter Precedence Rules

Parameter precedence rules determine which parameters the Burstware Bridge uses when playing a media file. These are the parameter precedence rules:

1. Parameters set in the Burstware Bridge Settings dialog on a player machine override parameters passed in the URL specifying a media file, except for the ForbidDiskCache and BackBufferSize parameters.
2. Parameters passed in the URL specifying a media file override default values for the Burstware Bridge.
3. If parameters are not explicitly set in the Burstware Bridge Settings dialog or in the URL, the default values for the Burstware Bridge are used.

Parameter Precedence Exceptions

As a general rule, when parameters can be set on either the player machine or in the URL specifying the media file, settings on the player machine take precedence. However, there are two exceptions to this rule: the ForbidDiskCache parameter and the BackBufferSize parameter.

BackBufferSize Parameter Exception

The BackBufferSize setting you specify for a given media file in the URL overrides the back buffer size set for an individual player machine. This is because the format of a given media file determines how large the back buffer should be.

ForbidDiskCache Parameter Exception

If ForbidDiskCache is set to true and you specify a value of disk for the BufferType parameter, Burstware ignores the BufferType setting and creates a memory buffer.

Summary of Precedences and Default Values

Default values for the configuration parameters are summarized in the following table. This table also summarizes where each parameter can be specified, and whether a URL or machine setting takes precedence.

Parameter	Default Value	Where Specified	Precedence
BufferType	Memory	URL or machine	machine[a]
BackBufferSize	512 KB	URL or machine	URL
BufferDirectoryPath	Value of $TEMP environmental variable on client machine	URL or machine	machine
BufferSize	Depending on the system, up to half of free memory	URL or machine	machine
ForbidDiskCache	false	URL only	URL
MediaPlayRate	1500 Kbps	URL only	URL
PauseWhenDataStarved	true	URL or machine	machine
ReconnectInterval	500 milliseconds	URL or machine	machine
ReconnectTimeout	20 seconds	URL or machine	machine
UserNetworkBandwidth	100 Mbps	URL or machine	machine
WarnOnSlowConnection	true	URL or machine	machine
SecondsOfDataToBuffer	5 seconds	URL or machine	machine
AccessControlString	none	URL only	URL

[a] If ForbidDiskCache is set to true and you specify a value of disk for the BufferType parameter on a player machine, Burstware ignores the BufferType setting and creates a memory buffer.

About the Burstware Buffer

The Burstware buffer is a memory or disk buffer on the player machine that the Burstware system uses to cache media data. Bursting ensures that the Burstware buffer always contains data and the burst-enabled player has immediate access to the data it needs. Caching the data locally provides for uninterrupted viewing and frees up network resources, resulting in their more efficient use.

When you configure a burst-enabled player, you can specify the following buffer settings:

- Buffer type
- Buffer directory path
- Buffer size
- Back buffer size
- Forbid disk caching
- Seconds of data to buffer

The following sections provide information on these buffer settings.

Buffer Types and Sizes

Burstware allows you to choose both the type and size of the Burstware buffer used by the burst-enabled player. The type you choose depends on the needs of your particular environment. The size of the buffer is calculated automatically by the burst-enabled player depending on the size of the requested media file and on how much disk space/memory is available, but you can also set the buffer size manually.

There are two buffer types available:

- Memory
- Disk

Whether a memory buffer (the default) or disk buffer is used, a burst-enabled player estimates available buffer space in the same way. When a burst-enabled player requests a media file, it checks the disk space (for disk buffer) or memory available (for memory buffer) to verify that there is sufficient space to cache the file. A burst-enabled player automatically determines the buffer size needed, based on the size of the file requested and the resources available on the player machine. The buffer size could potentially be as large as the file itself.

Memory Buffer

A memory buffer is a good choice when you have memory available and do not need a very large buffer. In general, for Internet deployments—in which you have little control over client machines and client machines might have slow disks—Burst.Com strongly recommends using a memory buffer. This is especially true for playing files encoded at a high bit rate (2 Mbps or higher).

To provide the full advantages of Burstware, the player machine should be able to allocate at least 4MB for the memory buffer.

Disk Buffer

Newer machines typically have large amounts of disk space, which can support a large buffer size—10 MB to 1 GB or more—when a large memory buffer cannot be supported.

However, a disk buffer is not as fast as a memory buffer and can affect system performance. On older machines, or when viewing files encoded at a high bit rate (2 Mbps, for example), disk overhead might slow the machine down enough to affect the viewing experience.

The file used for the disk buffer is automatically deleted after the burst-enabled player exits. On Windows NT, even if the burst-enabled player exits abnormally or is killed, the file is immediately and automatically deleted.

On Windows 95 or 98, if the burst-enabled player is killed or exits abnormally, the file is deleted the next time the machine is rebooted or the burst-enabled player comes up and invokes the Burstware Bridge.

When you choose a disk type of buffer, you have the option of specifying the directory path used for creating the buffer. See BufferDirectoryPath Parameter, on page 146.

Disk caching can particularly present a problem with high-bitrate files because the disk must keep pace both with writing the file as it is bursted into the buffer and, simultaneously, reading it as it is played. This can be too much for a slow disk to handle.

NOTE: Many laptop machines have slow disks.

Under certain circumstances, you may want to prevent players from using disk buffers. For example, you may want to ensure that end users cannot copy media data from a disk cache. By setting the **ForbidDiskCache** parameter to true in the URL specifying the media file on your web page, you can prevent the end user from specifying a buffer type of disk for his or her individual player machine. See ForbidDiskCache Parameter, on page 147.

You can also use the Burstware access control feature to ensure that content is not cached to an end user's disk. See the section, Burstware Access Control, on page 197 for more information.

The Back Buffer

The back buffer is a section of the Burstware buffer that retains media data skipped over or already read. The Burstware system retains the media data in the back buffer until the section of the buffer being used for new data is completely filled. Then, with the next data delivery, the Burstware system moves the oldest data out of the back buffer to make room for newer data.

A larger (at least 1 MB) back buffer is useful when you are using an AVI or QuickTime file format that separates the audio and video portions of the media file. These formats often require the burst-enabled player first read a section of the video data, then seek backwards to read the corresponding audio data. If the back buffer is large enough, the burst-enabled player can read the audio data locally in the back buffer, without waiting for it to be delivered from the Burstware Server.

Back Buffer Size

By default, the back buffer is 512 KB. You can reset the size of the back buffer using Burstware's **BackBufferSize** parameter. See the section. BackBufferSize Parameter. on page 145.

For QuickTime and AVI file formats, you should use a back buffer size of at least 1048 KB. For all other file formats, you can use 512 KB.

Setting Parameter Values on an Individual Machine

The Burstware Bridge Settings application lets you set Burstware Bridge configuration parameters for an individual player machine.

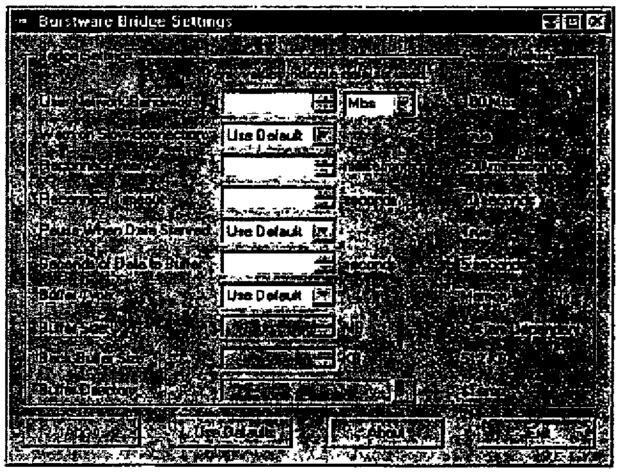
NOTE: When you set Burstware Bridge parameters on an individual machine, their values override those of any parameters passed from a URL specifying a media file except MediaPlayRate, BackBuffer, and ForbidDiskCache.

Find the Burstware Bridge Settings via the Start menu, under Programs/ Burstware/Burstware Bridge Settings.

The Burstware Bridge Settings window displays the following parameters with their default values:

- UserNetworkBandwidth
- WarnOnSlowConnection
- ReconnectInterval
- ReconnectTimeout
- PauseWhenDataStarved
- SecondsofDatatoBuffer
- BufferType
- BufferSize
- BackBufferSize
- BufferDirectoryPath

The following screenshot illustrates the Burstware Bridge Settings window:



Scenarios in Which to Set Parameters

There are certain scenarios in which changing the value of configuration parameters could optimize video delivery in your deployment. Several of these parameters can be set in the URL specifying the media file in the HTML code of a web page, seamlessly passing the parameter to player machines that request the file.

NOTE: **Parameters set in the URL generally do not affect machines in which the parameters have been changed in the Burstware Bridge Settings dialog.** See the section, Parameter Precedence Rules, on page 154 **for more information.**

The next sections describe such scenarios.

You want to prevent content from being copied illegally onto a user's machine

By default, Burstware uses a disk buffer on a burst-enabled player client machine. To prevent users from copying cached media files onto their hard drive, set the **ForbidDiskCache** parameter to **true** in the URL specifying the media file in the HTML code on your web page. This prevents users from changing the buffer type to disk in the Burstware Bridge Settings dialog on their machines.

You can also use the Burstware access control feature to ensure that content is not cached to an end user's disk. See the section, Burstware Access Control, on page 197 for more information.

You expect bandwidth fluctuations in your deployment

An unstable network connection may cause bandwidth fluctuations, or cause the speed of the network connection to drop below the media play rate, thereby compromising video quality. To stabilize video delivery, set the **SecondsOfDataToBuffer** parameter to a higher number of seconds (such as 10) to allow more data to accumulate in the player machine's buffer before video play begins. Set this parameter in the URL specifying the media file in the HTML code on your web page.

At start-up time and after a seek, a burst-enabled player reads a large chunk of data from the Burstware buffer. By "cushioning" the buffer with sufficient data before play begins, you can prevent buffer depletion and the resulting interruptions in play that can occur when the player reads the first big chunk. SecondsOfDataToBuffer specifies the amount of data in the cushion.

You do not anticipate significant network bandwidth fluctuations and want to minimize delays at start-up time and after seeks

If you do not expect much bandwidth fluctuation, set the SecondsOfDataToBuffer parameter to a lower number of seconds (such as 3) to minimize delays at start-up time and after seeks. Set this parameter in the URL specifying the media file in the HTML code on your web page.

At start-up time and after a seek, a burst-enabled player reads a large chunk of data from the Burstware buffer. By "cushioning" the buffer with sufficient data before play begins, you can prevent buffer depletion and the resulting interruption in play that can occur when the player reads the first big chunk. SecondsOfDataToBuffer specifies the amount of data in the cushion.

Video quality is poor or play is pausing on machines with sufficient network capacity to handle the media's encoded rate

If a machine with sufficient network capacity is experiencing poor video quality, the machine may be unable to provide the proper transfer rate from its disk. Try setting the **BufferType** parameter to **memory** in the Burstware Bridge Settings dialog on the player machine.

In general, for Internet deployments—in which you have little control over client machines and client machines might have slow disks—Burst.Com strongly recommends using a memory buffer. This is especially true for playing files encoded at a high bit rate (2 Mbps or higher). The web page author should also set the ForbidDiskCache parameter to true to prevent individual users from overriding the BufferType setting and using a disk cache.

You want to restrict bandwidth on your video/audio-enabled site

You may want to restrict bandwidth on your video- or audio-enabled site, for a number of reasons, such as:

- Your ISP or hosting service charges you by bytes delivered
- You want to leave a portion of the user's bandwidth free for other activities while delivering video or audio

In these cases and others, you can restrict the bandwidth used by Burstware to deliver data to an individual client by setting the UserNetworkBandwidth parameter in the play request URL in the HTML code on your web page.

CHAPTER 6

The Burstware Bridge Monitor

The Burstware Bridge Monitor is a client application that monitors a Burstware Bridge on an individual player machine. The Burstware Bridge Monitor displays information about media file delivery, such as how much data the server has delivered to the player and how much of that data has been played, as well as information about network conditions.

In addition, you can use the Burstware Bridge Monitor to view statistics on multiple media files currently being played by burst-enabled players.

This chapter describes how to install the Burstware Bridge Monitor, and how to use it to obtain network and media file play information.

Installing the Burstware Bridge Monitor

You install a Burstware Bridge Monitor individually on each player machine using a self-extracting executable file, called **BurstMon.exe**, which your system administrator can make available on the network.

To install from BurstMon.exe:

1. Double-click the BurstMon.exe icon.
2. Click OK to complete the installation.

Using the Burstware Bridge Monitor

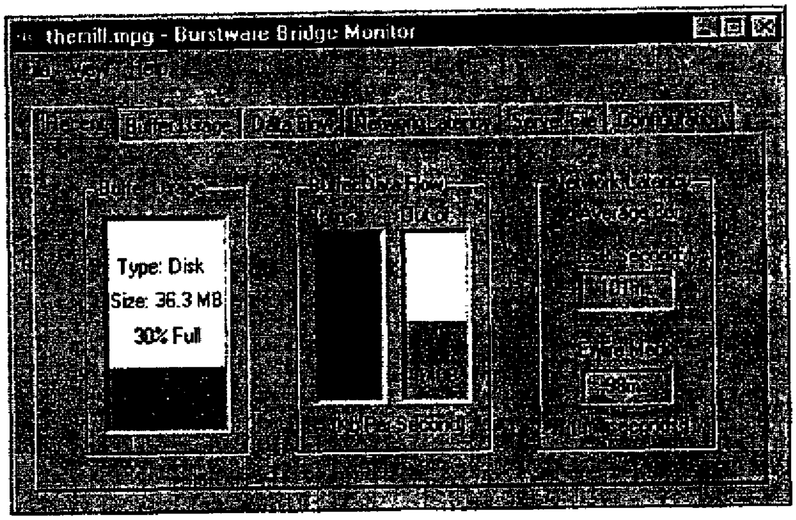
The installation program installs the Burstware Bridge Monitor in the Burstware folder. To access the Burstware Bridge Monitor, click the Burstware Bridge Monitor shortcut in the Programs/Burstware folder in your Start menu.

The Burstware Bridge Monitor window opens a window with the following tabs, each of which displays a panel:

- Recent
- Buffer Usage
- Data Flow
- Network Latency
- Server-File
- Configuration

The following sections provide detailed information about each of the Burstware Bridge Monitor panels.

Recent



Type: Disk
Size: 36.3 MB
30% Full

The Recent panel displays information about the Burstware buffer and the data flow into it during the last two seconds of play. Specifically, it displays:

- Size of the buffer
- Percentage of the buffer that is full
- Rate at which the data is flowing to the buffer from the server, and from the buffer to the burst-enabled player, in Kbps.

- Average network latency, that is, the time it took the individual bursts of data to get from the server to the client

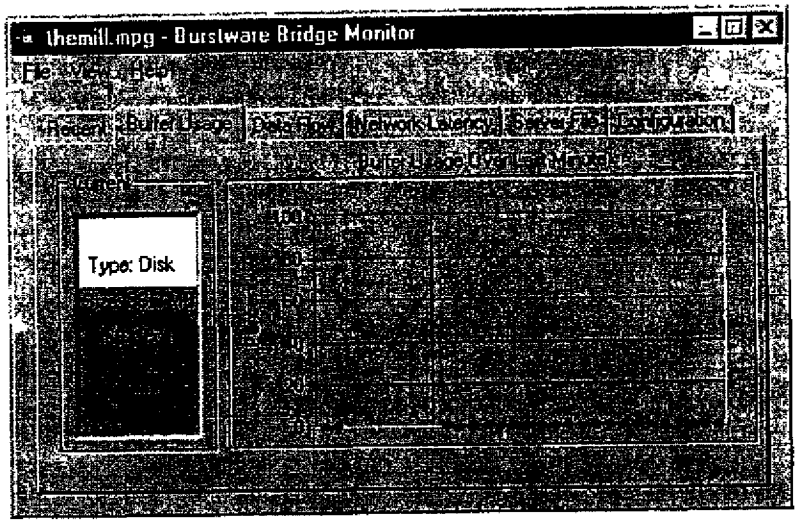
This information lets you watch Burstware in action. You can check the average network latency for information about network conditions. If the network latency goes up, the network is either busy or getting busier, and there may be delays in video delivery.

Determining Media Play Rate from Data Flow Graph

The Data Flow area in this panel can also be used to determine the encoded bit rate of the video being played. By watching the amount of data in the "Out of" graph for ten to twenty seconds while a video is playing, you can ascertain the average encoded bit rate (also known as the "media play rate") of the file. For certain types of files this can vary over time, but ten to twenty seconds is usually a large enough sample to obtain an accurate average play rate.

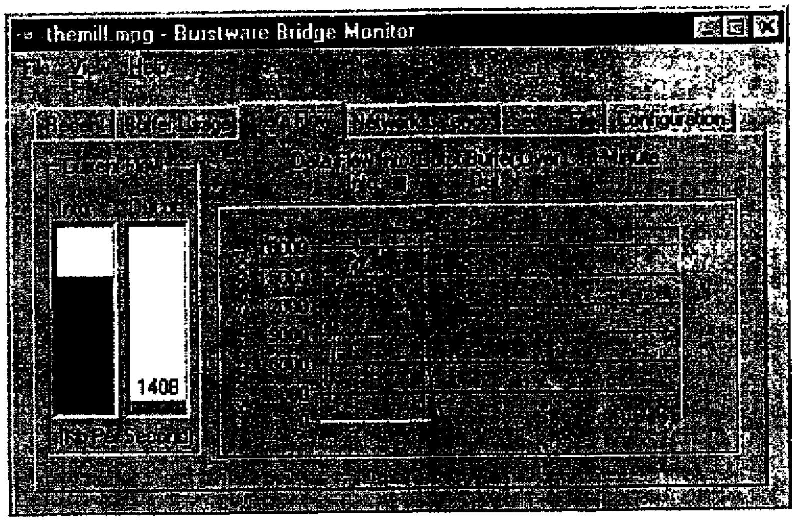
Determining the media play rate of a file is important, because when you play a file from a Burstware Server, you must give the server the media play rate to help ensure a high quality viewing experience. This is done by setting the **MediaPlayRate** parameter in the URL used to play the file. See Chapter 8, Optional Configuration of Burst-Enabled Players, for more information on the **MediaPlayRate** parameter.

Buffer Usage



The Buffer Usage panel shows how full the Burstware buffer is as a percentage of its total capacity. The Current field displays the current usage. The Buffer Usage Over Last Minute graph displays the usage over the last minute, broken down into two-second increments.

Data Flow

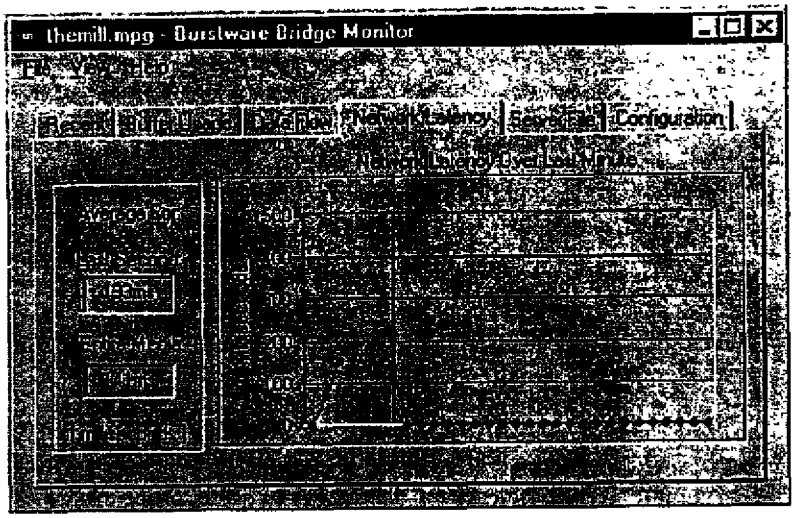


The Data Flow panel reports the rate at which data is being delivered from the server into the Burstware buffer and the rate at which the burst-enabled player is reading data from the Burstware buffer, in Kbps. The Current Flow field displays the current flow rate of data into and out of the buffer in the last two seconds.

The graph titled "Data Flow Into/From Buffer Over Last Minute" displays the flow rate into and out of the buffer over the last minute, broken down into two-second increments.

The Current Flow area in this panel can also be used to determine an appropriate media play rate for a media file. By watching the amount of data in the "Out of" graph for ten to twenty seconds while a video is playing, you can determine an average media play rate.

Network Latency

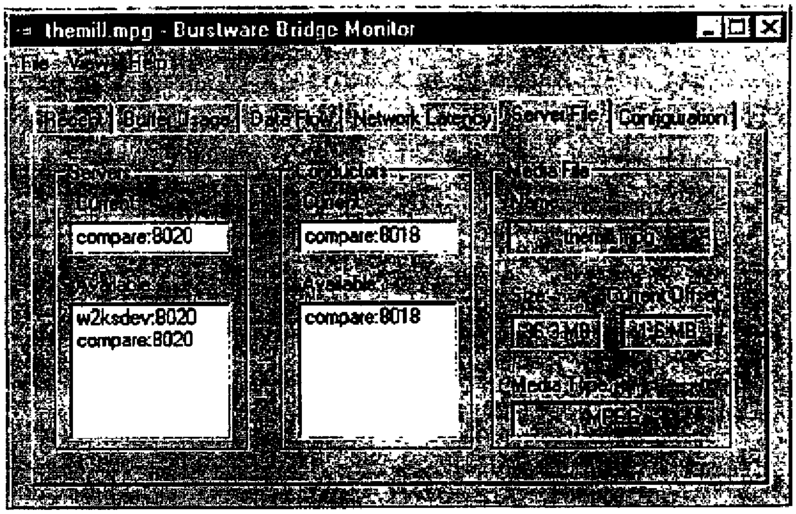


The Network Latency panel displays the time—in milliseconds—for the individual bursts of data to get from the server to the client over the last minute.

The Average for the Last Second field displays the average latency over the last second. The Entire Video field displays the average network latency for the entire media file. The graph entitled "Network Latency Over Last Minute" displays the network latency over the last minute, broken down into two-second increments.

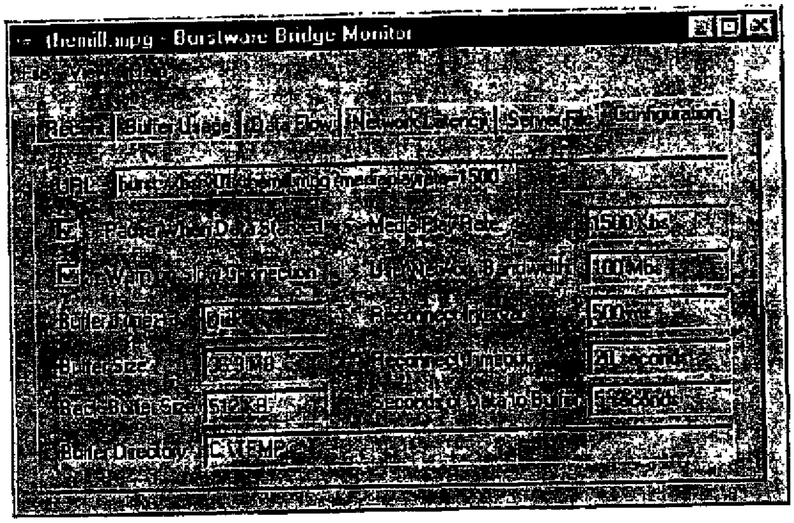
You can check network latency values for information about network conditions. If the network latency goes up, the network is either busy or getting busier, or a server has become unavailable, and there may be delays in video delivery.

Server-File



The Server-File panel shows information about available servers and conductors, the server to which the client is currently connected, the conductor that established the client-server connection, and the media file being delivered to the client. The Media File section displays the name of the media file, its size, and the position in the media file at which the burst-enabled player is currently reading.

Configuration



The Configuration panel displays parameter settings used by the Burstware Bridge for the media file(s) currently playing. Specifically, it displays:

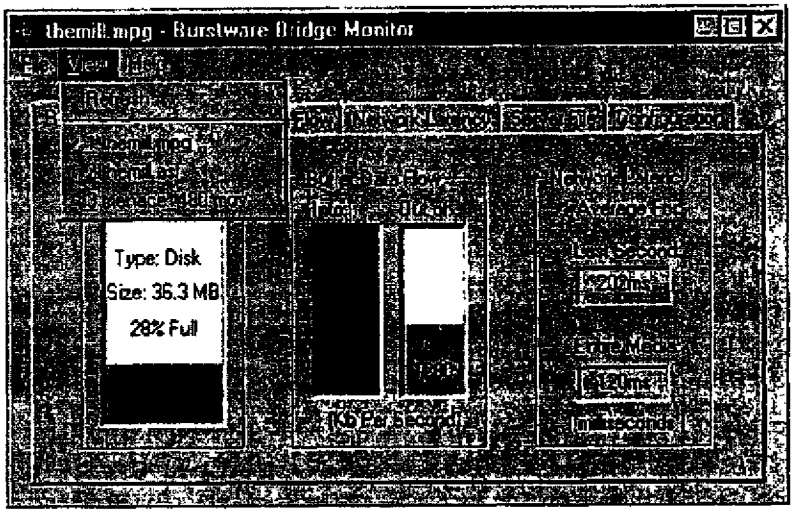
- The URL of the file(s) playing
- Whether the PauseWhenDataStarved parameter is turned on
- Whether the WarnOnLowBandwidth parameter is turned on
- The value of the BufferType parameter
- The value of the BufferSize parameter
- The value of the BackBufferSize parameter
- The value of the BufferDirectoryPath parameter
- The value of the MediaPlayRate parameter
- The value of the UserNetworkBandwidth parameter
- The value of the ReconnectInterval parameter
- The value of the ReconnectTimeout parameter
- The value of the SecondsOfDataToBuffer parameter

The Configuration panel settings are read-only. Parameters can be changed in the Burstware Bridge Settings dialog on the client machine or in the URL specifying the media file. For information on changing parameters, see Chapter 8, Optional Configuration of Burst-Enabled Players.

Monitoring Multiple Media Files

If multiple burst-enabled players are running simultaneously on a particular client machine, you can view statistics on multiple media files currently being played. To view statistics for multiple files:

1. Pull down the View menu and select the file for which you would like to view statistics, as illustrated below. A new Burstware Bridge Monitor window opens for the file.
2. Repeat for each file to view. A new window appears for each file selected.



CHAPTER 10

Conductor and Server Runtime Management

Once your Burstware deployment is up and running, you can monitor the status of its individual components and of the overall deployment through the conductor and server graphical user interfaces (GUIs). These windows report configuration, performance, and logging information. Unless you specify otherwise, the GUIs appear automatically on your desktop when you start a conductor or server.

NOTE: **Conductors and servers running as daemons do not display GUIs.** See the section, Manually Starting Burstware on Solaris and Linux, on page 89.

In certain situations, you may want to start a component without displaying its GUI. The section Command Line Parameters, on page 80 explains how to start a conductor or server without displaying its GUI.

By regularly monitoring the status of your Burstware deployment, you can anticipate and prevent problems and tune your deployment to minimize costs and maximize service to end users.

You can also remotely monitor conductors and servers with the Burstware Console application, allowing your entire deployment to be monitored from a single Network Operating Center (NOC). The Burstware Console is included as part of Burst.Com's Professional Services CD-ROM. For more information on the Burstware Console, contact your sales representative.

The GUIs and the Burstware Console report a variety of information, such as:

- How long a server or conductor has been running
- How many bytes a server has delivered over a given time period
- How much excess bandwidth capacity a server has
- How a server or conductor is configured

In addition to the information available in the GUIs, Burstware Conductors and Burstware Servers produce detailed log files that contain useful information about media file requests, the amount of data delivered to a particular client and across all clients, detailed client profiles, server loads, errors, and more.

This chapter explains how to monitor and interpret runtime information and how to manage log files.

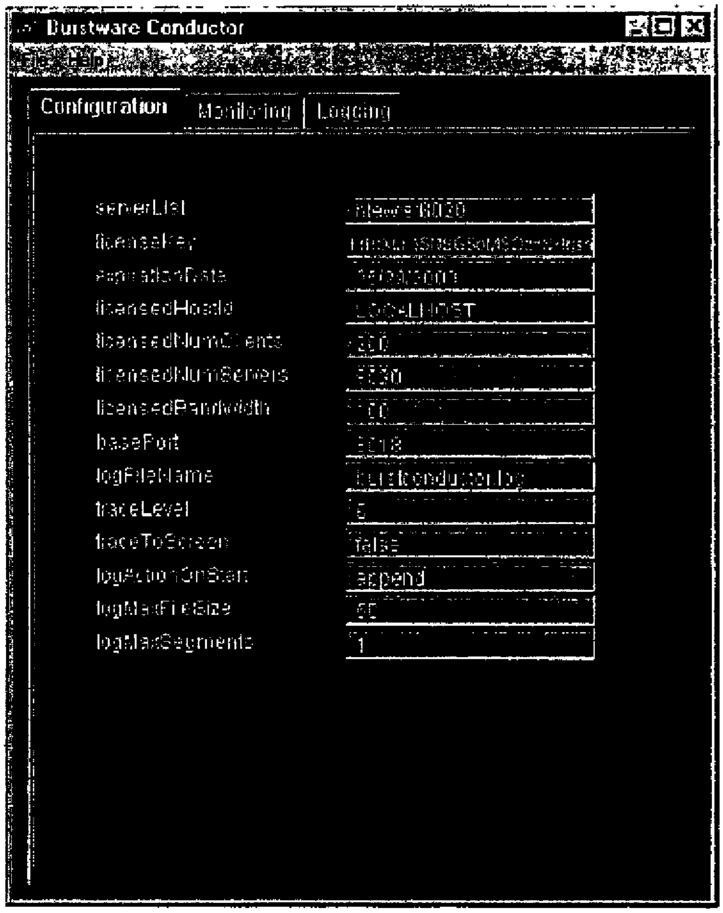
Information is presented in these sections:

- Runtime Configuration
- Monitoring
- Logging
- Log File Management

Runtime Configuration

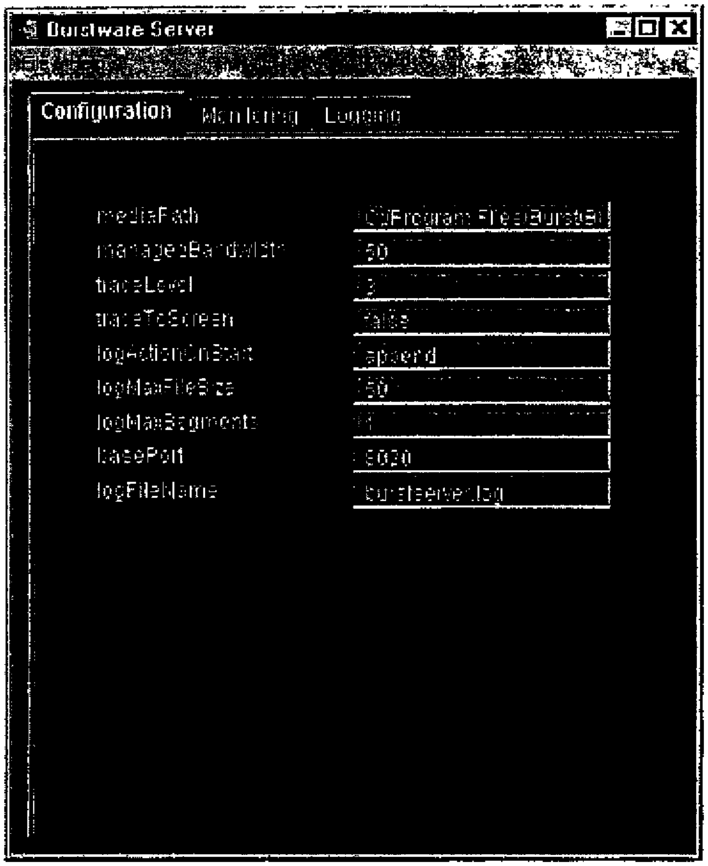
Once you have started the Burstware Conductor or Burstware Server, you will see a GUI like the one in Figure 7. Click the Configuration tab to display the Burstware Conductor or Burstware Server configuration screen.

These screens, illustrated in Figure 7 and Figure 8, respectively, enable you to view and change the current configuration parameter values for the conductor and server.



serverList [illegible]:8020
licenseKey [illegible]
expirationDate [illegible]
licensedHostId LOCALHOST
licensedNumClients 200
licensedNumServers [illegible]
licensedBandwidth 100
basePort 8018
logFileName burstconductor.log
traceLevel 5
traceToScreen false
logActionOnStart append
logMaxFileSize 50
logMaxSegments 1

Figure 7: Conductor configuration screen



mediaPath	C:\Program Files\BurstBi
managedBandwidth	50
traceLevel	3
traceToScreen	false
logActionOnStart	append
logMaxFileSize	50
logMaxSegments	4
basePort	6020
logFileName	burstserver.log

Figure 8: Server configuration screen

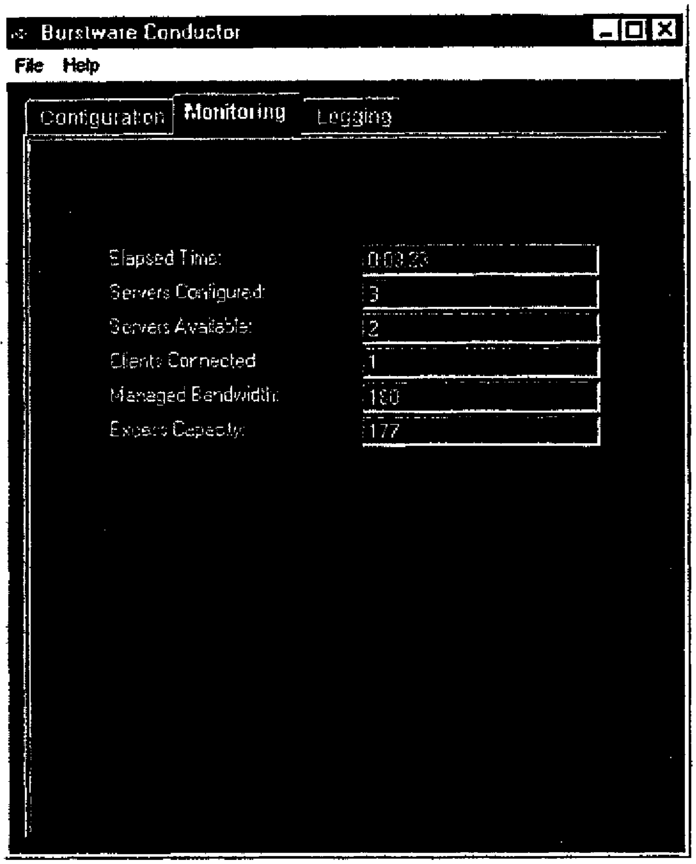
You can change values in the highlighted fields. Press the Tab key to navigate among fields. Once you modify a field, press the Tab or Return key to make the new value take effect.

Changes Persist for Duration of Current Session

Changes to these fields apply only for the duration of the conductor or server's current session; when the conductor or server exits, changes made to these fields will be lost. To make persistent changes, edit the conductor and server initialization files, as described in the section, About Initialization Files, on page 95.

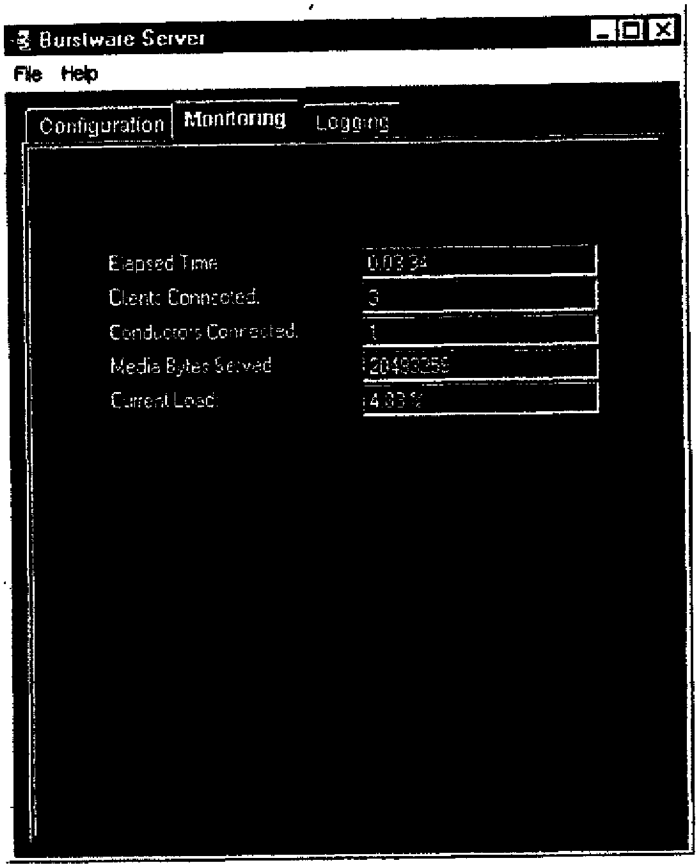
Monitoring

Click the Monitoring tab to display the Burstware Conductor and Burstware Server monitoring screens. These screens, illustrated in Figure 9 and Figure 10, respectively, report a variety of information about the status of conductors and servers.



Elapsed Time: 0:03:23
Servers Configured: 3
Servers Available: 2
Clients Connected 1
Managed Bandwidth: 180
Excess Capacity: 177

Figure 9: Conductor monitoring screen



Elapsed Time 0:03:34

Clients Connected 3

Conductors Connected 1

Media Bytes Served 20493256

Current Load 4.03 %

Figure 10: Server monitoring screen

Explanation of Conductor Monitoring Fields

The conductor monitoring screen contains these fields:

Field	**Function**
Elapsed Time	Reports how long the conductor has been running, in the format hh:mm:ss.
Servers Configured	Reports the number of Burstware Servers managed by the conductor. These servers are in the serverList field on the configuration screen.
Servers Available	Reports the number of servers currently running to which the conductor is routing new player connection requests. Servers can be in one of three states: available, standby, or offline. See the section, Load Balancing and Failover with Available and Standby Servers, on page 38 for details.
Clients Connected	Reports the number of burst-enabled players currently connected, across all available servers this conductor manages.
Managed Bandwidth	Reports the sum of the managed bandwidths of all available servers this conductor manages, in Mbps.
Excess Capacity	Reports the sum of the excess capacity across all available servers this conductor manages, in Mbps. A server's excess capacity is equal to its managed bandwidth minus its percentage current load. Excess Capacity measures a conductor's ability to accept new play requests. See Current Load in the table on page 183 for more information.

Explanation of Server Monitoring Fields

The server monitoring screen contains these fields:

Elapsed Time

Reports the elapsed time since the server was started.

Clients Connected

Reports the number of burst-enabled players connected to the server.

Conductors Connected

This field reports the number of Burstware Conductors managing the server. This number also includes the number of Burstware Consoles monitoring the server.

Media Bytes Served

Reports the number of multimedia bytes delivered by the server since it was started.

Current Load

Reports the percentage of the server's managed bandwidth currently committed to service clients.

Because a Burstware Server delivers data in faster than real time, it uses as much bandwidth as network conditions and client buffer levels allow, up to the value of the server's managedBandwidth parameter. Current load does not measure the amount of bandwidth the server is currently using. Rather, it measures the minimum bandwidth needed to maintain a high quality viewing experience for all clients. For more information on managed bandwidth, see the section, Setting Managed Bandwidth, on page 45.

What to Look for When Monitoring

When monitoring your deployment, verify the following:

- All components are running the correct version
- Each active conductor and its standby conductor report the same numbers for the number of servers configured and available, the amount of managed bandwidth, and the number of clients connected

  This verifies that the two conductors are configured and operating identically.
- All servers report the correct number of connected conductors
- If a server reports that clients are connected, then the Media Bytes Served field is updating

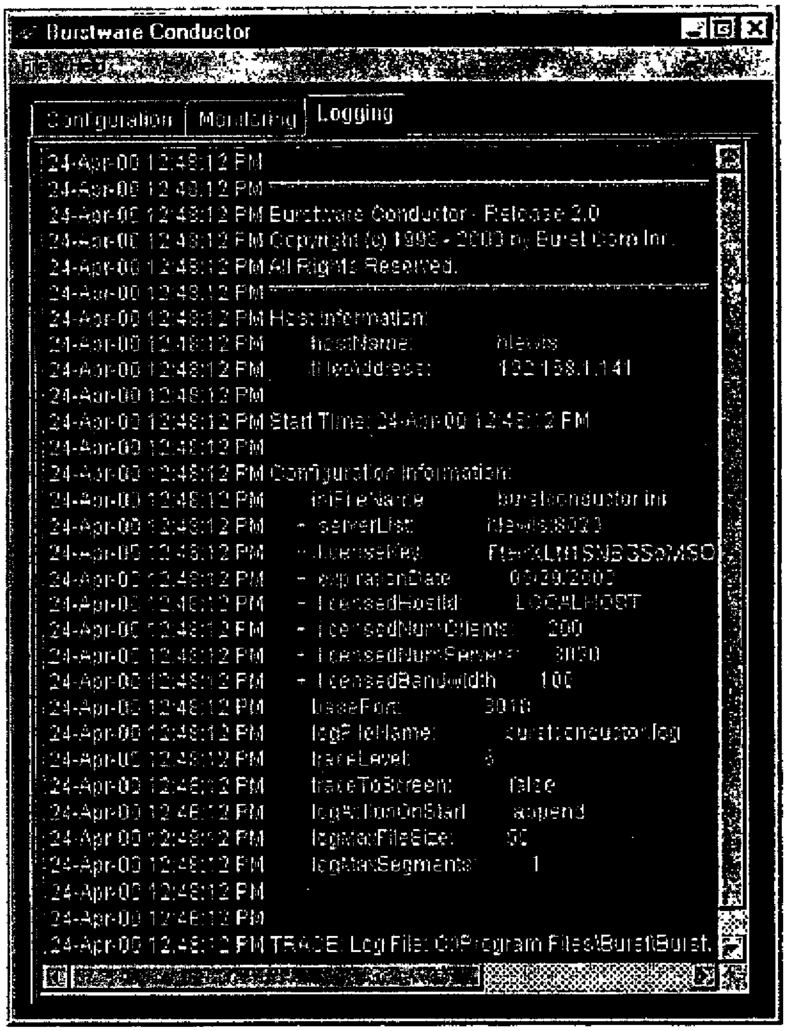
  If players are connected but the Media Bytes Served field is not updating, it could mean that every player is stopped. But it could also indicate a problem with the server or the network connection. In this case, you may want to stop and restart the server.

Remote Monitoring with the Burstware Console

The Burstware Console is a standalone application that allows remote monitoring of conductors and servers. The Burstware Console is included as part of Burst.Com's Professional Services CD ROM. For more information on the Burstware Console, contact your sales representative.

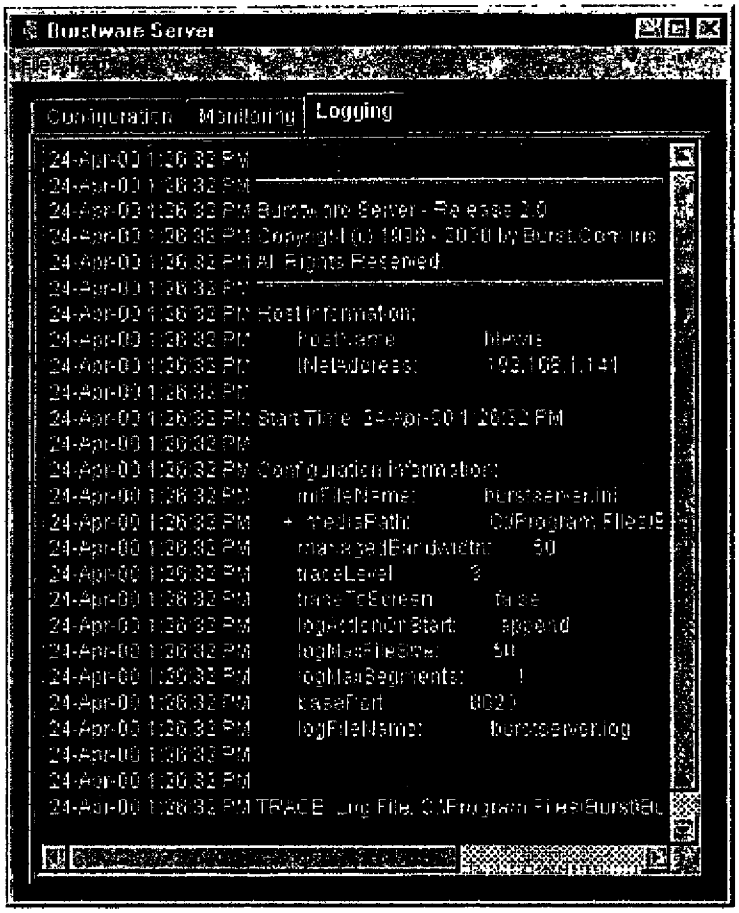
Logging

Click the Logging tab to bring up the Burstware Conductor and Burstware Server logging screens, illustrated in Figure 11 and Figure 12, respectively. Use the logging screens to review static information, such as the value of configuration parameters and the version of the conductor or server, and dynamic information, such as play requests and errors.



```
24-Apr-00 12:48:12 PM
24-Apr-00 12:48:12 PM ----------------------------------------------------------
24-Apr-00 12:48:12 PM Burstware Conductor - Release 2.0
24-Apr-00 12:48:12 PM Copyright (c) 1998 - 2000 by Burst.com Inc.
24-Apr-00 12:48:12 PM All Rights Reserved.
24-Apr-00 12:48:12 PM ----------------------------------------------------------
24-Apr-00 12:48:12 PM Host information:
24-Apr-00 12:48:12 PM      hostName:          rlewis
24-Apr-00 12:48:12 PM      ipAddress:         192.168.1.141
24-Apr-00 12:48:12 PM
24-Apr-00 12:48:12 PM Start Time: 24-Apr-00 12:48:12 PM
24-Apr-00 12:48:12 PM
24-Apr-00 12:48:12 PM Configuration information:
24-Apr-00 12:48:12 PM      iniFileName          burstconductor.ini
24-Apr-00 12:48:12 PM    - serverList:         rlewis:8023
24-Apr-00 12:48:12 PM    - licenseKey:         FtcrXLM1SNBGSoMSO
24-Apr-00 12:48:12 PM    - expirationDate       06/29/2000
24-Apr-00 12:48:12 PM    - licensedHostId        LOCALHOST
24-Apr-00 12:48:12 PM    - licensedNumClients:     200
24-Apr-00 12:48:12 PM    - licensedNumServers:      3030
24-Apr-00 12:48:12 PM    - licensedBandwidth      100
24-Apr-00 12:48:12 PM      basePort:          3010
24-Apr-00 12:48:12 PM      logFileName:          burstconductor.log
24-Apr-00 12:48:12 PM      traceLevel:        3
24-Apr-00 12:48:12 PM      traceToScreen:        false
24-Apr-00 12:48:12 PM      logActionOnStart      append
24-Apr-00 12:48:12 PM      logMaxFileSize:       50
24-Apr-00 12:48:12 PM      logMaxSegments:        1
24-Apr-00 12:48:12 PM
24-Apr-00 12:48:12 PM
24-Apr-00 12:48:12 PM TRACE: Log File: C:\Program Files\Burst\Burst.
```

Figure 11: Conductor logging screen



```
24-Apr-00 1:26:32 PM
24-Apr-00 1:26:32 PM ------------------------------------------------
24-Apr-00 1:26:32 PM Burstware Server - Release 2.0
24-Apr-00 1:26:32 PM Copyright (c) 1998 - 2000 by Burst.Com Inc
24-Apr-00 1:26:32 PM All Rights Reserved.
24-Apr-00 1:26:32 PM ------------------------------------------------
24-Apr-00 1:26:32 PM Host Information:
24-Apr-00 1:26:32 PM        HostName:            blewis
24-Apr-00 1:26:32 PM        InetAddress:         192.168.1.141
24-Apr-00 1:26:32 PM
24-Apr-00 1:26:32 PM Start Time: 24-Apr-00 1:26:32 PM
24-Apr-00 1:26:32 PM
24-Apr-00 1:26:32 PM Configuration Information:
24-Apr-00 1:26:32 PM        iniFileName:         burstserver.ini
24-Apr-00 1:26:32 PM     +  mediaPath:           C:\Program Files\B
24-Apr-00 1:26:32 PM        managedBandwidth:    50
24-Apr-00 1:26:32 PM        traceLevel:          3
24-Apr-00 1:26:32 PM        traceToScreen:       false
24-Apr-00 1:26:32 PM        logActionOnStart:    append
24-Apr-00 1:26:32 PM        logMaxFileSize:      50
24-Apr-00 1:26:32 PM        logMaxSegments:      1
24-Apr-00 1:26:32 PM        basePort:            8020
24-Apr-00 1:26:32 PM        logFileName:         burstserver.log
24-Apr-00 1:26:32 PM
24-Apr-00 1:26:32 PM
24-Apr-00 1:26:32 PM TRACE Log File: C:\Program Files\BurstBL
```

Figure 12: Server logging screen

Screens Can Display All Log Events or a Subset of Events

The logging screen can display either all the information contained in the log file, or just a subset of that information, depending on the value of the traceToScreen configuration parameter:

- If traceToScreen is set to true, the logging screen displays everything the log contains and updates this screen as events occur
- If traceToScreen is set to false, the logging screen displays only static information from the log file, such as the name of the log file, the conductor or server version number, information about the host machine, and configuration parameter values
- The default is set to false

The next section discusses the contents of the log file in more detail.

What's In a Log File?

Burstware Conductor and Burstware Server write log entries to their respective log files whenever notable conductor or server events occur. Depending on the level of detail set by the traceLevel configuration parameter, these events include:

- Commands—sign-on, open, exit—sent by any burst-enabled player to the conductor or server, including:
  - Media files requested by clients
  - Statistics and time stamps for each command
- Errors

  Error entries begin with "ERROR" and other entries begin with "TRACE", as shown in Sample Server Log, on page 190.
- Dynamic changes to configuration parameters
- Detailed statistics for each player at the end of its session (server log file only), such as how long it was connected to the server, which files it requested, and how many bytes of data the server delivered to it

You can use the Burstware Log Toolkit—available from Burst.Com Professional Services—to extract log information, consolidate information from multiple logs, and convert the information into formats that can be used to report such information as:

- Who watched which videos and when
- Which errors occurred and when
- Video quality
- Client throughput
- Server loads over time
- Which files are most popular

You can also use the log toolkit to support automated billing and other applications. Contact your sales representative or email sales@burst.com for more information on the log toolkit.

Sample Conductor Log

```
29-Mar-00 5:07:57 PM
29-Mar-00 5:07:57 PM ************************************************************
29-Mar-00 5:07:57 PM Burstware Conductor - Release 2.0
29-Mar-00 5:07:57 PM Copyright (c) 1998 - 2000 by Burst.Com Inc.
29-Mar-00 5:07:57 PM All Rights Reserved.
29-Mar-00 5:07:57 PM ************************************************************
29-Mar-00 5:07:57 PM Host information:
29-Mar-00 5:07:57 PM      hostName:          barney
29-Mar-00 5:07:57 PM      INetAddress:        982.239.3.122
29-Mar-00 5:07:57 PM
29-Mar-00 5:07:57 PM Start Time: 29-Mar-00 5:07:57 PM
29-Mar-00 5:07:57 PM
29-Mar-00 5:07:57 PM Configuration information:
29-Mar-00 5:07:57 PM      iniFileName:        burstconductor.ini
29-Mar-00 5:07:57 PM    + serverList:         fred:8020
29-Mar-00 5:07:57 PM    + licenseKey:         < your license key here>
29-Mar-00 5:07:57 PM    + expirationDate:       12/31/2000
29-Mar-00 5:07:57 PM    + licensedHostId:       LOCALHOST
29-Mar-00 5:07:57 PM    + licensedNumClients:    500
29-Mar-00 5:07:57 PM    + licensedNumServers:     3
29-Mar-00 5:07:57 PM    + licensedBandwidth:    3
29-Mar-00 5:07:57 PM      basePort:          8018
```

```
29-Mar-00 5:07:57 PM    logFileName:        burstconductor.log
29-Mar-00 5:07:57 PM    traceLevel:        6
29-Mar-00 5:07:57 PM    traceToScreen:       false
29-Mar-00 5:07:57 PM    logActionOnStart:     append
29-Mar-00 5:07:57 PM    logMaxFileSize:      50
29-Mar-00 5:07:57 PM    logMaxSegments:       1
29-Mar-00 5:07:57 PM
29-Mar-00 5:07:57 PM
29-Mar-00 5:07:57 PM TRACE: Log File: C:\Program
Files\Burst\Burstware\Conductor\burstconductor.log
29-Mar-00 5:07:59 PM TRACE: Using access control plugin Sample JAM 1.0
29-Mar-00 5:08:01 PM TRACE: Unable to connect to Server: fred:8020
29-Mar-00 5:09:09 PM TRACE: Scheduler: Summary: Servers Configured: 1 Servers
Available: 0 Clients Connected: 0 Managed Bandwidth: 0 Excess Capacity: 0
29-Mar-00 5:14:22 PM TRACE: Setting Configuration parameter: traceToScreen to value:
true
29-Mar-00 5:14:32 PM TRACE: Setting Configuration parameter: traceLevel to value: 5
2|SignOn|1|0|1|0|231691605|100000|192.168.1.134|wilma|false|WMPUser_{3929d1e0-dbe0-
11d2-8262-00105a179b68}_300|wilma
flintstone|none|1.9|none|BURST_DOMAIN|mplayer2.exe
30-Mar-00 12:38:26 PM TRACE: Client2: Command: 2|Open|1|0|u2.mp3|false|1500
30-Mar-00 12:38:26 PM TRACE: Client2: Forwarding request to serverList: fred:8020
30-Mar-00 12:38:26 PM TRACE: Client2: Command: 2|Exit|1|0
30-Mar-00 12:44:53 PM TRACE: Scheduler: Summary: Servers Configured: 1 Servers
Available: 1 Clients Connected: 1 Managed Bandwidth: 50 Excess Capacity: 6552600
30-Mar-00 12:46:53 PM TRACE: Scheduler: Summary: Servers Configured: 1 Servers
Available: 1 Clients Connected: 0 Managed Bandwidth: 50 Excess Capacity: 6553600
```

Sample Server Log

```
29-Mar-00 5:17:18 PM
29-Mar-00 5:17:18 PM ************************************************************
29-Mar-00 5:17:18 PM Burstware Server - Release 2.0
29-Mar-00 5:17:18 PM Copyright (c) 1998 - 2000 by Burst.Com Inc.
29-Mar-00 5:17:18 PM All Rights Reserved.
29-Mar-00 5:17:18 PM ************************************************************
29-Mar-00 5:17:18 PM Host information:
29-Mar-00 5:17:18 PM      hostName:          fred
29-Mar-00 5:17:18 PM      INetAddress:        267.123.9.0
29-Mar-00 5:17:18 PM
29-Mar-00 5:17:18 PM Start Time: 29-Mar-00 5:17:18 PM
29-Mar-00 5:17:18 PM
29-Mar-00 5:17:18 PM Configuration information:
29-Mar-00 5:17:18 PM      iniFileName:        burstserver.ini
29-Mar-00 5:17:18 PM    + mediaPath:          C:/Program Files/Burst/Burstware/
Media;d:/media
29-Mar-00 5:17:18 PM      managedBandwidth:     50
29-Mar-00 5:17:18 PM      traceLevel:        3
29-Mar-00 5:17:18 PM      traceToScreen:        false
29-Mar-00 5:17:18 PM      logActionOnStart:      append
29-Mar-00 5:17:18 PM      logMaxFileSize:       50
29-Mar-00 5:17:18 PM      logMaxSegments:        1
29-Mar-00 5:17:18 PM      basePort:          8020
29-Mar-00 5:17:18 PM      logFileName:        burstserver.log
29-Mar-00 5:17:18 PM
29-Mar-00 5:17:18 PM
29-Mar-00 5:17:18 PM TRACE: Log File:
C:\ProgramFiles\Burst\Burstware\Server\burstserver.log
1|Signon|1|0|1|0|burstware|267.123.9.030|barney|
29-Mar-00 5:17:35 PM TRACE: Setting Configuration parameter: managedBandwidth to
value: 80
29-Mar-00 5:17:45 PM TRACE: Setting Configuration parameter: traceToScreen to value:
true
29-Mar-00 5:17:50 PM TRACE: Setting Configuration parameter: managedBandwidth to
value: 30
29-Mar-00 5:55:38 PM TRACE: Using burst.util.LargeRandomAccessFile
29-Mar-00 5:56:53 PM TRACE: Client3: Command:
3|SignOn|1|0|1|0|221227690|100000|267.123.9.040|wilma|false|WMPUser_{3929d1e0-dbe0-
11d2-8262-00105a179b68}_138|wilma flintstone|none|1.9|none|IVT_DOMAIN|mplayer2.exe
29-Mar-00 5:56:53 PM TRACE: Client3: Command: 3|Open|1|0|u2.mp3|false|1500
29-Mar-00 5:56:54 PM TRACE: Client3: Session Statistics:
```

```
                  Client3: Connection Time:1 (seconds)
                  Client3: Session Statistics: End.
29-Mar-00 5:56:54 PM TRACE: Client3: End of Session.
29-Mar-00 5:57:56 PM ERROR: Console1: Console: Unexpected error:
java.net.SocketException: Connection reset by peer
```

The longer your deployment is running and the greater the number of logged events, the larger the log files can grow. By restricting the maximum size of conductor and server logs, you can minimize the strain on available storage and simplify the search for information within a log file. The next section explains how to do this.

Log File Management

Burstware Conductor and Burstware Server provide three log-related server configuration parameters in their initialization files to help you manage the log:

- logActionOnStart
- logMaxFileSize
- logMaxSegments

When a Conductor or Server Restarts

The logActionOnStart configuration parameter determines what happens to the previous session's log file once a new session begins. A conductor or server can do one of three things with the previous session's log file at this time:

- Append to the file
- Delete the file and start a new one
- Convert the file to a named and numbered log segment and start a new log file

  The segment name is determined by log file naming conventions. See the section, Log File Naming Conventions, on page 192.

The default behavior is to append to the previous session's log file.

While a Conductor or Server Is Running

When the log file grows to the size specified in **logMaxFileSize**, Burstware Conductor or Burstware Server converts it to a named and numbered segment and starts a new log file. Use the **logMaxFileSize** configuration parameter to restrict the size of the current session's log file.

The default setting allows the log file to grow to 50 MB before the conductor or server converts it to a segment and starts a new log file.

When the number of segments reaches **logMaxSegments**, the conductor or server deletes the oldest segment to make room for the new segment. Use logMaxSegments to restrict the number of segments that can accumulate.

If you do not want the conductor or server to save log segments, set logMaxSegments to 0.

Chapter 6, Configuring Conductors and Servers. contains more information about log-related conductor and server configuration parameters.

Log File Naming Conventions

You can specify the file where Burstware Conductor or Burstware Server writes log entries with the **logFileName** configuration parameter. You can qualify the file with either a full path or a path relative to the directory where the conductor or server is started:

The default for conductors is **burstconductor.log** in the directory where the conductor is started. The default for servers is **burstserver.log** in the directory where the server is started.

NOTE: If you run more than one server or conductor on a machine, each should have a unique log file name.

Segment Naming

When the conductor or server converts the log file to a segment, the segment name takes this form:

<logFileName>_<hostname>_<portnum>_<series_num>.<extension>

When the conductor or server creates a segment, it determines the series number by examining all existing segments, locating the greatest series number, and incrementing it by 1.

Example

Consider this scenario:

- Burstware Server is running on a machine named "server1" on port 8020
- logFileName is set to its default, "burstserver.log"
- logActionOnStart is set to "save"
- logMaxFileSize is set to 3 MB
- logMaxSegments is set to 2
- There are no log segments present

The server is restarted, beginning a new server session. The server converts the log file to a segment with this name:

```
burstserver_server1_8020_1.log
```

and starts a new log file. The new log file grows to 3 MB. The server converts the log file to a segment with this name:

```
burstserver_server1_8020_2.log
```

and starts a new log file. The new log file again grows to 3 MB.

Because two segments have already accumulated, the server must delete the oldest segment before creating a new one. The server deletes the segment named burstserver_server1_8020_1.log and writes the log file to this name:

```
burstserver_server1_8020_3.log
```

CHAPTER 11

Burstware Security

This chapter describes the security features available with Burstware. The key features are:

- Password protection of a Burstware domain

  Password protection of a domain prevents unauthorized users from viewing or changing the settings that affect the operation of Burstware.

NOTE: A Burstware domain consists of a conductor, its standby, and the servers they manage. By assigning a password to a Burstware domain, you control access to all of the servers and conductors in that domain.

- Implement a range of content access control mechanisms

  Burstware allows the implementation of access controls—either simple or complex—which grant or deny an end user's access to specific content items.

Password-Protecting a Burstware Domain

Burstware provides password protection at the domain level. Assigning a password to a Burstware domain has the effect of restricting access to all servers and conductors in the domain.

By password-protecting a domain, you can:

- Prevent unauthorized conductors and servers from becoming part of your Burstware deployment
- Isolate Burstware domains from one another

  For example, you can prevent a conductor from one domain and a conductor from another domain from inadvertently managing the same server.

- Prevent unauthorized remote monitoring of your Burstware deployment by a Burstware Console

  See the section, Deployments Using the Burstware Console, on page 197.

Default Password for a Domain

Until you explicitly assign a password to your Burstware domain, as described in the following paragraph, the default password for the domain is burstware.

Assigning a Password to a Domain

You set a password for a component—Burstware Conductor or Burstware Server—by setting the consolePassword parameter in the component's initialization file.

NOTE: A Burstware Conductor must share a common password with any server it attempts to manage. Each server in a domain must have the same password as the conductor that manages it.

Conductor and server initialization files do *not* contain the consolePassword parameter by default—you must add a new line to the initialization file.

To assign a password to a domain:

1. Choose a password value for the domain.
2. Edit the initialization file for each conductor, server, and remote console in the domain, setting consolePassword to the value you chose. To do this, add the following line to the initialization file:

   `consolePassword=`*myPassword*

   where *myPassword* is the password you have chosen. It is important to end this line with a carriage return.

Environments with Multiple Burstware Domains

If you have multiple Burstware domains, Burst.Com recommends that you assign a unique password to each domain in your deployment.

Deployments Using the Burstware Console

The Burstware Console, which allows remote monitoring of servers and conductors, also requires a password to connect to the components it monitors. A Burstware Console must share a common password with any conductor or server it attempts to monitor.

The Burstware Console is included as part of Burst.Com's Professional Services CD-ROM. For more information on the Burstware Console, contact your sales representative.

Burstware and Virtual Private Networks

If your Burstware deployment requires stringent security, Burst.Com recommends that you run all conductors, servers, and consoles as part of the same Virtual Private Network (VPN). This ensures encryption of all communication between Burstware components and prevents unauthorized users from obtaining the domain password.

Burstware Access Control

Access control is the ability to fulfill or deny a play request made to the Burstware system, based on a definable policy or set of policies. This allows you to control access to your content files, ensuring that your content is secured against unauthorized viewing.

The Burstware access control system allows you to implement policies that secure your content against unauthorized access, while permitting authorized users to view the content to which you allow them access.

By default, Burstware grants all requests for multimedia file access, thus allowing every end user to play back any video stored on a Burstware server. However, in many circumstances, this open door policy is too permissive. You may want to selectively restrict or grant access to authorized users, for a portion or for all of your content.

Burstware Conductor includes an access control interface that accepts plug-in components. This interface enables you to implement access control policies ranging from very simple to highly sophisticated, depending on your needs. Your options include:

- Using the default open door access control referred to above
- Using an access control module available from Burst.Com or a third party vendor

  Burst.Com provides, as a professional service, its Simple Access Control Module (SAM), enabling you to define access rules in an easy-to-use text format similar to that used by firewalls. This offering includes the Java source code for SAM, allowing customization of SAM's policies. For details, contact your Burst.Com sales representative or send an email to sales@burst.com.
- Write your own access control module

  The document called "Writing a Custom Access Control Module", on the Burst.Com website—www.burst.com—contains the information you need to write your own Burstware custom Java access control module (JAM).
- Link to an external authentication system or a relational database.

  For stringent or unique access control requirements, Burstware Conductor's access control interface allows you to link to an external tool for access control.

The sections that follow provide background information about Burstware access control and describe the Burstware access control architecture.

Access Control Applications

Burstware's access control functionality is useful in a variety of applications. Several of them are described in this section.

Pay-Per-View (the virtual video store)

In this scenario, you are a content provider and want to make videos available for viewing over the Internet, but only after the end user has paid for a given viewing. You have a web site that collects payment via a credit card, and then notifies the Burstware access control system that access by this particular user (identified by an IP Address or a unique player ID, for example) is allowed one viewing of a particular movie, or unlimited viewing for the next 24 hours. After this notification, the web page begins playback of the video through a burst-enabled player. The play request goes through the access control module of the conductor, which grants play requests only to authorized viewers.

Corporate Access Levels

In a corporation, access to videos may be restricted by department. For example:

- A CEO's monthly video newsletter should be made available to all employees
- Only Human Resource department personnel should be able to view sensitive personnel videos
- Only the Vice-President of Sales should be able to view the sales pipeline video reports prepared by regional sales managers

The Burstware access control module can use access control lists (ACLs) to give different people access to different videos, depending on the corporate group or groups to which they belong.

Subscription-based Access

In this scenario, an online brokerage company provides video reports of stock and market analysis. Brokerage customers can subscribe to video reports on a monthly basis and choose from varying levels of service. Higher levels of service are reserved for premium customers, and offer more comprehensive and more up-to-date analysis of a wider range of stocks. The access control system ensures that only authorized users view these videos, based on whether they have paid their monthly fee (or have done sufficient trading in the preceding month) and their level of service.

Burstware Access Control Overview

There are three components to the Burstware access control system. Every play request involves all three components. Each component, and the role it plays in implementing access control, is described below:

- Burst-enabled player

The burst-enabled player gathers data about each play request, including identifying information about the user making the request. The burst-enabled player then sends this information to the conductor.

- The access control module

  The access control module is a plug-in component for the conductor that can be customized or replaced. The access control module applies access control policies for each play request and determines whether to fulfill or deny the play request. The default access control module that ships with Burstware applies a simple policy—it allows *all* play requests to be fulfilled.

- The access control interface

  The access control interface is built into Burstware Conductor. It passes the data gathered by the burst-enabled player to the conductor's access control module. The access control module makes the decision to fulfill or deny the play request; the access control interface enforces this request by sending the player machine the server list to fulfill the play request, or an error message when the play request has been denied.

Information Used in Access Control

As part of the access control process, information is collected from a burst-enabled player machine and the URL and is passed to the conductor. The following are examples of some of the information passed:

- Client IP Address
- Media File Name
- Unique identifier of the burst-enabled player
- Whether the burst-enabled player is using a memory or disk buffer
- Custom information in an **AccessControlString** parameter

For details about how to pass custom information using the **AccessControlString**, see the section, Using Custom Information for Access Control Decisions. on page 204.

Default Burstware Access Control

Figure 13 shows the default access control sequence.

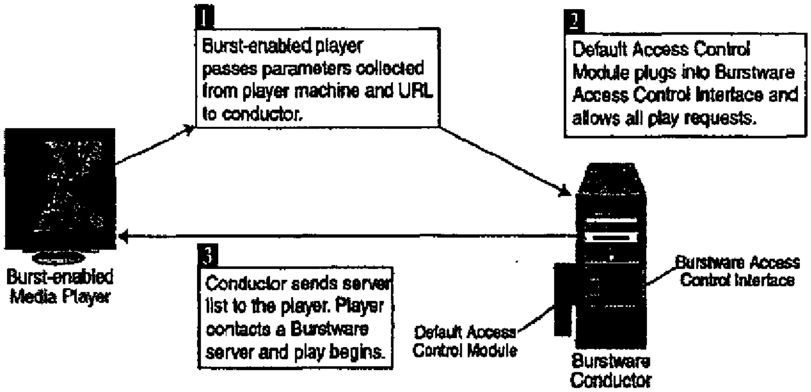

1
Burst-enabled player passes parameters collected from player machine and URL to conductor.
2
Default Access Control Module plugs into Burstware Access Control Interface and allows all play requests.
3
Conductor sends server list to the player. Player contacts a Burstware server and play begins.
Burst-enabled Media Player
Default Access Control Module
Burstware Access Control Interface
Burstware Conductor

Figure 13: Default access control sequence

Other Access Control Options

If you want more extensive access control, you can use an off-the shelf access control module, such as the Simple Access Control Module (SAM) available as a professional service from Burst.Com. You can also write your own custom access control module in Java (custom JAM) and plug it into the Burstware Conductor. For more information on writing a custom JAM, see the document called "Writing a Custom Access Control Module", on the Burst.Com website—www.burst.com.

The diagrams in the next section illustrate the access control sequence using these options.

Simple Access Control Module

Figure 14 shows an implementation of access control using the Simple Access Control Module (SAM) provided by Burst.Com.

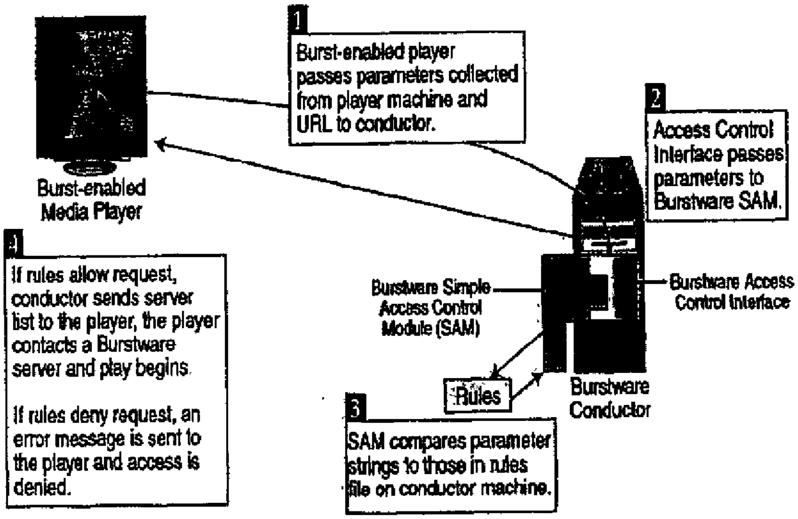

1
Burst-enabled player passes parameters collected from player machine and URL to conductor.
2
Access Control Interface passes parameters to Burstware SAM.
Burst-enabled Media Player
4
If rules allow request, conductor sends server list to the player, the player contacts a Burstware server and play begins.
If rules deny request, an error message is sent to the player and access is denied.
Burstware Simple Access Control Module (SAM)
Burstware Access Control Interface
Rules
Burstware Conductor
3
SAM compares parameter strings to those in rules file on conductor machine.

Figure 14: Simple Access Control Module (SAM)

Custom Access Control Module

Figure 15 shows an implementation of access control using a custom Java access control module.

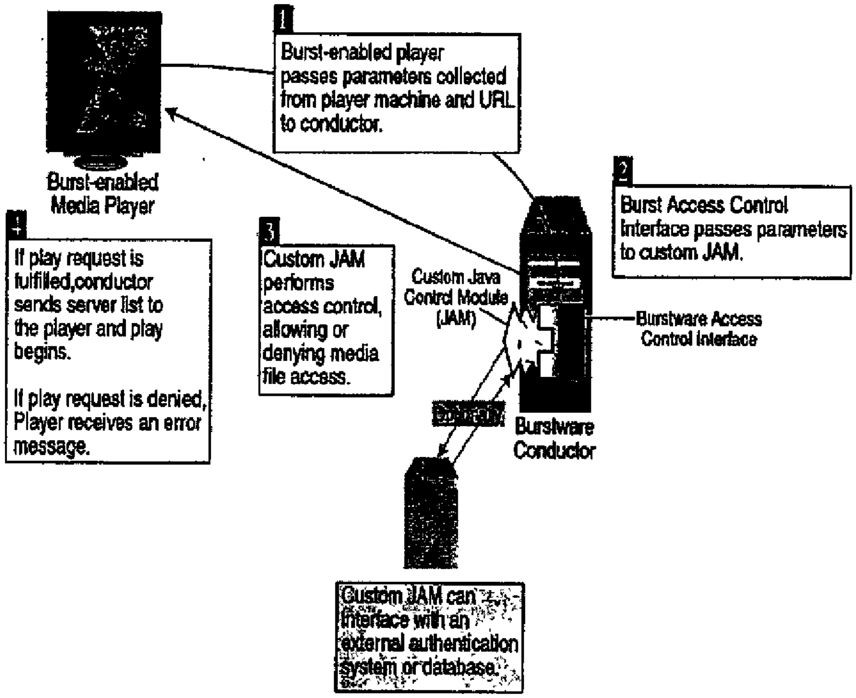

1
Burst-enabled player passes parameters collected from player machine and URL to conductor.
Burst-enabled Media Player
2
Burst Access Control Interface passes parameters to custom JAM.
4
If play request is fulfilled,conductor sends server list to the player and play begins.
If play request is denied, Player receives an error message.
3
Custom JAM performs access control, allowing or denying media file access.
Custom Java Control Module (JAM)
Burstware Access Control Interface
Burstware Conductor
Custom JAM can interface with an external authentication system or database.

Figure 15: Access control using a custom Java access control module (JAM)

Using Custom Information for Access Control Decisions

Burstware passes a large amount of information about the play request and the client to the access control module, which it uses to decide whether or not to grant the request. In addition to the information Burstware obtains, you can also pass information from the client machine to the access control module using the **AccessControlString** parameter. This parameter is optionally specified in the URL of the play request.

Access Control String

You can use any scheme to generate the access control string, such as an algorithm implemented in JavaScript. The access control string is passed unchanged to the Burstware conductor's access control module.

A Burstware application can use an access control string in a number of ways:

- As an encrypted representation of a user name and password

  For example, when the end user requests a multimedia file, the application might query the end user for a user name and password, and encrypt this information prior to requesting the media file from Burstware. The encrypted name and password are then passed into Burstware as the **AccessControlString** parameter, so the conductor's access control module can perform user authentication and determine if the request should be fulfilled.

- To store any information useful in logging, since the access control string is written verbatim to the conductor log

When using the access control string, the application generating the URL (for example, a web page containing a burst-enabled player) cooperates with the conductor's access control module. Typically, the application generating the URL uses the access control string as a "key" and the access control module decides whether the key "fits" the play request. This allows authentication to be performed on each request— the access control module uses the access control string to authenticate the user.

Burstware's Simple Access Control Module (SAM) enables you to use regular-expression string matching to decide whether the key "fits". For systems that require more secure or more sophisticated authentication, a custom Java access control module (JAM) must be used.

For more information on how to use the **AccessControlString** parameter, see the section, AccessControlString Parameter, on page 152. The document called "Writing a Custom Access Control Module", on the Burst.Com website—www.burst.com—contains more information on custom Java access control interfaces.

Access Control Logs

The Burstware Conductor log file contains, among other things, a record of all play requests, as well as a record for each request denied by the access control module. The conductor log—called **burstconductor.log** by default—is useful in monitoring your access control system.

To analyze a conductor log, you can either examine the log file directly or use the Burstware Log Toolkit—available as a professional service from Burst.Com—to help you analyze the logs.

The following is an excerpt from a conductor log file, containing these three lines:

- A SignOn message (phase one of a play request)
- An Open message for the same client (phase two of a play request)
- An access denied message for the same client, indicating the play request was denied by the access control module

```
25-Feb-00 6:37:40 PM TRACE: Client1: Command:
1|SignOn|1|0|1|0|1962174464|100000|192.168.1.1|shepher
d|false|WMPuser_{340a95c0-ac0a-11d3-b7c2-
0060087678c2}_286|wilma flintstone|none|1.2.5

25-Feb-00 6:37:40 PM TRACE: Client1: Command:
1|Open|1|0|celine.mpg|false|1500

25-Feb-00 6:37:40 PM ERROR: Client1: Access denied:
Account expired.
```

```
Client1: Access denied details: Rule 27 failed.
```

An “access denied” log entry includes:

- The client session ID
- The error message displayed to the end user explaining the denial
- An internal error message giving details of the denial
  When using Burst.Com’s SAM, this error message includes the text of the rule responsible for the denial.

For more information on conductor logs, see the section, What’s In a Log File?, on page 187.

Index