



KnowNow LiveServer Administration Guide

KnowNow, Inc.
997 East Arques Avenue
Sunnyvale, California 94085 USA

Legal Notices

Copyright © 2000–2008 KnowNow, Inc. All rights reserved.

All rights reserved. No part of this work covered by copyright herein may be reproduced in any form or by any means—graphic, electronic, or mechanical—including photocopying, recording, taping, or storage in an information retrieval system, without prior written permission of the copyright owner.

KnowNow™, the KnowNow logo, and the KnowNow icon are trademarks of KnowNow, Inc.. Other KnowNow product names used herein may be trademarks or registered trademarks of KnowNow, Inc. .

FLEXIm™ is a trademark or registered trademark of Macrovision© Corporation.

Linux is a registered trademark of Linus Torvalds.

Microsoft©, Windows, Active Directory®, and Active Directory® Application Mode are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Sleepycat and the names of Sleepycat Software products are trademarks, registered trademarks, or service marks of Sleepycat Software, Inc..

Sun© Solaris Operating System is a trademark or registered trademark of Sun Microsystems, Inc. in the United States or other countries.

SUSE is a trademark of SUSE LINUX Products GmbH, a Novell business. Novell is a registered trademark of Novell, Inc. in the United States and other countries.

All other brand, company, and product names referenced in this publication may be trademarks, registered trademarks, or service marks of their respective holders and are used here for informational purposes only.

Statement of Conditions

KNOWNOW, INC. PROVIDES THIS PUBLICATION “AS IS” WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE IMPLIED WARRANTIES OR CONDITIONS OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. IN NO EVENT SHALL KNOWNOW BE LIABLE FOR ANY LOSS OF PROFITS, LOSS OF BUSINESS, LOSS OF USE OR DATA, INTERRUPTION OF BUSINESS, OR FOR DIRECT, INDIRECT, SPECIAL, INCIDENTAL, OR CONSEQUENTIAL DAMAGES OF ANY KIND, EVEN IF KNOWNOW HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES ARISING FROM ANY DEFECT OR ERROR IN THIS PUBLICATION OR IN THE KNOWNOW SOFTWARE.

KnowNow, Inc. may revise this publication from time to time without notice. Some states or jurisdictions do not allow disclaimer of express or implied warranties in certain transactions; therefore, this statement may not apply to you.

Version

LiveServer version: Release 3.5

Document version: May 1, 2008

Table of Contents

Preface	xvii
What This Guide Covers	xvii
Additional Sources of Information	xviii
Document Conventions	xix
How To Contact KnowNow, Inc.	xx
Chapter 1 Introducing KnowNow's LiveServer	1
LiveServer Concepts	2
Architectural Components	3
LiveServers	3
LiveServer Modules	4
Connectors	4
LiveAdapters	5
LiveServer Terminology	5
Events	6
Topics	6
Metadata Subtopics	7
Journal Topics	8
Routes	9
Subscriptions and Routes	9
Event Lifetimes and Initial Route Population	10
How LiveServer Can Be Used	11
Basic Use	11
Aggregation	12
Chapter 2 Installing KnowNow's LiveServer	15
System Requirements and Supported Platforms	16
LiveBrowser System Requirements	17
Required Linux Privileges	17
Upgrading a Previous Installation on Any Platform	18
Installing and Uninstalling KnowNow Products	19
Installation Order for KnowNow Products	19
Uninstall Order for KnowNow Products	20

- Installing LiveServer on Windows22
 - Starting LiveServer in Windows.....29
 - Stopping and Restarting LiveServer in Windows31
 - Changing the LiveServer Windows Desktop Installation to a Service32
 - Removing the LiveServer Service in Windows33
- Installing LiveServer on Linux35
 - Pre-Installation Tasks.....35
 - Required C/C++ Libraries for Red Hat Enterprise Linux AS.....35
 - Additional Required C/C++ Libraries for Red Hat Enterprise Linux36
 - Installing LiveServer on Linux Step-by-Step38
 - Setting the Number of Client Connections39
- Verifying that LiveServer is Running40
- Chapter 3 Introducing the KnowNow System Administration Console ... 45**
- Accessing the KnowNow System Administration Console46
- The Main KnowNow System Administration Window48
 - Accessing the Capabilities of the KnowNow System Administration Console49
 - LiveServer States.....50
 - Configuration Status Field.....50
 - The Reload Button51
- Accessing LiveServer Documentation and Help.....52
- Entering Data and Editing LiveServer Parameters53
 - Specifying Units in Parameters53
 - Tcl Expressions in LiveServer Parameters.....54
 - Special Characters in Parameters.....54
- Chapter 4 Configuring LiveServer..... 55**
- Location of LiveServer Configuration Information.....56
- Default Settings57
- Configuring the Service Network.....58
 - The Service Network Parameters60
 - Always use HTTPS login.....60
 - Domain60
 - Enable HTTP.....61
 - Enable HTTPS61
 - Hostname.....62
 - Clustered Deployments Require This Parameter.....62
 - HTTP Port62
 - HTTP Proxy Server Settings63

HTTPS Port	63
Realm	63
Server Description	63
Show server name in errors	63
SMTP Server	63
Characterizing Your Workload	64
The Workload Parameters	65
Default Expiration Time	65
Default Session Reconnect Time	65
External shutdown timeout	66
Internal shutdown timeout	66
Maximum Event Size	66
Maximum Session Reconnect Time	67
Maximum Tunnel Queue Size	67
Number of processors	67
Peak Deliveries	67
Peak New Connections	68
Peak Publications	68
Publishers	68
Subscribers	68
Recovery	69
Enabling and Configuring Recovery	69
Recovery Parameters	71
Checkpoint Every Write	71
Checkpoint Interval	71
Database Directory	71
Database read timeout	71
Debug database operations	71
Debug file upload manager	72
Enable file uploads (certificates, licenses, etc)	72
Enable string dictionary	72
Persistence WorkQueue Limit	72
Snapshot Directory	72
Disabling Recovery	72
High Availability (Clusters)	74
Configuring SSL	75
SSL Forwarding Parameters	77
Forward using Service Network Certificate	77

Forwarding Certificate	77
Forwarding Key	77
Using OpenSSL to Generate Certificates and Keys	78
Uploading Certificates and Keys	80
Adding Trusted CA Root Certificates	82
Configuring the JavaScript Connector	85
JavaScript Parameters	86
Domain	86
Multi-tiered Deployments	87
Cross-Domain Configuration Example	88
Enable Debug	88
LiveServer URL prefix	89
Enabling Java Support	90
Java Parameters	92
Classpath for Java VM	92
Customer added classes	92
Enable Java-based filters?	92
JDBC drivers	92
KN imported classes	92
KN server classes	93
Options for Java VM	93
Root of Java Home directory	93
Log, Monitor, Process Manager, and Memory	94
Filter Modules	95
Loading Custom Filter Modules	95
Unregistering Custom Filter Modules	97
Restrictions on Using Filter Modules	97
Filter Module Debugging Options	98
Configuring and Managing Licenses	99
Configuring Licenses and Licensing	99
License Parameters	100
Location of license file(s) or license manager	101
Start LMGRD on LiveServers?	101
Monitoring Licenses	102
Permissions (Flat Files, LDAP)	103
Configuring LiveAdapters	104
KnESS Parameters	107
Alarm events expire	107

Anonymous users expire after	107
Database password.....	107
Database user name.....	108
Default database post frequency.....	108
Default page size for paging support.....	108
Enable Exchange Adapter	108
Enable Moreover	108
Enable Reporting Subsystem	108
Enable Sphere Adapter	108
Exchange adapter threads per processor	108
Exchange Adapter UDP Port for receiving updates.....	109
Exchange Server Domain Name.....	109
Exchange Server Host Name	109
Exchange Server Password.....	109
Exchange Server Port.....	109
Exchange Server Protocol.....	109
Exchange Server Public folder Base URI.....	109
Exchange Server Sign-in Account	109
Feed threads per processor	110
JDBC connection URL.....	110
JDBC Driver name	110
List Management Debug	110
Log Level for Exchange Adapter	110
Log Level for LiveAdapter for Reporting.....	110
Log Level for LiveAdapter for RSS	110
Log Level for LiveAdapter for Sphere.....	110
Login Style	111
Login user expire after.....	111
Moreover Metabase URL.....	111
Partition Period for Reporting Database.....	111
RDBMS Used	111
Reporting adapter threads per processor.....	111
Sphere adapter threads per processor.....	111
Sphere Customer ID	111
Sphere Partner ID.....	112
Sphere Server URL.....	112
Sphere URL for Find Related Articles	112
Sphere URL for Search Histogram	112

Threshold for feedwatchdog warnings	112
Chapter 5 Managing Topics	113
Understanding and Managing Topic Space	114
Best Practices in Creating Topics	114
Events	114
Navigating Topic Space	115
Usable /kn_system Subtopics	116
Working with Topics	117
Topics and Topic Properties	119
Viewing Topic Properties	119
The Property Editor	121
Topic Properties and Topic Behaviors	122
Editing a Topic Property	124
Adding a New Topic Property	124
Deleting a Topic Property	125
Creating a Subtopic	125
Deleting a Topic or Subtopic	127
Topics and Filter Modules	128
Viewing Filters on a Topic	128
Attaching a Filter to a Topic	130
Editing Filter Parameters	131
Enabling Global Filter Properties	132
Detaching a Filter From a Topic	132
Working With Events and Event Properties	133
Viewing Events and Event Properties	133
Sending Test Events and Deleting Events	135
Sending a Test Event	135
Deleting an Event	137
Deleting All Events in a Topic	137
Editing, Adding, and Deleting Event Properties	138
Editing and Deleting Event Properties	138
Adding Event Properties	139
Managing Routes and Route Properties	140
Working with Routes	140
Viewing a Topic's Routes	140
Creating a New Route for a Topic	142
Deleting All Routes from a Topic	143

Deleting a Single Route from a Topic	144
Working with Route Properties	144
Viewing Route Properties	145
Editing, Adding, or Deleting a Route Property	145
Attaching a Filter to a Route	146
Detaching a Filter from a Route	147
Chapter 6 Selecting a Permissions System	149
Selecting a Permissions Module	150
Configuring Flat Files Permissions	153
Flat Files Parameters	154
Audit	154
Audit Forbidden	154
Authentication policy	154
Debug	157
Default URL on session login	157
Open Access URLs	157
Template for login form	158
Chapter 7 Implementing Security with LDAP	159
LiveServer LDAP Overview	160
Integrating ESS With Your LDAP Environment	161
Prerequisites for LDAP Installation	161
Perl Environment Requirements	162
Setting Up the Perl Environment	163
Background Information on LDAP Integration Tasks	164
Create an ESS Container and ESS Admin User	164
Load LiveServer Schema Extensions	164
eDirectory Example: Perl-based Schema Load from RFC2252-compliant Files	
166	
Create the LiveServer Users and Groups	167
Load Permission Patterns	168
Configure LiveServer for LDAP	168
Reduce the LSadmin User's Privileges (In AD)	169
Supporting Files	169
Deploying ESS in a Dual LDAP Environment: AD With OpenLDAP	170
Task 1: Install and Validate the OpenLDAP Server	170
Task 2: Create an ESS Container and ESS Admin User in OpenLDAP	171
Task 3: Load LiveServer Schema Extensions in OpenLDAP	172

- Task 4: Load Permission Patterns in OpenLDAP 172
- Task 5: Load the LiveServer Users and Groups in AD 173
- Task 6: Configure the LiveServer’s LDAP Module 174
- Task 7: Validate Your Configuration 177
- Deploying ESS in a Dual LDAP Environment: AD With ADAM 179
 - Task 1: Install ADAM with the Partition Name O=ess,C=knadam 179
 - Task 2: Create an ESS Container and ESS Admin User in ADAM 180
 - Task 3: Load LiveServer Schema Extensions in ADAM 181
 - Task 4: Load the LiveServer Users and Groups in AD 181
 - Task 5: Load the Permission Patterns in ADAM 183
 - Task 6: Configure LiveServer’s LDAP Module 184
 - Task 7: Validate Configuration 186
- Deploying ESS in a Dual OpenLDAP Environment 188
 - Task 1: Install and Validate the OpenLDAP Servers 188
 - Task 2: Create an ESS Container and ESS Admin User in the OpenLDAP Authorization Server 189
 - Task 3: Load LiveServer Schema Extensions in the OpenLDAP Authorization Server 190
 - Task 4: Load Permission Patterns in the OpenLDAP Authorization Server 190
 - Task 5: Load the LiveServer Users and Groups in the OpenLDAP Authentication Server 191
 - Task 6: Configure the LiveServer’s LDAP Module 192
 - Task 7: Validate Your Configuration 196
- Deploying ESS in a Standalone OpenLDAP Environment 197
 - Task 1: Install and Validate the OpenLDAP Server 197
 - Task 2: Create an ESS Container and ESS Admin User 198
 - Task 3: Load LiveServer Schema Extensions in the OpenLDAP Authorization Server 199
 - Task 4: Load the LiveServer Users and Groups 199
 - Task 5: Load the Permission Patterns 200
 - Task 6: Configure LiveServer’s LDAP Module 201
 - Task 7: Validate Configuration 204
- Deploying ESS in a Standalone AD Environment 205
 - Task 1: Create an ESS Container and ESS Admin User 205
 - Task 2: Load LiveServer Schema Extensions 206
 - Task 3: Load the LiveServer Users and Groups 207
 - Task 4: Load the Permission Patterns 210
 - Task 5: Configure LiveServer’s LDAP Module 211

Task 6: Validate Configuration	214
Task 7: Reduce the LSadmin User's Privileges.....	214
Reducing the LSadmin User's Privileges	215
Remove the LSadmin User From the Domain Admins	215
Edit the LSadmin User's Permissions	215
Verify the LSadmin User's Rights	221
Selecting and Configuring an LDAP Module	223
LDAP and LDAP Test Parameters	225
Audit.....	226
Authentication policy.....	226
Debug	226
Default URL on session login.....	226
Directory Type	226
LDAP Administration DN.....	226
LDAP Administration Password	227
LiveServer Authoring DN	227
LiveServer Authoring Password	227
LiveServer DN	227
LiveServer Password.....	227
Open Access URLs	228
Root Distinguished Name.....	228
Template for login form	228
Advanced LDAP Configuration	229
Advanced LDAP Parameters	230
Bind timeout	230
Encode passwords in knsettings.conf	230
Follow referrals during search.....	230
Group container DN	230
Group member attribute	230
Group member Of attribute.....	230
Group objectClass	231
LDAP authentication method	231
List subgroups of a user or a group in LDAP queries	231
LiveServer Username	231
Permission change poll interval.....	231
Realm	231
Trust Login identity.....	232
User container DN	232

User identifier attribute	232
User objectClass	232
Chapter 8 Managing Permissions	233
Creating Access Patterns	234
The Access Pattern Table	236
The LiveServer's Default Access Patterns	237
Syntax in Access Patterns	238
Determining Which Access Pattern Applies	240
Viewing and Editing Access Patterns	241
Creating New Access Patterns	244
Naming Access Patterns	244
Creating a New Access Pattern From Scratch	244
Creating a New Access Pattern By Cloning	245
Deleting Access Patterns	247
Chapter 9 Managing Users and Groups	249
Introducing the Users/Groups Tab	250
Managing Users	252
The LiveServer's Default Users and Groups	252
LDAP User Types	254
Adding, Finding, and Deleting Users	255
Configuring Internal Users	257
Internal System Users Parameters	258
Daemon Password	258
Daemon Protocol	259
Daemon User	259
Managing Groups	260
Chapter 10 Implementing Clusters	263
Introducing LiveServer Clusters	264
Cluster Module Information	264
How Clustering Works	265
Failover Mechanism	266
Flow Control and System Recovery	266
Scalable Connections	266
Load Balancer Configuration	267
Important Cluster Considerations	267
Cluster Creation Overview	268

Cluster Creation Checklist	269
Clusters and Security	269
Enabling and Configuring a Multicast Cluster	271
Initial Multicast Configuration: Creating the First Node	271
Multicast Parameters	274
Cluster port	274
Database snapshot timeout	274
Debug	275
File transfer wait	275
Liveness wait	275
Multicast address	275
Multicast port	276
Ordering level	276
Partition-safe	277
Recovery script URL	277
Adding Second and Additional Nodes to a Cluster	278
Editing Nodes in an Existing Cluster	280
Deleting Nodes in an Existing Cluster	281
LiveAdapters Clusters	282
Creating LiveAdapters Clusters on Windows 2000 or Older and Linux	282
Windows 2000 and Older Shared Directory Notes	283
Linux Shared Directory Notes	284
Creating LiveAdapters Clusters on Windows 2003	284
Chapter 11 Performance Monitoring and Process Management	287
Performance Measurement and Tuning	288
Monitoring Activity	290
Configuring and Using the Process Manager	293
Configuring Processes	294
Monitoring Processes	298
Configuring and Viewing Logs	301
Configuring Log Options	301
Log Configuration Parameters	302
Clear alarms after	303
Crash if a performance measurement is out of range	303
Debug	303
Enable command processing time measurement	303
Enable Leak Check	303

Enable Stack Check	303
Enable thread stack size checking	303
Log all request headers	303
Log debug topic locks?.....	303
Log memory statistics?.....	304
Log request headers if forbidden	304
Maintain Access Logs	304
Max Backup	304
Maximum response time allowed for any command processing	304
Performance range for background operations	304
Performance range for fast operations.....	304
Post logs and alarms as events	305
Roll Log Time	305
Roll Logs	305
Server log file (base) name	305
Trace configuration subsystem	305
Trace event expiration and periodic event subsystem (reaper).....	305
Trace LiveServer TCL Interface.....	305
Trace network (post & HTTP request) subsystem	305
Trace statistics-handling	306
Trace TCL scripts (configuration and permissions).....	306
Trace tunnel (delivery) subsystem	306
Viewing Log Files	306
Configuring and Monitoring Alarms	308
Configuring Alarms	308
Viewing Alarms	310
Configuring Memory	313
Memory Configuration Parameters	314
Crash dump file location.....	314
Crash dump file policy	314
Keep detailed XML reaper statistics	314
Memory Cache Topic Period	314
Memory for Tunnel Buffers.....	314
Memory Notice Threshold	315
Reaper Interval	315
Reaper Interval Minimum.....	315
Stack Size	315
Temporary Topic Expires.....	315

Temporary Topic Recognize	316
Tunnel Error Tolerance	316
Chapter 12 Backup, Restore, and Other Utilities.....	317
The LiveServer Utilities.....	318
Accessing the Event Publisher	320
Backing Up Your Database.....	321
Restoring Your Database.....	322
Moving Recovery Information.....	324
Troubleshooting.....	325
Verifying the LiveServer Service Under Windows.....	325
Glossary	327
Index	335



Preface

The *KnowNow LiveServer Administration Guide* is for system administrators and developers who are installing and configuring KnowNow's™ LiveServer. It provides information on installing, configuring, and troubleshooting LiveServer. It also includes information on using the KnowNow utilities. This guide contains a glossary of important KnowNow terms and concepts, and is thoroughly indexed.

This guide is written on the assumption that the reader possesses a general knowledge of HTTP and of Web server administration activities, including security procedures.

What This Guide Covers

This guide describes how to install KnowNow's LiveServer, describes LiveServer terminology and concepts, explains LiveServer configuration, and gives information on the KnowNow utilities. Many chapters describe how to use specific aspects of the KnowNow System Administration console, which is a graphical user interface for configuring and managing topics, routes, permissions, users, and much more.

The chapters in this book are as follows:

- [Chapter 1 , "Introducing KnowNow's LiveServer"](#): A brief overview of the KnowNow architecture, terminology, and concepts.
- [Chapter 3 , "Introducing the KnowNow System Administration Console"](#): Introduces the KnowNow System Administration console and describes how to use it.
- [Chapter 2 , "Installing KnowNow's LiveServer"](#): Provides installation and start-up/shutdown instructions.
- [Chapter 4 , "Configuring LiveServer"](#): Provides information on how to configure LiveServer using the KnowNow System Administration console.

- [Chapter 5 , "Managing Topics"](#): Provides information for understanding and managing topic space and for using the KnowNow System Administration console to manage topics.
- [Chapter 6 , "Selecting a Permissions System"](#): Describes the different permissions modules and how to select a permissions module for your system. Includes information on configuring the Flat File and 2.0 Compatible permissions modules.
- [Chapter 7 , "Implementing Security with LDAP"](#): Describes how to integrate LiveServer with LDAP systems. Also describes how to configure the LDAP Test and LDAP permissions modules.
- [Chapter 8 , "Managing Permissions"](#): Describes the default permissions patterns and tells how to customize them or create your own permissions patterns.
- [Chapter 9 , "Managing Users and Groups"](#): Describes how to manage users and groups using the KnowNow System Administration console.
- [Chapter 10 , "Implementing Clusters"](#): Provides information on configuring and implementing LiveServer clusters using the KnowNow System Administration console. Also has information on clusters when using KnowNow LiveAdapters.
- [Chapter 11 , "Performance Monitoring and Process Management"](#): Describes how to use the KnowNow System Administration console to view performance statistics and to configure and monitor processes.
- [Chapter 12 , "Backup, Restore, and Other Utilities"](#): Introduces the KnowNow utilities and provides information on backing up and restoring LiveServer databases, as well as some troubleshooting information.
- [Glossary](#): Provides a list of KnowNow and other terms and their definitions.
- An index provides quick access to the contents of this book.

Additional Sources of Information

Additional information can be found in the following sources. The documentation can be accessed as described under ["Accessing LiveServer Documentation and Help"](#) on [page 52](#).

- [KnowNow LiveServer Developer's Guide](#), which provides information for using LiveServer and the Connectors, including information on customizing the Connectors and writing new Connectors using the included APIs. This book is in PDF and help file formats.
- Documentation for the LiveServer APIs is provided in HTML format.

- *LiveServer_release_notes.html* contains a list of supported platforms, new features, known issues, and late-breaking information.
- The *KnowNow LiveBrowser Developer's Guide* provides information on using the LiveBrowser and its components.
- The KnowNow Web site, <http://www.knownow.com>, provides additional documentation and support.

Document Conventions

<i>Italic</i>	Italics are used for emphasis, for the titles of books, for file names, and for variables.
Boldface type	Boldface type is used for emphasis and for the names of menu, button, and other controls/commands.
Code	Code excerpts and command line sequences are shown in this type face.
Ellipsis . . .	An ellipsis is used in code examples and syntax to indicate that irrelevant information is not shown.
<i>/name/</i> and <i>/name</i>	<p>A name enclosed in slashes is a directory name. It does not necessarily mean that the directory is at the root level; in fact, in most cases, the directory is a subdirectory in a hierarchy.</p> <p>A name starting with a slash represents a topic name; as with directories, the topic could be anywhere in a topic hierarchy.</p>

How To Contact KnowNow, Inc.

We welcome any comments you may have about our software products or our documentation. Our goal is to provide functional and easy-to-use products that help you work more efficiently. For more information, contact us at

KnowNow, Inc.
997 East Arques Avenue
Sunnyvale, California 94085 USA

Phone: (408) 585-1800

Fax: (408) 585-1801

Web site: <http://www.knownow.com>

For technical support, or if you have comments or suggestions, contact Support at

support@knownow.com

(408) 585-1835

For sales inquiries, contact sales@knownow.com or call (408) 585-1866.

Chapter 1 Introducing KnowNow's LiveServer

KnowNow's architecture provides reliable, asynchronous, and loosely coupled communication over the Web. By using URLs for topic names and HTTP as the communication protocol, KnowNow's LiveServer makes it possible for applications that communicate using HTTP to publish events without modification. Topic routing and flexible subscriptions enable dynamic data aggregation and syndication, without knowing all endpoints or user domains. Finally, by riding on top of the existing technologies that most companies already have in place, LiveServer makes integration fast, incremental, and cost-effective.

LiveServer's flexible architecture makes it possible to create integrations that were not economically feasible before. This chapter provides an overview of LiveServer's architecture and how it accomplishes these feats in the following sections:

- ["LiveServer Concepts" on page 2](#)
- ["How LiveServer Can Be Used" on page 11](#)

LiveServer Concepts

KnowNow's LiveServer is a software platform for integrating applications in a simple, incremental, and cost-effective manner. LiveServer combines publish-subscribe event delivery with Internet-style routing to bring reliable, asynchronous communications to applications over the Internet. In this model, applications publish information, in the form of events, and these events are categorized by topics.

Other applications subscribe to these topics and receive the events' information as it occurs. For example, an investment bank might run a publish-subscribe system with a separate topic for each stock traded on the NASDAQ and NYSE. Information related to WalMart (WMT), such as ticker prices, breaking news, new research reports, and other desired information is published to the /WalMart topic. All traders who subscribe to this topic receive the various pieces of information (in other words, events) as they are published.

In short, KnowNow's LiveServer accepts events as input, then routes them to other destinations. This is very comparable to an IP router: it accepts packets from one network and sends them to a specified destination. An IP router interconnects networks; KnowNow's LiveServer interconnects applications. Applications publish events to LiveServer, and LiveServer copies those events to other applications which have subscribed to them. KnowNow's LiveServer accepts published events via HTTP as well as subscription requests via HTTP.

LiveServer also provides Web services through SOAP, WSDL, and REST support. For more information on those services, see the *KnowNow LiveServer Developer's Guide*.

This chapter complements the introductory chapter in the *KnowNow LiveServer Developer's Guide*, which goes into more technical detail. As an introduction to the information in that book, the essential concepts for understanding LiveServer are presented under the following headings:

- "Architectural Components"
- "LiveServer Terminology" on page 5
- "Events" on page 6
- "Topics" on page 6
- "Routes" on page 9
- "Event Lifetimes and Initial Route Population" on page 10

Architectural Components

The KnowNow architecture comprises a number of components, each connected by HTTP as the native communication protocol:

- “LiveServers” on page 3
- “LiveServer Modules” on page 4
- “Connectors” on page 4
- “LiveAdapters” on page 5

LiveServers

KnowNow’s LiveServer manages topics, events, and subscribers. Users and applications publish events to LiveServer, and LiveServer then deliver these events to subscribers. Unlike traditional publish-subscribe middleware, LiveServer works in a loosely coupled, decentralized manner.

LiveServer supports a flexible method of topic subscriptions. As in any publish-subscribe system, a topic subscriber can be an end user or an application. All users or applications that subscribe to a topic receive events on that topic. However, KnowNow provides additional capabilities that support a great deal more flexibility and power.

With LiveServer, a topic can be a subscriber to another topic. For example, an investment bank might create a “Retailing Index” topic that subscribes to individual company topics such as WalMart, KMart, Target, and others. As a subscriber to each of these, the Retailing Index topic now provides a dynamically aggregated information stream. Users can subscribe to this topic, filter it, and perform other operations on it for on-the-fly data aggregation. Also, because LiveServers can subscribe to each other’s topics, corporations can share information and distribute it to their local users, without sharing these users’ information or requiring a single large namespace.

In many ways, LiveServer looks and acts like a standard HTTP Web server. Multiple LiveServers can be clustered for failover and scalability. Topics are exposed as standard URLs, and LiveServer uses HTTP as its native communication protocol. Publishing and subscribing are performed with HTTP POST and GET. This enables applications that have an HTTP interface to publish events to the LiveServer without having to make any modifications at all; an HTTP POST to the correct topic’s URL is all that is required.

LiveServer Modules

KnowNow LiveServers use modules that perform a variety of functions, such as data transformation, content filtering, authentication, and authorization. In addition, you can write your own modules to perform operations such as filtering and security, or write your own custom recovery modules. Information on using the LiveServer's API to write your own modules is provided in the *KnowNow LiveServer Developer's Guide*. Loading modules into LiveServer and enabling their use is an easy task; you simply use the KnowNow System Administration console as described in [Chapter 4, "Configuring LiveServer."](#)

With the capability of adding modules, LiveServers become extensible platforms for real-time data interchange among decentralized applications.

Connectors

Connectors are optional, client-side components that manage persistent, bidirectional connections between an application and a LiveServer. They are non-intrusive and lightweight; the JavaScript Connector works with "zero-install" and is automatically downloaded into the browser without local installation or modification. By holding open a connection and periodically sending small "heartbeat" packets, the Connector delivers bidirectional communication over HTTP.

Beyond the persistent connection, Connectors also perform event delivery functions such as auto-checkpointing and offline queuing. Auto-checkpointing ensures that events are delivered and received reliably, and that duplicates are automatically removed. Offline queuing enables publishers and subscribers to continue working when network connections are not available. If an application is publishing events and a connection is not available, the Connector simply queues the events locally, then transparently delivers these events when a connection is re-established.

In fact, the Connector is optional. Some applications do not require a persistent connection; they simply want to publish events. Since a publish operation to a LiveServer happens via a HTTP POST, applications can communicate with LiveServers without Connector, and without any modifications. Conversely, LiveServer can publish to an external URL, so an application can subscribe without a Connector. In doing so, the application does not receive persistent connections, offline queuing, or checkpointing; and in some cases this is an acceptable trade-off.

Connectors are available for a variety of platforms and languages, including JavaScript, Java, .NET, .NET Compact Framework, C++, and ActiveX/COM. For more information about the Connectors, see the *KnowNow LiveServer Developer's Guide*.

LiveAdapters

LiveAdapters are special kinds of modules that make it possible to connect to different kinds of services. Some LiveAdapters are internal; they work beneath the surface to provide additional functionality to other KnowNow products, such as KnowNow's Enterprise Syndication Solution (KnowNow ESS). Others have graphical user interfaces. For example, the LiveAdapter for RDBMS is used to create database business methods that are transmitted as events and therefore can be shared in a number of different ways. The LiveAdapter for Microsoft Exchange, which is included with KnowNow ESS, makes it possible to create channels that exchange data between ESS and Microsoft Exchange. Each LiveAdapter is documented where appropriate.

LiveServer Terminology

[Table 1-1](#) lists some of the most important terms relating to LiveServer, all of which are discussed in more detail in this and other KnowNow books. For more definitions, see the [Glossary on page 327](#)

Table 1-1. Important LiveServer terms.

Term	Description
Events	Messages sent by an application or person specifically to trigger actions.
Topics	A collection of events to which KnowNow clients can subscribe or publish events.
Routes	The path an event travels during the publish/subscribe process.
Publish, publication	The act of sending an event from a KnowNow LiveServer.
Subscribe, subscription	The act of receiving an event from a KnowNow LiveServer.
Tunnels	Tunnel routes are smart routes from the LiveServer to clients using persistent connections.

Events

An event is a basic unit of information that LiveServer knows how to deal with. KnowNow events are made up of header/value pairs, with the header and the value separated by a colon; for example, "price:%9.03." Applications can use any header/value pairs that make sense for their application. In addition, KnowNow provides a rich set of headers to accomplish a number of tasks as described in the *KnowNow LiveServer Developer's Guide* (which book also has a great deal more information on events).

A typical event might contain the following set of headers and values:

```
kn_id: 141059u9103
kn_time_t: 996172943
price: 9.03
qty: 1000
symbol: SUNW
```

This event contains three headers defined by the application (price, qty, and symbol) and two kn_headers that have special meaning to LiveServer:

- `kn_id` holds a unique identifier for this event. Applications can generate their own IDs (by including a `kn_id` header and value when the event is published to LiveServer) or can rely on LiveServer to generate them. If an event is published to a LiveServer without a `kn_id` field, that LiveServer will automatically generate and apply a unique `kn_id` header.
- `kn_time_t` contains the time (measured in seconds since January 1, 1970) that this event was first published. LiveServer applies this header to any event it receives which does not already contain a `kn_time_t` field. In the example above, the event was first published on Thursday, July 26, 2001 at 11:42:23am.

The headers `price`, `qty`, and `symbol` are application-defined headers which have meaning to the publisher. Note that LiveServer does not place any restrictions on these headers; non-ASCII data need only be encoded in UTF-8 format before transmission into LiveServer.

Topics

Every time an event is published to a LiveServer, it must be published to a topic. A topic is basically the address on LiveServer to which an event gets published. Subscriptions to that topic are notified with every change.

Every topic on a LiveServer is fully addressable as a URI. For example, the topic `/stock-Quotes` on a LiveServer running on `example.com` can be addressed as:

```
http://example.com/kn/stockQuotes
```

Topic naming supports [the full UTF-8 specification \(RFC2279\)](#). A topic name can contain Latin letters, digits, slashes, underscores, dashes, periods, and non-ASCII characters.

You can create a topic just by referencing the topic. Events are sent to a topic by a publisher, after which the topic forwards that event along all of its routes to all the subscribers to that topic.

LiveServer manages a hierarchy of topics much like a standard Web server manages a hierarchy of URLs. For example, an application might use `/stockQuotes/nasdaq` to publish a feed of Nasdaq price updates, and `/stockQuotes/nyse` for publishing NYSE updates. Note that if an event is published to `/stockQuotes`, only subscribers to that topic will be sent a copy of the event; subscribers to `/stockQuotes/nyse` will not receive the event. The topic hierarchy is initially empty when you start up a new LiveServer, and topics get created on the fly as events are published into them.

Metadata Subtopics

LiveServer maintains metadata subtopics for every topic in its hierarchy. These are called `/kn_subtopics` and `/kn_routes`. The following example illustrates what these topics do and how they can be used within a topic hierarchy:

```
/stockQuotes
  /stockQuotes/kn_subtopics
  /stockQuotes/kn_routes
  /stockQuotes/nyse
    /stockQuotes/nyse/kn_routes
    /stockQuotes/nyse/kn_subtopics
  /stockQuotes/nasdaq
    /stockQuotes/nasdaq/kn_subtopics
    /stockQuotes/nasdaq/kn_routes
```

`/kn_subtopics` contains an event for each of the subtopics of the given topic; in other words, it contains a list of all the subtopics of that topic. In the preceding example, `/stockQuotes/kn_subtopics` contains an event for `/stockQuotes/nyse`, an event for `/stockQuotes/nasdaq`, and an event for `/stockQuotes/kn_routes`. LiveServer will publish an event into the `/kn_subtopics` topic whenever a subtopic gets created; thus if the subtopic `/stockQuotes/ftse` were created, a new event would be published to `/stockQuotes/kn_subtopics` referring to the new subtopic. Note that `/kn_subtopics` topics can be subscribed to just like any other topic, making it possible for an application to find out when new topics are created as well as to enumerate the existing topic hierarchy. However, only LiveServer may publish directly to the `/kn_subtopics` topic.

Similarly, `/kn_routes` contains an event for each of the routes out of a given topic; in other words, it contains a list of all the active routes to that topic. For instance, if a route is created from `/stockQuotes/nasdaq` to a new topic called `/stockQuotes/all`, LiveServer will publish an event to `/stockQuotes/nasdaq/kn_routes` with the full URI of the destination topic (in this case, `http://example.com/kn/stockQuotes/all`). Just as with `/kn_subtopics`, it is possible to subscribe to `/kn_routes` topics; applications can then receive notification whenever a new route is created out of a given topic. Only LiveServer may publish to a `/kn_routes` topic.

Journal Topics

A journal topic, or journal, is any topic whose final path element is `/kn_journal`. For example: `/who/jenkins/kn_journal` would be a valid journal topic; while `/who/jenkinskn_journal` would not be.

Journal topics are special in that:

- they do not have subtopics (no `/kn_subtopics`)
- although they can have routes out of them (`/kn_routes` subtopics), the routes are constrained to be tunnels

Journal topics can only be used as the target of a route from another topic (for example, a route from `/stockQuotes/nasdaq` to `/who/steve/kn_journal`). Journal topics are the endpoints to which Connectors can connect to create tunnels, which are the persistent TCP connections to LiveServer.

When an application initializes a Connector, the Connector connects to LiveServer (using an HTTP GET request) and creates a tunnel from a random journal topic name. Some Connectors use a naming convention for their journal topics such that they all begin with `/who`, but this is not required by LiveServer. As long as the journal topic ends with `/kn_journal`, LiveServer will create the journal and associated tunnel.

Later, when the application asks the Connector to subscribe to a given topic (for example, subscribe to `/stockQuotes/nyse`), the Connector actually creates a route out of `/stockQuotes/nyse` to the journal topic associated with its connection. When an event is later published to `/stockQuotes/nyse`, it is forwarded along the route to the user's journal, and from there sent down the tunnel to the Connector.

You can have more than one tunnel out of a single journal, which makes it possible for more than one Connector to connect to the same stream of events. In the case where a single journal topic has more than one tunnel associated with it, any event delivered to the journal topic will be sent down all of the connected tunnels.

LiveServer supports two kinds of tunnels: simple tunnels and JavaScript (js) tunnels. Connectors select the kind of tunnel to create when they connect to a LiveServer. With the exception of the JavaScript Connector, all Connectors use the simple tunnel format to communicate with a LiveServer. For more information on these tunnel formats, see the *KnowNow LiveServer Developer's Guide*.

Routes

One of the most powerful things the KnowNow LiveServer allows you to do is set up routes out of a topic. When an event is published into a topic, LiveServer automatically forwards that event along all the routes out of the topic. Routes can be made to any URI (HTTP and HTTPS protocols are supported). This means, for instance, that you could create a route from the topic `/stockQuotes/nyse` to a CGI script or servlet on another machine (for example, a route to `http://example.com/servlet/nyseListener`).

In this example, whenever an event is published into `/stockQuotes/nyse`, LiveServer makes an HTTP POST request to the URI referenced in the route, sending the event as a part of the request. Since the destination of a route can be any URI, routes can be made between topics on the same KnowNow LiveServer, between topics on different LiveServers, or from a LiveServer to any application that can receive an HTTP POST request. In addition, you can create an unlimited number of routes out of a given topic, making fan-out of events easy to accomplish.

Subscriptions and Routes

The KnowNow LiveServer does not distinguish between a subscriber to a topic and a route out of that topic. As LiveServer only routes to URIs, what is referred to as a *subscription* in another publish and subscribe system is just a route out of a topic in a KnowNow LiveServer. At a high level, what LiveServer does when it receives an event to publish is as follows:

1. Determines the topic to which the event is to be routed.
2. Creates that topic if it does not already exist (and adds a `/kn_subtopics` entry to the new topic's parent topic).
3. Delivers the event into the topic.
4. For each entry in the topic's `/kn_routes` subtopic, sends the event to the URI specified in the route. LiveServer supports HTTP and HTTPS protocol URIs; LiveServer sends the event data as an HTTP POST to the specified location.

KnowNow Connectors allow applications to maintain a persistent connection to LiveServer. Therefore, when an application subscribes to (creates a route out of) a particular topic, any events published to that topic are sent to the application over a live TCP connection, instead of LiveServer making an HTTP POST to the application. Connectors accomplish this using a third type of special topic known as a journal topic.

Event Lifetimes and Initial Route Population

When an event is delivered to a topic, it is stored inside the topic and then forwarded along any routes out of the topic. Unlike other publish and subscribe systems, LiveServer allows events to live beyond the time of their initial publishing. Events can include a `kn_expires` header that sets the time the event will expire. Events published to LiveServer that do not include this header are subject to a (configurable) default expiration time. (The parameter that controls this value is **Default Expiration Time**.)

When a route is created out of a topic, the route creation request can include parameters for initial route population (IRP), which requests that LiveServer immediately forward existing non-expired events along the route.

Using the `kn_history_` headers, route requests can ask LiveServer to deliver the last n events or all the events seen within the previous n seconds, as well as ask for other such information. For more on headers, see the *KnowNow LiveServer Developer's Guide*.

How LiveServer Can Be Used

To provide some ideas of a few of the many ways in which LiveServer and its capabilities can be used, this section describes two LiveServer usage scenarios under the following headings.

- “Basic Use”
- “Aggregation” on page 12

Basic Use

The most basic use of LiveServer is as follows:

- A client publishes events to a topic, which resides on a KnowNow LiveServer.
- The topic then immediately passes the events to all the subscribers that are subscribed to that topic.

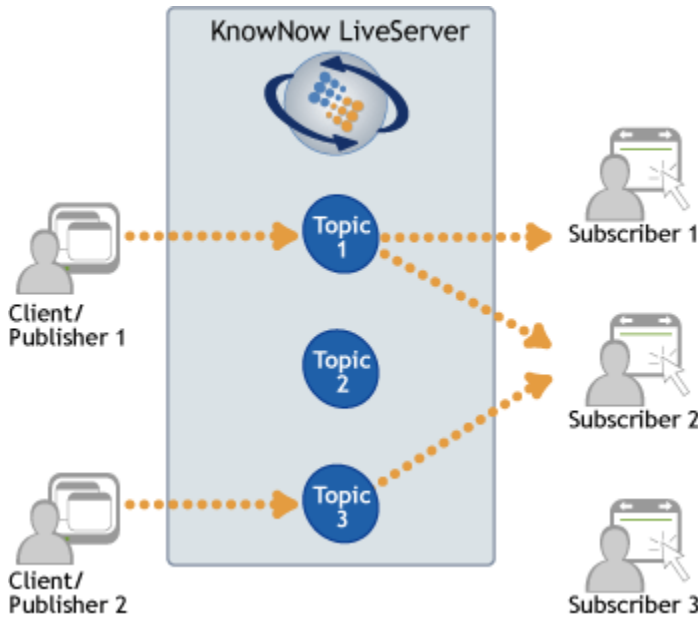
Subscribers can also subscribe to multiple topics.

In the following example:

- Two subscribers (Subscriber 1 and 2) have subscribed to Topic 1.
- Subscriber 2 has also subscribed to Topic 3.
- When the client/publisher publishes an event to Topic 1, both subscribers will be notified.

- If the client/publisher were to publish to Topic 3, only Subscriber 2 would be notified.

Figure 1-1. Using a KnowNow LiveServer —basic scenario.



Aggregation

Topics within a KnowNow LiveServer can also be subscribed to one another. In this way, events transmitted on a set of topics may feed into one master topic. In addition, events can be transmitted to topics on another KnowNow LiveServer via aggregation.

In the scenario shown in the figures on the following page,

- Events published to Topic 1 are delivered to Subscriber 1, as well as Topic 2. Topic 2 then delivers the events to its subscribers (Subscriber 2).
- Events published directly to Topic 2 are sent only to Subscriber 2.
- Events published to Topic 3 are delivered to Subscriber 2 as well as to Topic 1 on the other LiveServer. Topic 1 on the other LiveServer then delivers the events to Subscriber 3. See [Figure 1-2 on page 13](#).

Topics and routes are managed by applications which are written to work with LiveServer through the use of KnowNow Connectors. More information about these Connectors can be found at <http://support.knownow.com>.

Figure 1-2. Using a KnowNow LiveServer—aggregation scenario.

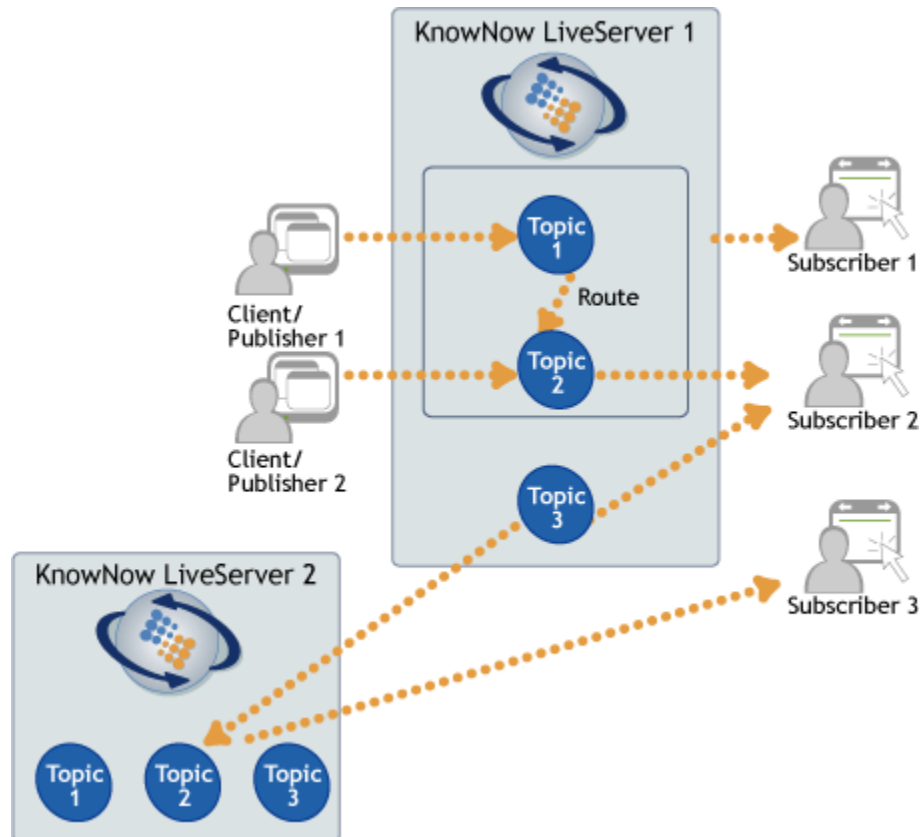
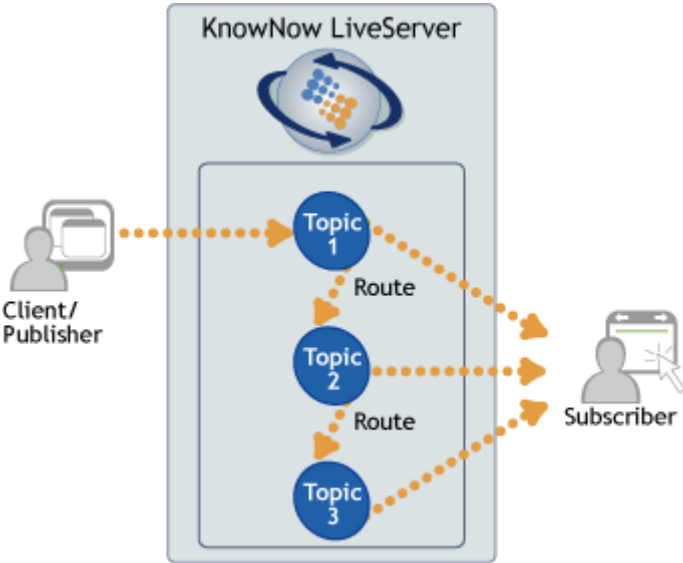


Figure 1-3. Special case—subscribing to two topics along a route.



If you subscribe to two or more topics along a route, you get the same event more than once.

Chapter 2 Installing KnowNow's LiveServer

This chapter provides instructions for installing and starting up LiveServer under the following headings:

- "System Requirements and Supported Platforms" on page 16
- "Upgrading a Previous Installation on Any Platform" on page 18
- "Installing and Uninstalling KnowNow Products" on page 19
- "Installing LiveServer on Windows" on page 22
- "Installing LiveServer on Linux" on page 35
- "Verifying that LiveServer is Running" on page 40

System Requirements and Supported Platforms

LiveServer has been tested on the following platforms:

- Microsoft® Windows Server 2003
- Microsoft® Windows 2003 Enterprise Edition SP1 (Japanese)
- Red Hat Enterprise Linux AS 3
- Red Hat Linux Enterprise 4.0 (64-bit Linux) update 3 or better on an Intel Xeon or AMD Opteron processor platform; with this configuration, topic space is greatly increased over 32-bit LiveServers
- SUSE™ LINUX Enterprise Server 9 and 10



Note: Regardless of platform, to be able to access the KnowNow System Administration console, you must have Internet Explorer 5.5 or later.

The minimum and recommended system requirements for a LiveServer host machine are provided in [Table 2-1](#). As with any complex system, your mileage may vary.

Table 2-1. Minimum and recommended system requirements for LiveServer.

Item		Minimum Setting	Recommended Setting
Free disk space (to allow for installation, working space, log files, and the like)	With or without recovery	500 MB	3 GB
	Without recovery	128 MB	512 MB
Memory	Without recovery	128 MB	512 MB For a cluster, we recommend using this formula: (2*(RAM size)) + 256 MB working space
	With recovery	512 MB	1 Gigabyte
Processors	Type of processor	PIII 500 MHz	P4 Xeon 2.4 GHz
	Number of processors	1	2

LiveBrowser System Requirements

LiveBrowser is installed automatically when you install LiveServer. If you wish, you can install LiveBrowser separately on any kind of Web server. The minimum system requirements for LiveBrowser are as follows.

- LiveBrowser requires an installed LiveServer, either on the same machine or on a different machine.
- LiveBrowser applications run on any platform supporting the following browsers:
 - Microsoft Internet Explorer 5.5, Internet Explorer 6.0, or Internet Explorer 6.0 with Service Pack 2 on Windows XP
 - Mozilla 1.6, Mozilla Firefox 1.0, and Mozilla Firefox 1.5 on Windows
 - Netscape® Navigator 7.0 and 7.1

Required Linux Privileges

At run time, the minimum write privileges required by LiveServer on Linux are:

- Write privileges to the LiveServer's /knrouter_db/ subdirectory and all files within it.
- Write privileges to the LiveServer's /log/ subdirectory and all files within it.

For installation, write privileges are also needed for the installation directory (normally, /opt/knserver/).

Upgrading a Previous Installation on Any Platform

If you have a previous installation of KnowNow's LiveServer, perform the following tasks to upgrade to a newer version.



Note: Upgrading a clustered installation is a complicated procedure. Please contact KnowNow support for help.

1. Although the installation program detects prior installations and backs up your files, it is a good idea to make a backup copy of your configuration files to save your custom configuration settings. The configuration files are located in the `/conf/` directory. It is recommended that you back up the entire `/conf/` directory.
2. During a LiveServer upgrade or reinstallation, the following files are overwritten. If you have modified these files, make a backup before upgrading or reinstalling LiveServer.
 - `/LiveServer_install/runtime/pages/index.html`
 - `/LiveServer_install/runtime/pages/livebrowser/js/kn_config.js`
3. After performing any of the above actions (if needed), install LiveServer as described in the following sections:
 - [“Installing LiveServer on Windows” on page 22](#)
 - [“Installing LiveServer on Linux Step-by-Step” on page 38](#)
4. After you install LiveServer, verify that LiveServer is working as described under [“Verifying that LiveServer is Running” on page 40](#).

Installing and Uninstalling KnowNow Products

When installing and uninstalling LiveServer and other KnowNow products, the following sequences are highly recommended.

- Install LiveServer first before installing other KnowNow products.
- Before installing or uninstalling other KnowNow products, first shut down LiveServer, then restart it after you have installed all the other products. (Though many KnowNow products will shut down LiveServer for you after prompting for approval.)
- Uninstall LiveServer last after uninstalling other KnowNow products.

Somewhat more detailed steps to follow for installing and uninstalling KnowNow products are provided under the following headings:

- [“Installation Order for KnowNow Products”](#)
- [“Uninstall Order for KnowNow Products” on page 20](#)

Installation Order for KnowNow Products

The following steps provide the order in which to install ESS and other KnowNow products, with the assumption that you will be using the ESS Reporting feature and KnowNow’s WebParts. The steps that are required only for specific products are indicated. It is assumed you will already have Microsoft SharePoint MOSS installed. The *KnowNow Enterprise Syndication Solution Administration and Developer’s Guide* has more information on installing and configuring ESS, reporting, and WebParts.

To install ESS with both reporting and WebParts,

- 1. For ESS Reporting:** If you haven’t already done so, install a supported RDBMS. The supported RDBMS platforms are listed with the system requirements in the *KnowNow Enterprise Syndication Solution Administration and Developer’s Guide* and in the *KnowNow LiveAdapters User’s Guide*.
- 2. For ESS Reporting:** In the supported RDBMS, create a database to be used for the ESS Reporting data. Note the database user’s credentials. Ensure that the user has “create table” permissions.
- 3.** Install LiveServer, ESS, and LiveAdapters. If you are using ESS Reporting, use the KnowNow System Administration console to enable authentication policy 2. For step-by-step instructions on installing LiveServer, see other sections in this chapter.

For information on enabling authentication policies, see [Chapter 6](#), "Selecting a Permissions System."

KnowNow Live, SpeedReader, and SpeedWriter are automatically installed with ESS. For detailed information on ESS and these other products, see the *KnowNow Enterprise Syndication Solution Administration and Developer's Guide*.

- 4. For WebParts:** Run KnowNow's WebParts installation program. This installer creates virtual directories for KnowNow, copies code to these directories, and installs KnowNow's WebParts.
- 5. For WebParts:** Using the KnowNow System Administration console, configure LiveServer for cross-domain use by setting the following parameters.
 - a.** Set the Service Network **Domain** parameter to **example.com**.
 - b.** Set the Service Network **Hostname** parameter to **example.com**.
- 6. For WebParts:** Set the SharePoint trust level as described in the *KnowNow Enterprise Syndication Solution Administration and Developer's Guide*.
- 7. For ESS Reporting:** If you didn't do it during the LiveAdapters installation process, copy your database's JDBC jar files to `/LiveServer_install_location/modules/liveadapters/lib/`.
- 8.** Install KnowNow Alerts Desktop.
- 9.** Install any desired KnowNow Connectors.
- 10.** Restart LiveServer.
- 11. For ESS Reporting:** Perform additional configuration steps for reporting as described in the *KnowNow Enterprise Syndication Solution Administration and Developer's Guide*.

Uninstall Order for KnowNow Products

The following steps provide the order in which to uninstall every KnowNow product. The basic principle is to reverse the installation order.

To uninstall every KnowNow product,

- 1.** Shut down LiveServer.
- 2.** If you are uninstalling because you wish to make a clean install of a different version of LiveServer and other KnowNow products, back up any necessary files.

- 3.** Uninstall each Connector.
- 4.** Uninstall KnowNow Alerts Desktop.
- 5.** Uninstall LiveServer.

Installing LiveServer on Windows

You should normally install LiveServer as a service, though if you are performing development work you can install it as a desktop application.



Note: Before you can install LiveServer as a service, make sure you have administrative privileges. Otherwise, LiveServer will be installed as a desktop application, which means it won't start up automatically when your system starts up. You can choose to install LiveServer as a desktop application and later change the installation to a service. For more information on changing the installation, see [“Changing the LiveServer Windows Desktop Installation to a Service”](#) on page 32.

Also, before installing, make sure that the Windows Services window is closed.

To install the KnowNow LiveServer on Windows,

1. Run the LiveServer installation program. If you receive a security warning asking you if you wish to run the software, choose **Run**.

Figure 2-1. Confirm installation.



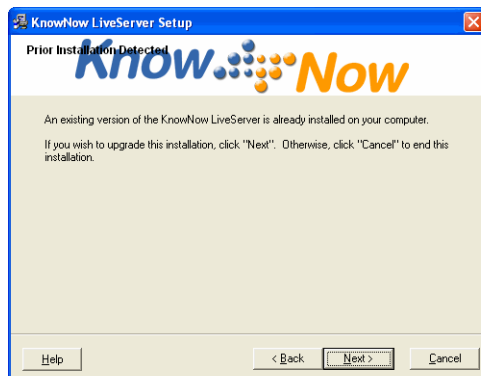
2. The welcome message appears. Choose **Next**.

Figure 2-2. Welcome message.



3. The installation program checks to see if you have a previous, older installation of LiveServer. If you do, the Prior Installation Detected window appears. (If you do not have a previous installation, the License Agreement window appears as shown in [step 4](#).) If you do have a prior, older installation, you can upgrade it by choosing **Next**. If you have a previous installation that is newer than the one you are trying to install, a dialog box is displayed telling you that you need to uninstall that newer version before installing this version.¹

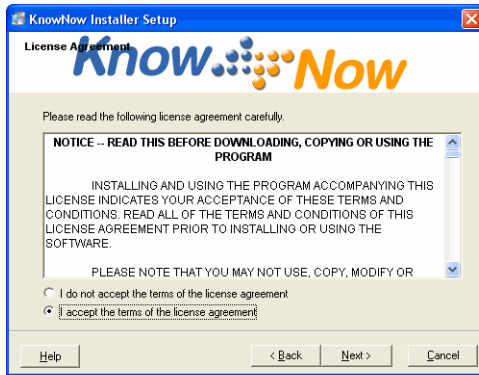
Figure 2-3. Prior installation.



1. Need to update this step the next time I get a new version.

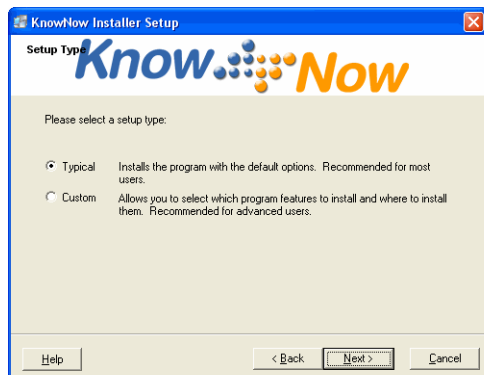
- The License Agreement window is displayed next. Choose **I accept the terms of the license agreement**, then choose **Next**. (**Next** is disabled until you accept the terms.)

Figure 2-4. License agreement.



- Next, the Setup Type window appears, asking whether you want a **Typical** or **Custom** installation. Either installation choice installs LiveServer as a service if it is possible.
 - If you wish to change the default installation directory or make other changes to the default installation, select **Custom**, then choose **Next** and proceed with [step 6 on page 25](#).
 - If you select **Typical**, choose **Next**, then skip to [step 12 on page 28](#).

Figure 2-5. Setup type.



- If you have chosen a custom installation, the next window asks you if you want to install LiveServer as a service. Installing as a service is recommended and is also the only supported production deployment configuration.

Figure 2-6. Install as a service.

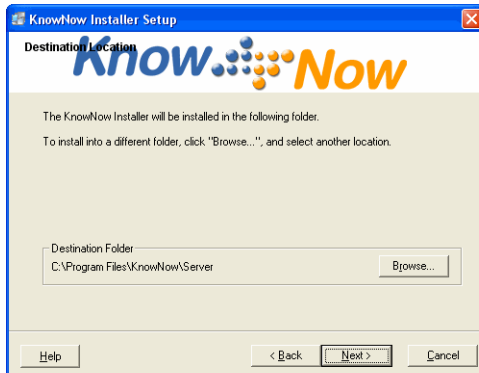


- To install as a service, select **Install as a Windows Service**. If you do **not** want to install it as a service, select **Install as a desktop application**. In either case, make your selection, then choose **Next**.

If you have chosen to install as a service but you are not logged in with administrative privileges, a warning is displayed and LiveServer will be installed as a desktop application. If you wish, you can proceed with the installation and then change the installation type later; for more information, see [“Changing the LiveServer Windows Desktop Installation to a Service”](#) on page 32.

8. Your next choice is to install LiveServer in its default location (C:\Program Files\KnowNow\Server), or install it somewhere else. Either browse to a new destination folder, or leave the default, then choose **Next**.

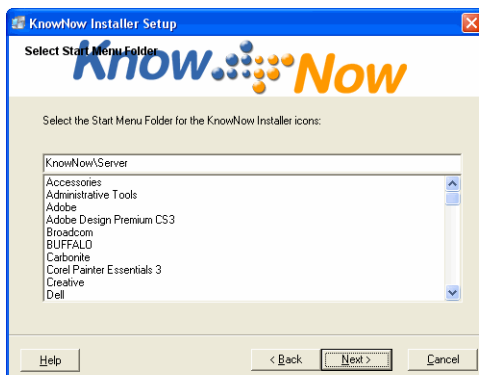
Figure 2-7. Choosing a location for LiveServer.



9. Your next option is to specify which Start menu folder you want the KnowNow components to be available in.
 - If you wish to use a KnowNow Start menu folder, you don't need to select anything.
 - If you wish to use a different Start menu folder, scroll through the list and select the desired folder.

In either case, when you are satisfied with your choice, choose **Next**.

Figure 2-8. LiveServer Start Menu location.

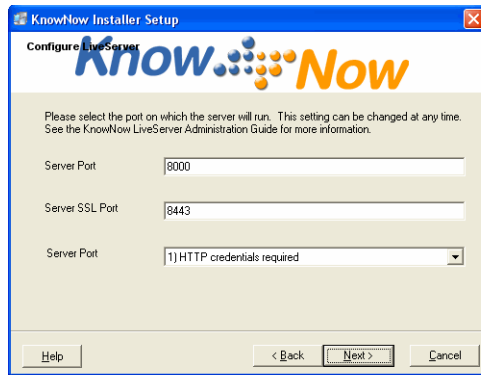


10. Now, you can choose port numbers for your installation, or accept the default port numbers. For a list of default port numbers, see “[HTTP Port](#)” on page 62. You can later change port numbers if you need to by configuring the **HTTP Port** or **HTTPS Port** parameters.

You can also set the default authentication policy in the third field. For more information, see “[Authentication policy](#)” on page 154.

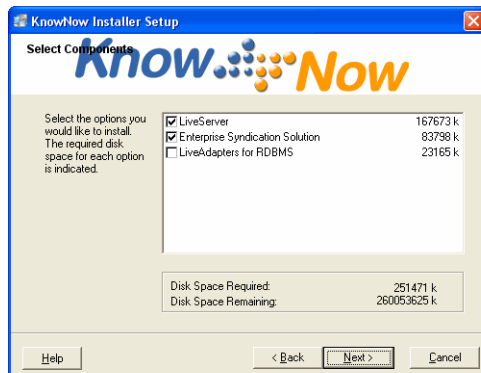
Once everything is as you wish it to be, choose **Next**.

Figure 2-9. LiveServer port numbers and login policy selections.



11. The next window offers you the option to install various components. At a minimum, install LiveServer. To use ESS, KnowNow Live, Web Parts, and so on, also choose Enterprise Syndication Solution. If you wish to use ESS's reporting feature, also choose LiveAdapters for RDBMS.

Figure 2-10. List of programs to install.



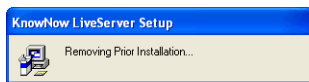
12. The next window announces that you are now ready to install LiveServer. If you are ready to proceed, choose **Next**.

Figure 2-11. Ready to install.



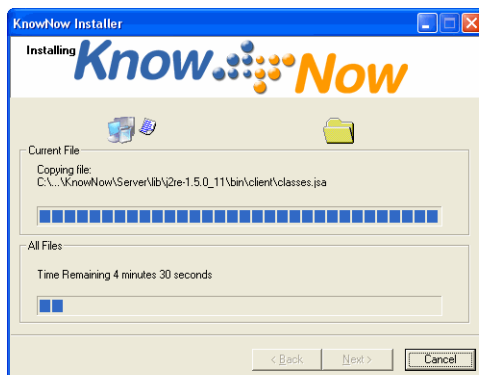
If you have a previous installation, you may see an alert box notifying you that the prior installation is being removed.

Figure 2-12. Removing prior installation.



13. Next, a window displaying the installation progress appears. The installation process takes several minutes.

Figure 2-13. Installation in progress.

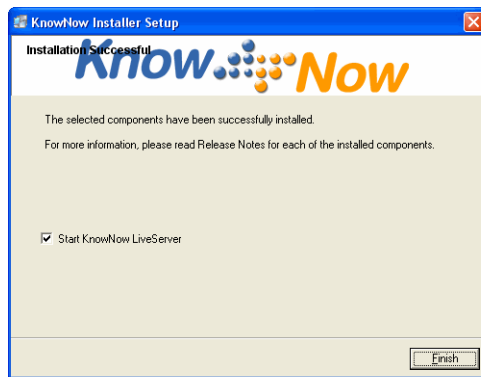


14. As the installation nears completion, if you had a prior installation, a window displays information on where backups of your prior configuration files were stored (the same information as presented under [“Upgrading a Previous Installation on Any Platform”](#) on page 18).²

If there was a problem removing the prior installation, you may see a window asking what you would like to do. If you choose to reboot (with an additional confirmation required), when your system restarts, the LiveServer installation program restarts.

15. Finally, the Installation Successful window appears. Enable or disable the checkbox, then choose **Finish**.

Figure 2-14. Installation successful.



If you chose to start LiveServer, you are now ready to use LiveServer. If you did not choose to start LiveServer, then you can later start it as described under [“Starting LiveServer in Windows”](#) on page 29.



Note: If you have a personal firewall on your computer, you may need to configure it to allow the LiveServer’s processes to run.

Starting LiveServer in Windows

Once you have installed LiveServer and it is running, you can access it by opening a browser window directed at your machine as described under [“Verifying that LiveServer is Running”](#) on page 40.

2. Check this next time.

Normally, LiveServer will start up automatically each time you start up the machine it is installed on. However, if you chose not to start it up at the end of installation, or if at any other time you need to manually start up LiveServer, either as a service or as a desktop application, you can do so using one of the following methods. You can also use these methods to restart LiveServer on failure.



Note: These instructions are for Windows 2003; minor changes may be required for different Windows platforms.

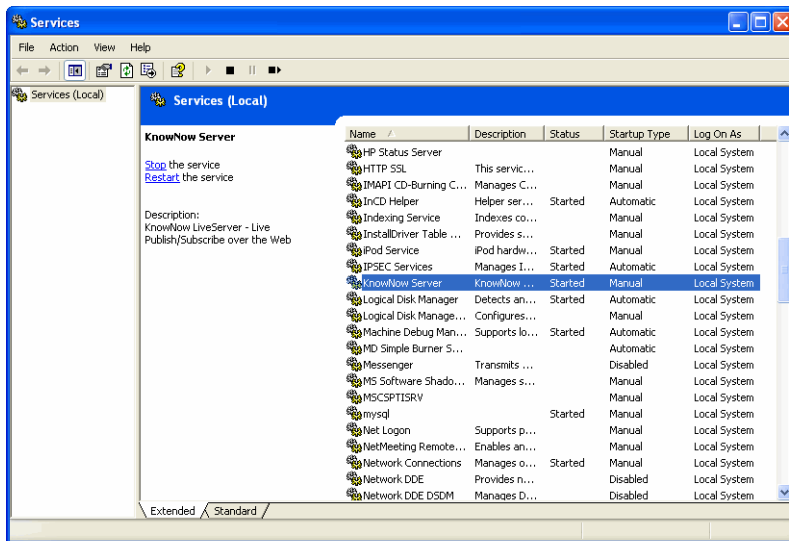
To start the LiveServer service in Windows 2003 (method 1),

1. From the **Start** menu, choose **All Programs > KnowNow > LiveServer > Start KnowNow LiveServer Service**. If you don't have that option in your **Start** menu, use the next procedure.

To start the LiveServer service in Windows 2003 (method 2),

1. From the Windows **Start** menu, select **Control Panel > Administrative Tools > Services**. The Windows Services window appears.

Figure 2-15. Windows services.



2. Select and right-click on **KnowNow Server**. A pop-up menu appears.

3. Choose **Start**. The word **Started** displays in the **Status** column.

To start or stop LiveServer using the command line,

1. Enter the following commands to start or stop LiveServer.

```
net start "KnowNow Server"  
net stop "KnowNow Server"
```

To start the LiveServer desktop application,

1. Choose the **Start LiveServer** icon on your desktop.

Stopping and Restarting LiveServer in Windows

There are three ways to stop LiveServer in Windows, depending on whether you are running LiveServer as a service or as a desktop application. If you wish to remove LiveServer as a service completely (without uninstalling LiveServer), rather than just stop the service for a period of time, see the instructions under ["Removing the LiveServer Service in Windows"](#) on page 33.

If you are running LiveServer as a service, to stop LiveServer (preferred method),

1. In the KnowNow System Administration console, use the **Restart Server** (or **Restart Cluster**) button.

The alternative instructions provided below are given in case this method does not work, or in case you wish to stop LiveServer without restarting it.

If you are running LiveServer as a service, to stop LiveServer (alternative 1),

1. From the **Start** menu, choose **All Programs > KnowNow > LiveServer > Stop KnowNow LiveServer Service**. If you don't have that option in your **Start** menu, use the next procedure.

If you are running LiveServer as a service, to stop LiveServer (alternative 2),

1. Start the **Services** control panel in windows (accessible through the Control Panel's Administrative Tools).
2. Right-click on **KnowNow Server**. A pop-up menu appears.

3. Choose **Stop**. The service is stopped, though it remains in the list of services so you can restart it later if you wish (using the instructions under “Starting LiveServer in Windows” on page 29).

If you are running LiveServer as a desktop application, to stop LiveServer,

1. The preferred method is to use the **Restart Server** (or **Restart Cluster**) button in the KnowNow System Administration console. Since LiveServer is running as an application, it will not restart and instead will shut down gracefully.

If none of these methods succeeds in stopping LiveServer, then, as described below, use the Windows Task Manager to kill the nsd process.



Warning: If you use the following procedure, you risk corrupting the LiveServer database. If this happens, and subsequent attempts to start up the LiveServer fail with a knpersistence-related error, delete the /knrouter_db/ directory and restart LiveServer.

Using the Windows Task Manager to shut down the LiveServer process,

1. Right-click on the task bar and choose **Task Manager**.
2. Select the **Processes** tab.
3. Right-click on the KnowNow LiveServer executable (nsd.exe) and choose **End Process Tree**.

Changing the LiveServer Windows Desktop Installation to a Service

If you did not initially install LiveServer as a service, you can change the desktop application into a service by performing the following steps.

To install LiveServer as a service,

1. Before you begin, log in as Administrator or as a user with Administrator privileges.
2. From the **Start** menu, select **Run**. This opens the Run dialog box.
3. In the **Open** text field, enter **cmd**.
4. Choose **OK**. This opens a command prompt window.
5. At the command line, enter C: and press **Enter**, or enter the letter of the drive on which you installed LiveServer. In this example, drive C is used.

6. Change to the directory where LiveServer is installed. For example,

```
cd "\Program Files\KnowNow\Server"
```

7. Enter the following command. (For a complete list of the command-line options, see “Configuring the Service Network” on page 58.)

```
bin\nsd.exe -I -t conf\liveserver.tcl
```

8. After the service is installed, enter the word **exit** to close the command prompt window.

Removing the LiveServer Service in Windows

These are instructions for removing LiveServer as a service from your Windows system; for example, if you wish to convert it to a desktop application. Note that it does **not** uninstall LiveServer. To uninstall LiveServer entirely, use the Windows **Add/Remove Programs** capability, available through the Control Panel.

If you only wish to stop LiveServer for a time rather than remove it as a service completely, see the instructions under “Stopping and Restarting LiveServer in Windows” on page 31.

To remove the LiveServer service (method 1),

1. From the **Start** menu, choose **Programs > KnowNow > LiveServer > Uninstall KnowNow LiveServer**. If you don’t have that option in your **Start** menu, use the next procedure.

To remove the LiveServer service (method 2),

1. From the Windows **Start** menu, select **Run**. This opens the Run dialog box.
2. In the **Open** text field, enter the word **cmd**.
3. Choose **OK**. This opens a command prompt window.
4. At the command line in the command window, enter C: and press **Enter**. (Or type the letter of the drive on which you installed LiveServer. In this example, drive C is used.)
5. Change to the directory where LiveServer is installed. For example,


```
cd "\Program Files\KnowNow\Server"
```
6. Enter the following command. (For a complete list of the command-line options, see “Configuring the Service Network” on page 58.)

```
bin\nsd.exe -R -t conf\liveserver.tcl
```

7. After the service is removed, enter the word **exit** to close the command prompt window.

Installing LiveServer on Linux

Linux installation instructions are provided under the following headings:

- “Pre-Installation Tasks”
- “Installing LiveServer on Linux Step-by-Step” on page 38
- “Setting the Number of Client Connections” on page 39

Pre-Installation Tasks

If you wish to install LiveServer on any of the supported versions of Linux, you need to prepare your system by installing certain C/C++ libraries. These libraries, which are described under the following headings, are needed in order for LiveServer to run correctly.

- “Required C/C++ Libraries for Red Hat Enterprise Linux AS” on page 35
- “Additional Required C/C++ Libraries for Red Hat Enterprise Linux” on page 36

Required C/C++ Libraries for Red Hat Enterprise Linux AS

For Red Hat Enterprise Linux, in addition to the information on system requirements contained in the book, LiveServer requires that the minimum versions of the following be installed.

- The kernel version must be at least 2.4.20-28.7. This can be installed from kernel-2.4.20-28.7.i386.rpm.
- The following libraries must be installed (possibly available from <http://updates.redhat.com>; you may need to have an account at that site).
 - glibc-common-2.2.5-44.i386.rpm
 - glibc-2.2.5-44.i386.rpm
 - glibc-devel-2.2.5-44.i386.rpm
- The following libraries must be installed (possibly available from <http://updates.redhat.com>; you may need to have an account at that site).
 - libgcc-3.0.4-1.i386.rpm
 - libstdc++3-3.0.4-1.i386.rpm

If these libraries are already installed on your system, you can skip to the installation instructions under “[Installing LiveServer on Linux Step-by-Step](#)” on page 38. Otherwise, to upgrade these libraries, follow these instructions.

To upgrade these libraries,

1. Fetch the RPMs listed above,
2. Install the RPMs by running the following command when logged in as the root user:

```
rpm -i -U \  
libgcc-3.0.4-1.i386.rpm \  
libstdc++3-3.0.4-1.i386.rpm \  
glibc*-2.2.5-44.i386.rpm
```



Note: If you have previously installed additional glibc RPMs (profiling libraries, debugging libraries, and others.) you may receive an error message of the form “Error: removing these packages would break dependencies:” To complete the upgrade, you will need to fetch some or all of the remaining glibc*-2.2.5.i386.rpm packages from the /redhat/linux/updates/7.3/en/os/i386/ subdirectory and re-run the installation command.

3. Once you have installed these libraries, you can proceed with installing LiveServer as described under “[Installing LiveServer on Linux Step-by-Step](#)” on page 38.

Additional Required C/C++ Libraries for Red Hat Enterprise Linux

For additional support of Red Hat Enterprise Linux, it is required that you have installed at least the following library versions (possibly available from <http://updates.redhat.com>; you may need to have an account at that site).

- libgcc-3.2.2-5.i386.rpm
- libstdc++-3.2.2-5.i386.rpm
- glibc-2.3.2-27.9.i386.rpm
- glibc-common-2.3.2-27.9.i386.rpm
- glibc-devel-2.3.2-27.9.i386.rpm

And to support Red Hat Linux Enterprise 4.0 (64-bit Linux) on an Intel Xeon or AMD Opteron processor platform, you will need at least the following library versions (possibly available from <http://updates.redhat.com>; you may need to have an account at that site).

- libgcc-3.4.3-9.i386.rpm
- libstdc++-3.4.3-9.i386.rpm
- glibc-2.3.4-2.9.i386.rpm
- glibc-common-2.3.4-2.9.i386.rpm
- glibc-devel-2.3.4-2.9.i386.rpm

If these libraries are already installed on your system, you can skip to the installation instructions under “[Installing LiveServer on Linux Step-by-Step](#)” on page 38. Otherwise, to upgrade these libraries, follow these instructions.

To upgrade these libraries,

1. Fetch the RPMs listed above.
2. Install these RPMs by running the following command when logged in as the root user:

```
rpm -i -U \  
libgcc-3.2.2-5.i386.rpm \  
libstdc++-3.2.2-5.i386.rpm \  
glibc*-2.3.2-27.9.i386.rpm
```

Or

```
rpm -i -U \  
libgcc-3.4.3-9.i386.rpm \  
libstdc++-3.4.3-9.i386.rpm \  
glibc*-2.3.4-2.9.i386.rpm
```



Note: If you have previously installed additional glibc RPMs (profiling libraries, debugging libraries, and others.) you may receive an error message of the form “Error: removing these packages would break dependencies:” To complete the upgrade, you will need to fetch some or all of the remaining glibc*-2.3.2-27.9.i386.rpm packages from the `/redhat/linux/updates/9/en/os/i386/` subdirectory and re-run the installation command.

3. Once you have installed these libraries, you can proceed with installing LiveServer as described under “[Installing LiveServer on Linux Step-by-Step](#)” on page 38.

Installing LiveServer on Linux Step-by-Step

For Linux, LiveServer is provided in two RPM Package Manager (RPM) installation files:

- `liveserver-3.x.x.x-rhe30.rpm` (for 32-bit Linux)
- `liveserver-3.x.x.x-rhe40-64.rpm` (for 64-bit Linux)

(Where *x* represents a version number.) The default location for installation is `/opt/knserver/`. You will need to be logged in as the root user.

The RPM installation file installs LiveServer and places the files and directories in the correct locations. It also installs a `knserver` init script in `/etc/init.d` (similar in concept to the Windows service), which means that LiveServer will automatically start during the Linux boot process, and will be automatically restarted. Before starting, make sure you have updated the C and C++ run-time libraries as described under “[Pre-Installation Tasks](#)” on page 35.

To install on Linux,

1. Log in as the root user.
2. Execute one of the following commands, depending on which version of Linux you are running (replacing *x.x.x* with the LiveServer’s version number).

- For 32-bit Linux:

```
rpm -i /path_to_file/liveserver-3.x.x.x-rhe30.rpm
```

- For 64-bit Linux:

```
rpm -i /path_to_file/liveserver-3.x.x.x-rhe40-64.rpm
```

3. Start LiveServer using the following command. (On later occasions, LiveServer will restart itself, but for first installation, it must be manually restarted.) This command works if you are logged in as the root user.

```
/etc/init.d/knserver start
```

Later, if you are not logged in as the root user, you can start LiveServer with the following command:

```
sudo /etc/init.d/knserver start
```

If LiveServer fails to start the first time, look at the log files in `/opt/knserver/log/` for troubleshooting information. If you cannot determine from the log files why LiveServer did not start, zip any files in the `/log/` directory and send them to KnowNow Technical Support for further assistance.

4. You can stop LiveServer using the following command.

```
/etc/init.d/knserver stop
```

Setting the Number of Client Connections

After installing LiveServer, you may wish to configure the number of clients that can connect simultaneously. To do so, edit the LiveServer startup script `/etc/init.d/knserver` and insert the following line after the line that reads `"start() {"`:

```
ulimit -n 65536
```

This line tells the Linux system to increase the maximum file descriptors to 65,536. The default value varies by Linux platform, but is usually 1,024. Each LiveServer connection uses one file descriptor, so this number will limit the number of clients that can connect simultaneously. If the number is set too low, LiveServer will start refusing connections once it reaches the limit. For additional information on setting the number of connections, see the Workload parameter [Peak New Connections](#).

Verifying that LiveServer is Running

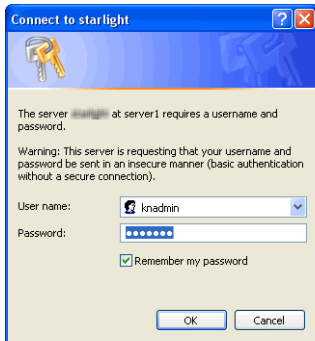
Once LiveServer is installed and started, you can verify that it is running by accessing the LiveServer status page.

The following instructions work for Windows and Linux; however, if you have trouble with this process on Windows, you can use the alternate instructions provided under “Verifying the LiveServer Service Under Windows” on page 325.

To verify that LiveServer is running,

1. Open Internet Explorer and connect to `http://localhost:portnumber/`, where *localhost* is the name or IP address of the machine running LiveServer, and *portnumber* is the port number. The port number is set during installation, though it can also be changed through the **HTTP Port** parameter. For a list of default port numbers, see “HTTP Port” on page 62.
2. If you see a login window (similar to that shown in Figure 2-16), either use your user name and password (if this is an upgrade), or use the default **knadmin** user with the default password of *knadmin*. Of course, you will want to change these defaults as soon as possible; for information on how to do so, see Chapter 9, “Managing Users and Groups.”

Figure 2-16. Logging in to LiveServer.



3. If LiveServer is running, you will see a Web page providing access to the KnowNow System Administration console, along with links to the LiveServer documentation

and to the KnowNow Web site. If you also installed KnowNow ESS, you will see a link to the KnowNow ESS Administration console as well.

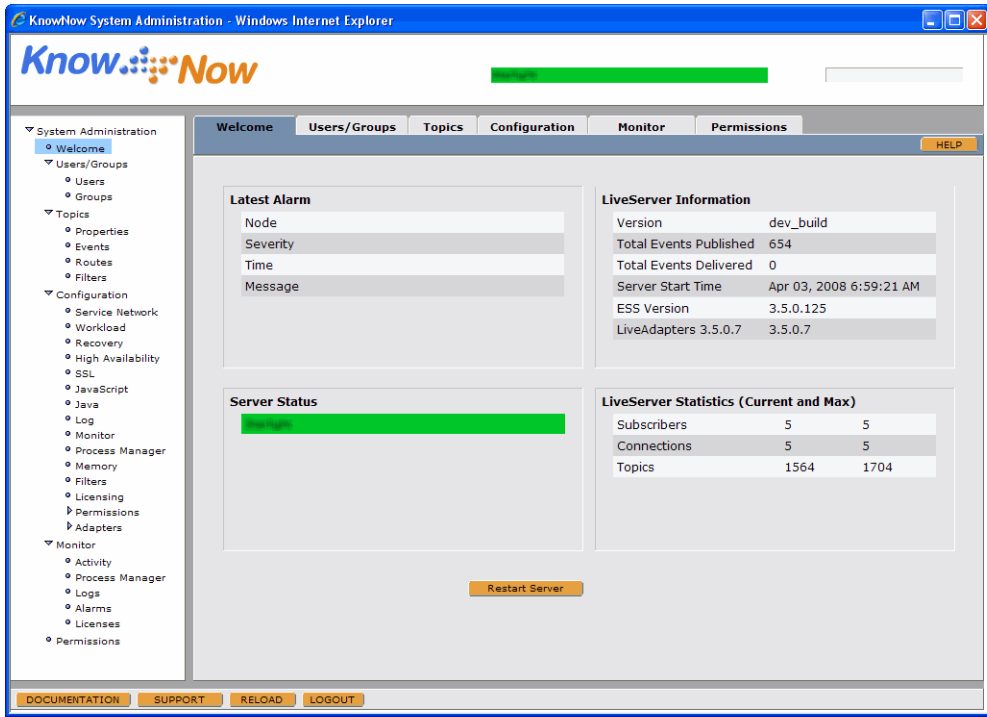
Figure 2-17. The KnowNow opening screen.



4. Choose **System Administration** to open the KnowNow System Administration console. Here, you can configure LiveServer, ESS, and LiveAdapters, choose your

permission and authentication policies, and perform other tasks as described in this book.

Figure 2-18. LiveServer is running.

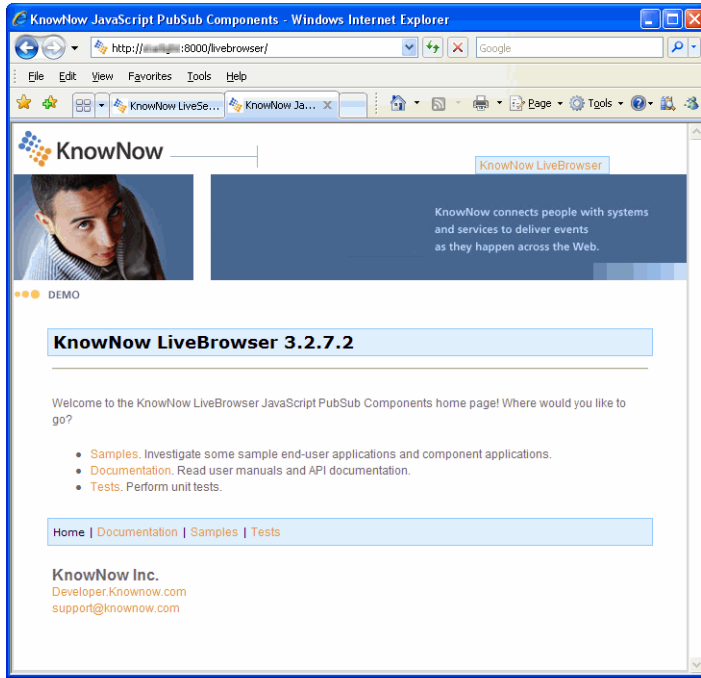


If the LiveServer's name is listed with green behind it, LiveServer is running. If you are not able to connect to LiveServer, see ["Troubleshooting"](#) on page 325, refer to the KnowNow Support site (as described under ["Additional Sources of Information"](#) on page xviii), or contact KnowNow Support (as described under ["How To Contact KnowNow, Inc."](#) on page xx).

5. To verify that LiveBrowser is also operating correctly, using any of the browsers supported by LiveBrowser, browse to `http://machinename:portnumber/livebrowser`. (You may need to sign in using the proper credentials as described in [step 2 on page 40](#).) If all is well, the welcome page appears, similar to what is shown in [Figure 2-19](#). For information on configuring and using LiveBrowser, see the *KnowNow LiveBrowser Developer's*

Guide. Links to the LiveBrowser documentation are provided on the LiveBrowser pages.

Figure 2-19. The LiveBrowser's Welcome page.



Chapter 3 Introducing the KnowNow System Administration Console

The KnowNow System Administration console is a graphical user interface that provides a way to configure and manage LiveServer. It uses the KnowNow LiveServer and Connector technology, so it is a demonstration of some of the kinds of things you can accomplish yourself with KnowNow products. Most of the chapters in this book explain how to use the KnowNow System Administration console to perform system configuration and other tasks; this chapter provides an overview of how to use the interface itself under the following headings:

- “Accessing the KnowNow System Administration Console” on page 46
- “The Main KnowNow System Administration Window” on page 48
- “Accessing LiveServer Documentation and Help” on page 52
- “Entering Data and Editing LiveServer Parameters” on page 53

Accessing the KnowNow System Administration Console

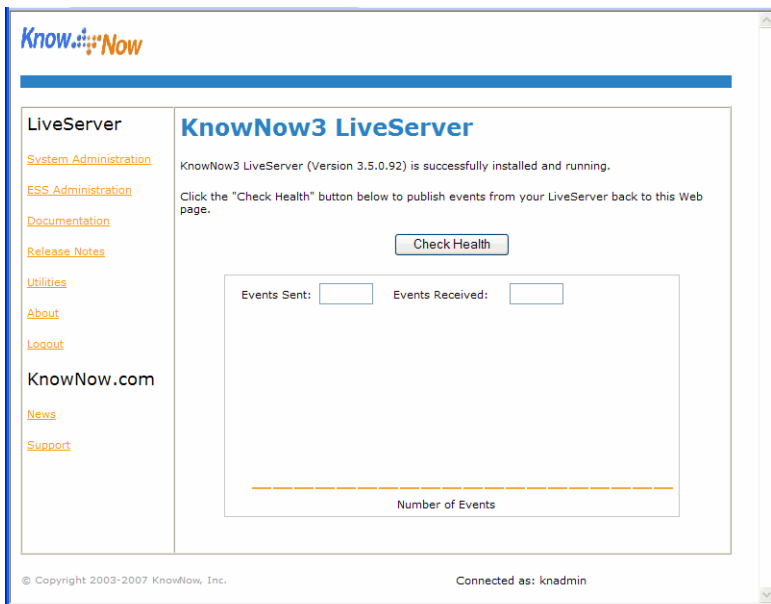
To access LiveServer, open an Internet Explorer browser window and enter `http://yourhost:portnum`. This opens a Web page providing access to the KnowNow System Administration console, along with links to the LiveServer documentation and to the KnowNow Web site. (For more information on accessing LiveServer, see “Verifying that LiveServer is Running” on page 40.)



Note: You must use a supported version of Internet Explorer to access the KnowNow System Administration console. For more information, see “System Requirements and Supported Platforms” on page 16.

You can log out here if you wish to log in as another user. Depending on the authentication policy, you may be prompted to log in. (For more information on authentication policies, see Chapter 6, “Selecting a Permissions System.”)

Figure 3-1. The KnowNow opening screen.



To access the KnowNow System Administration console, choose **System Administration** (under **LiveServer**). You will see the KnowNow System Administration Welcome page as discussed under “[The Main KnowNow System Administration Window](#)” on [page 48](#).

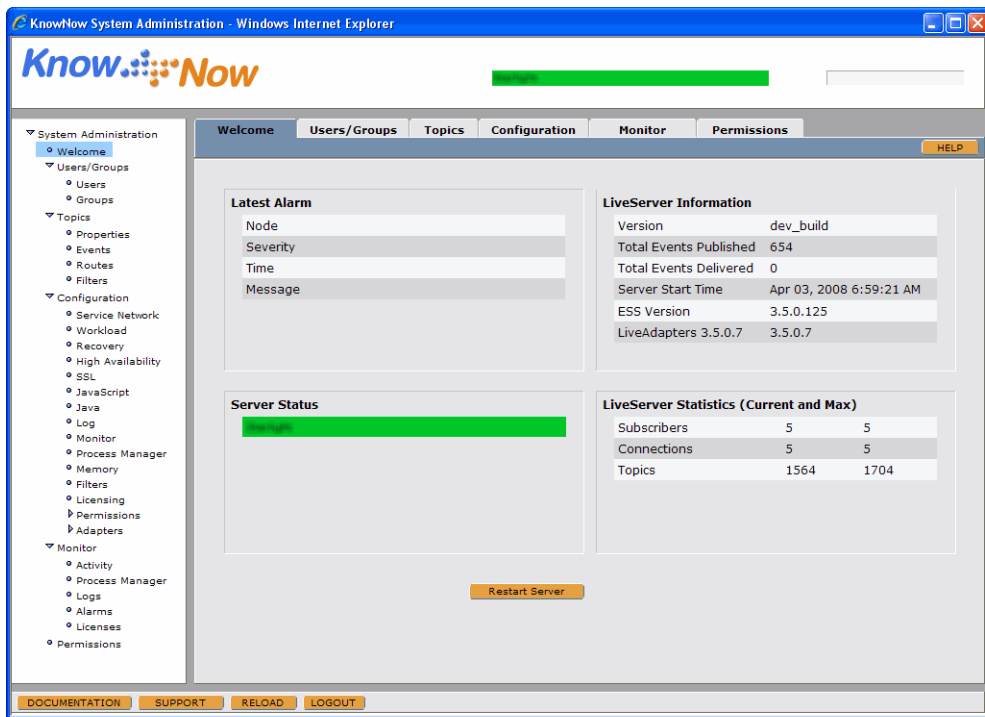
In addition to being able to log out or access the KnowNow System Administration console, you can also access the following by choosing the appropriate links:

- If you have ESS installed, the KnowNow ESS administration console.
- The LiveServer documentation. If you wish to add links to your own documentation, you can edit the `index.html` that is located in `/LiveServer_root/Server/servers/server1/pages/kn_docs/`. (The directory `/server1/` may differ depending on which server you are performing this task for.) Be careful not to change the links for the LiveServer’s documentation, and, if you change this file, be sure to back it up before upgrading to a newer version of the LiveServer, as otherwise your changes will be overwritten during the upgrade process.
- The LiveServer’s release notes.
- The LiveServer’s utilities; for more information, see [Chapter 12 , “Backup, Restore, and Other Utilities.”](#)
- Credits and thanks to those who have contributed to this product (by choosing the **About** link).
- A **Logout** link.
- Specific pages at the KnowNow Web site, such as **News** and the **Support** Center.

The Main KnowNow System Administration Window

To the left on the Welcome page is a list providing access to all the functions of the KnowNow System Administration console; to the right are tabbed pages providing detailed information about any selected aspect of LiveServer. These tabbed pages provide organized, task-oriented ways of configuring different aspects of LiveServer, such as topics, clusters, and so on.

Figure 3-2. The KnowNow System Administration Welcome page.



The KnowNow System Administration console is organized so that related tasks can be accessed at the same time, and so that most tasks only need to be performed once. For example, although the domain name is used by different modules and functions, it only needs to be set or changed in one place.

The tasks at a high level are

- configuring various aspects of the system, including the network, workload, SSL, recovery, and so on, which is described in more detail in [Chapter 4, "Configuring LiveServer."](#)

- configuring and managing topics, which is described in more detail in [Chapter 5](#) , "Managing Topics."
- selecting and configuring a security module, which is described in more detail in [Chapter 6](#) , "Selecting a Permissions System" and in [Chapter 7](#) , "Implementing Security with LDAP"
- managing permissions, users, and groups, which is described in more detail in [Chapter 8](#) , "Managing Permissions" and [Chapter 9](#) , "Managing Users and Groups."
- creating and configuring clusters, which is described in [Chapter 10](#) , "Implementing Clusters."
- monitoring performance and licenses,¹ including fault identification, isolation, and removal, which is described in more detail in [Chapter 11](#) , "Performance Monitoring and Process Management."
- using LiveServer utilities and performing troubleshooting activities as described in [Chapter 12](#) , "Backup, Restore, and Other Utilities."

Accessing the Capabilities of the KnowNow System Administration Console

Many of the functions and capabilities of the KnowNow System Administration console are accessible in two or more ways. For example, to access the Topics tab, you can either click on the word Topics in the table of contents to the left, or you can click on the Topics tab itself. Either action brings the Topics tab to the front.

Each tabbed page has additional capabilities that are accessed in several ways as well. For example, when you have a topic selected, you can view and manage that topic's properties, events, routes, and filters. The ability to manage those things is provided through a set of tabs on the Topics page. So, for example, if you wish to view the events for a selected topic, you can either click on Events under the word Topics in the table of contents, which will open the Events tab within the Topics tab, or you can click on the Events tab within the Topics tab. Either action opens the Events tab so you can view events for the selected topic. These are just examples of how you can access the same functionality in different ways. Details about accessing any given functionality are provided in later chapters.

1. Check this with new version.

LiveServer States

Whether you have one node or several in your system, they will each be listed on the Welcome page. Behind each node name is a band of color. The color indicates the state of that machine. Depending on the state, either green or red will be behind the node name on the Welcome tab. Each node can be in a different state.

- Green indicates that LiveServer can communicate with the machine.
- Red indicates LiveServer cannot communicate with the machine.

The states can be more finely defined as follows:

- Startup (red). The node is starting up but hasn't started recovery.
- Recovering (red). The node has identified a cluster node to start up from (even if that node is itself) and it is in the process of recovering.
- Active (green). LiveServer is servicing requests.
- Detached (red). A cluster peer node is said to be Detached if this node is no longer able to communicate with that peer.
- Offline (red). A cluster peer node is said to be offline if this node has never been able to communicate with it.

In addition, on the Welcome page, other information about the status of the KnowNow System Administration console's LiveServer is displayed.

For more information on system performance monitoring, see [Chapter 11 , "Performance Monitoring and Process Management."](#)

Configuration Status Field

Some of the changes you make using the KnowNow System Administration console require restarting LiveServer. Other changes are implemented instantly and do not require a restart.

When you make changes, if a restart is needed, it is indicated by a field in the upper right corner of the KnowNow System Administration console.

To restart LiveServer, select the **Restart Server** button on the Welcome page. (If you are running a clustered configuration, this button is labeled **Restart Cluster**.)


The Reload Button

The **Reload** button at the bottom of each page makes it possible to refresh the KnowNow System Administration console; for example, if the connection to LiveServer has been down. This command does not restart LiveServer or load previous configurations.

Accessing LiveServer Documentation and Help

The KnowNow documentation can be accessed from the opening screen (Figure 3-1 on page 46) by choosing **Documentation**. This link leads to a page of links to the KnowNow documentation.

The **Help** buttons in the KnowNow System Administration console lead to context-sensitive information in the KnowNow documentation.

 Also, in some panels in the KnowNow System Administration console, you will see a white lowercase letter *i* in a small blue circle. If you hover your cursor over that letter, you will see additional information on the field it is associated with.

Entering Data and Editing LiveServer Parameters

Entering data and editing the values of parameters in the KnowNow System Administration console is generally accomplished by entering data into fields or by choosing options from drop-down lists. Some fields are marked with a red asterisk (*); these fields are required and must be unique. Other fields are marked with a black asterisk; these fields are required but do not need to be unique.

Some parameters are set by specifying a value or a string.

- In the case of the values, there are certain rules that govern their use, as described under [“Specifying Units in Parameters” on page 53](#).
- In the case of strings, since the configuration file is written in Tcl, you can use Tcl expressions to set the parameters, as described under [“Tcl Expressions in LiveServer Parameters” on page 54](#).

The meanings of all parameters are provided in the section of the documentation where the configuration of those parameters is described; this section provides information that applies globally.

Specifying Units in Parameters

The KnowNow System Administration console uses consistent abbreviations for units of measure for values assigned to parameters. Note that case does not matter.

- Times are specified by providing a number followed by a unit of measure. The units of measure are **s**, **m**, **h**, **d**, and **w**, representing seconds, minutes, hours, days, and weeks. If you want to specify a year, use **52.57w**. In most cases, you can also use the word **infinity**.
- For payloads, the unit of measure is in bytes: **KB**, **MB**, **GB**, and **TB**, for kilobytes, megabytes, gigabytes, and terabytes respectively. At KnowNow, these mean 2^{10} , 2^{20} , 2^{30} , and 2^{40} . Case doesn't matter, so, for example, uppercase **KB** and lowercase **kb** mean the same thing.
- Routes are expressed in Hertz using the abbreviation **Hz**; for example, **1Hz**. You can use the letter **P** (which stands for per) followed by a time period to specify a frequency; for example, **60Ps** means 60 cycles per second, or 60 Hertz. Other ways you could express that same rate of 60 Hertz are **60Hz** and **3600Pm**. You can also use a slash (*/*) to replace the **P**; for example, **3600/m** is the same as **3600Pm**. Again, case is not important; you can enter **3600pm** and it means the same as **3600PM**, **3600Pm**, and **3600pM**.

The defaults for each of these units of measure are as follows:

- time is assumed to be in seconds
- memory is assumed to be in bytes
- rate is assumed to be in Hertz

So, for example, if you entered a value of **1** without a unit of measure for a time-based parameter, it would be assumed to be **1s**.

Tcl Expressions in LiveServer Parameters

Some LiveServer parameters can be edited using Tcl syntax in the KnowNow System Administration console. If you start a parameter value with a question mark (?), then the rest of the parameter is interpreted as a Tcl expression. Otherwise, it is interpreted as an absolute, literal piece of text.

For information on the LiveServer's parameters and variables, see the chapter or section in this book that relates to the area you are interested in configuring.

Special Characters in Parameters

The *knsettings.conf* file is in Tcl; therefore, you can use Tcl syntax when editing that file. In Tcl, some characters have special meanings and are interpreted by Tcl as commands even if included within quotation marks. These characters include the backslash (\), the dollar sign (\$), quotation marks (" "), brackets ([]), and braces ({ }).

To use special characters in *knsettings.conf*, such as in the value of a parameter, precede the special character with a backslash (\). For example, to include a brace in a variable, use \{. The same applies if you want to include a backslash: precede the backslash with a backslash, like this: \\.

For more information on Tcl and the Tcl syntax, see the Tcl Developer Xchange Web site: <http://www.tcl.tk>. A more detailed explanation of Tcl syntax can be found at <http://www.tcl.tk/scripting/syntax.html>. If you are new to Tcl, The Tcl'ers Wiki is a good place to start (<http://wiki.tcl.tk/>).

One of the more useful Tcl commands is `ns_info`, which you can use to get your host name, the version of LiveServer you are using, and so on. For information on `ns_info`, see <http://www.aolserver.com/docs/devel/tcl/api/admin.html>.

Chapter 4 Configuring LiveServer

After you have installed LiveServer, you may want to change some of the LiveServer configuration parameters. Configuration tasks are accomplished using the KnowNow System Administration console. This chapter provides information on configuring LiveServer under the following headings:

- “Location of LiveServer Configuration Information” on page 56
- “Default Settings” on page 57
- “Configuring the Service Network” on page 58
- “Characterizing Your Workload” on page 64
- “Recovery” on page 69
- “High Availability (Clusters)” on page 74
- “Configuring SSL” on page 75
- “Configuring the JavaScript Connector” on page 85
- “Enabling Java Support” on page 90
- “Log, Monitor, Process Manager, and Memory” on page 94
- “Filter Modules” on page 95
- “Configuring and Managing Licenses” on page 99
- “Permissions (Flat Files, LDAP)” on page 103
- “Configuring LiveAdapters” on page 104

Location of LiveServer Configuration Information

This chapter covers many of the LiveServer configuration tasks. However, some tasks require special attention and greater detail. Those tasks are described in separate chapters, even though the capabilities of performing those tasks are still found on the Configuration tab.

The following tasks can be accessed on the Configuration tab; the location of the information on how to perform those tasks is as given.

- To configure the service network, see [“Configuring the Service Network” on page 58](#).
- To configure the workload, see [“Characterizing Your Workload” on page 64](#).
- To configure recovery options, see [“Recovery” on page 69](#).
- To choose your high availability/cluster model and to configure clusters, see [Chapter 10 , “Implementing Clusters.”](#)
- To configure SSL, see [“Configuring SSL” on page 75](#).
- To configure JavaScript, see [“Configuring the JavaScript Connector” on page 85](#).
- To enable Java support, see [“Enabling Java Support” on page 90](#).
- To configure the log, monitor, Process Manager, and memory parameters, see [Chapter 11 , “Performance Monitoring and Process Management.”](#)
- To configure filters, or to edit, attach, and detach filters, see [“Topics and Filter Modules” on page 128](#). To load filter modules into LiveServer, or to delete them, see [“Filter Modules” on page 95](#).
- To configure licensing, see [“Configuring and Managing Licenses” on page 99](#).
- To configure permissions, see [Chapter 6 , “Selecting a Permissions System.”](#)
- To configure LDAP, see [Chapter 7 , “Implementing Security with LDAP.”](#)
- To configure LiveAdapters, see [“Configuring LiveAdapters” on page 104](#).

Default Settings

LiveServer is both secure and functional out of the box with the following default settings.

- The initial installation is for single-node service (not clustered), serving on HTTP port 8000 and HTTPS port 8443. For a list of the default port numbers, see [“HTTP Port” on page 62](#) and [“HTTPS Port” on page 63](#). For cluster-specific configuration information, including information on enabling and configuring clusters, see [Chapter 10, “Implementing Clusters.”](#)
- SSL and HTTPS are enabled by default.
- An initial set of users is configured. These users are described under [“The LiveServer’s Default Users and Groups” on page 252](#).
- An initial set of user-oriented topics is pre-installed: /who and /what.
- Security is enabled by default. For a discussion of the available security modules, as well as information on configuring security and managing permissions, see [Chapter 6, “Selecting a Permissions System,”](#) [Chapter 7, “Implementing Security with LDAP”](#) and [Chapter 8, “Managing Permissions.”](#)
- The recovery module is enabled by default and the **Checkpoint Interval** parameter is set to **60s** (60 seconds), so LiveServer persists its state to disk every minute.
- The default value for each parameter is listed with that parameter in the KnowNow System Administration console. If no default value is displayed, then there is no default value for that parameter.
- The LiveServer’s main configuration file is *knsettings.conf*. When you make changes to the LiveServer’s default settings using the KnowNow System Administration console, those changes are written to this file. Normally, you should not touch this file.

Configuring the Service Network

For a single machine, your service network is your set of NICs on your machine; in the case of a deployed cluster, the service network is the network that is facing your customers, providing access to the network. The cluster network is the internal network providing for communication among the clustered nodes.

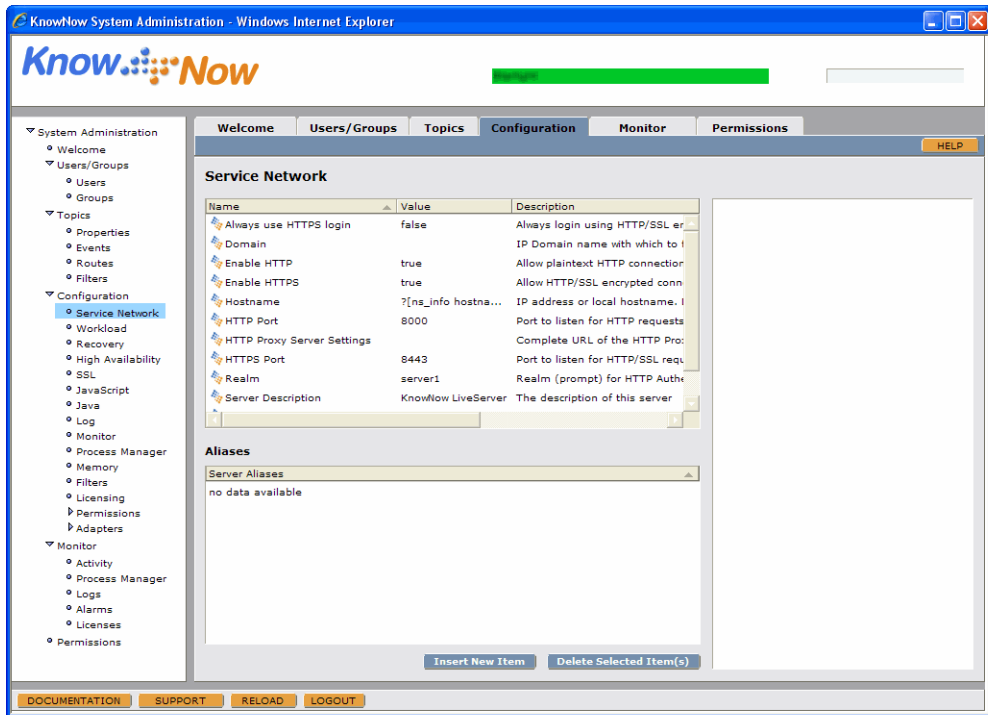
The KnowNow System Administration console's Service Network panel is where you configure your service network: domain name, HTTP/HTTPS protocols, host names, and so on. A number of the settings are used in other areas, such as by the High Availability choices.

To configure the Service Network,

1. Access the KnowNow System Administration console.
2. Under **Configuration**, choose **Service Network**. The Service Network panel is displayed. On this panel you will see a list of fully qualified names (aliases) for the machines that LiveServer is running on. This list can include various combinations

(such as *machine name:port number;IP address:port number*) that identify the machine as the host machine on which LiveServer is installed.

Figure 4-1. The Service Network panel.



- For each parameter you wish to configure, select that parameter. A field appears to the right for modifying the value of that parameter. Modify the value of the parameter in the field to the right, then choose **Update**. The meanings and values of the parameters are provided under “[The Service Network Parameters](#)” on page 60.
- The lower panel is labeled **Aliaes**. This is where you can add or delete other names for this LiveServer (machine or cluster). The list of server aliases is a list of names for this machine other than the names entered in the **Domain** and **Hostname** parameters. This identifies which host names are part of the cluster and which are not. This could be used to add the aliases for the host names that are part of the cluster.
 - To add a name, choose **Insert New Item** (at the bottom of the panel). To the right, a field labeled **Server Alias (Full URL prefix)** appears. Enter an aliases in

the field and choose **Insert**. Each name must have a full URL prefix that gets access to this LiveServer and must include the protocol section and a trailing /kn; for example, `http://www.example.com/kn` or `https://www.example.com:8443/kn`. Names can include path components in the URL, and support reverse proxies.

As you add these names, they are added to the list in **Aliases** section. For example, if you have a reverse proxy that wants to refer to LiveServer as `http://example.com/knownow/`, then enter that entire string as an alias.



Note: LiveServer verifies URL prefixes without regard to case. However, non-protocol, non-host components of HTTP URLs are case sensitive. Therefore, when adding entries to the list of names, use the proper case for those names.

If you choose **Cancel**, your addition will be canceled.

- To delete a name, select one or more names in the **Aliases** column and choose **Delete Selected Item(s)**. You can select discontinuous items by pressing and holding the control key before clicking on the items.

5. If you are done making changes, restart LiveServer.

The Service Network Parameters

The Service Network parameters are described in this section. For an explanation of the format for units in parameters, see [“Entering Data and Editing LiveServer Parameters” on page 53](#).

Always use HTTPS login

When set to **true**, logins will always require HTTP/SSL encrypted connections. The default value is **false**.

Domain

This is where you specify your LiveServer’s domain name. The domain name is the domain part of the host name that clients will use to access LiveServer. The JavaScript domain name defaults to this value; for more information, see [Domain on page 86](#).

A domain name looks something like the following:

```
example.com
```

For important information about this parameter in relation to clusters, see [Hostname](#).

Before configuring this parameter, determine which type of deployment you have:

1. Single-node systems where the usage of fully qualified host names (for example, `myliveserver.example.com`) is required
2. Clustered systems
3. Reverse proxy usage
4. SSL
5. Multi-tiered deployments where a third-party Web server serves up the Web pages for an application

If you have one of the first four deployments, then you will need to configure and use this parameter as well as the [Hostname](#) parameter.

For SSL, you will need to set the [Domain](#) parameter if the certificate sets the domain (for example, if you are using a wildcard certificate, where the domain is defined as `*.example.com`).

For information on configuring LiveServer for multi-tiered deployments where a third-party Web server serves up the Web pages for an application, see the JavaScript's [Domain](#) parameter, which is documented starting on [page 86](#).

Enable HTTP

Set this parameter to **true** or **false** to enable or disable the use of HTTP. You must enable at least either HTTP or HTTPS in order for service to be provided. The default value is **true**.

Note that a single-node LiveServer can be accessed with HTTP and HTTPS if both are enabled. If you must, you can disable HTTPS but not HTTP because Feedwatcher uses only HTTP.

Enable HTTPS

Set this parameter to **true** or **false** to enable or disable the use of HTTPS. You must have at least HTTP or HTTPS enabled in order for service to be provided. Setting this parameter to **true** enables the LiveServer's `nsopenssl` module, which supports SSL. The default value is **true**.

When this parameter is enabled, you will need to configure SSL as described under ["Configuring SSL" on page 75](#).

Hostname

This parameter specifies the host part of the name that clients will use to access LiveServer.

You can specify an IP address, but you don't need to. The default value of this parameter is `?[ns_info hostname]`, which tells LiveServer to figure out the host name by checking your machine. Do *not* include the domain name; that is specified using the **Domain** parameter.

Clustered Deployments Require This Parameter

For clusters, you need a software (or hardware) component that makes the machines in your cluster look like a single machine to your customers. The host and domain names that your customers use for this "single machine" *must* be configured in the Service Network panel (in the Hostname and **Domain** parameters). Therefore, you *must* specify a **Hostname** for clusters. For clusters, this parameter specifies the name of the cluster or of the cluster's load balancer. Note that a cluster node name cannot exceed 100 characters in length.

If running a cluster through a load balancer, and you want a unified view of the cluster through the load balancer, set this parameter to the name of the load balancer/cluster.

In clustered deployments, if you don't want LiveServer to use the host name, set the **Service Interface** parameter as described in [Chapter 10, "Implementing Clusters"](#) (step 4 on page 272).

For more information on clusters, see [Chapter 10, "Implementing Clusters."](#)

HTTP Port

This variable specifies the port that LiveServer listens on for requests. By default, LiveServer listens for requests on port 80 or 8000. For clusters, the default port number is 8001. For SSL, the default port number is 8443. For FLEXIm, the default port number is 27000. If the port you specify is already in use by another Web server, LiveServer won't restart.

The default value for this parameter is **8000**.

If you want to run your LiveServer on port 80,

1. Disable any other Web server (such as IIS or Apache) that is using port 80 or configure any other Web server to use a port other than 80.
2. Set this parameter to **80**.

HTTP Proxy Server Settings

Use this parameter to set the complete URL of the HTTP proxy server. The URL must be in the form of

```
http://user:password@host:port/realm?enable=false
```

HTTPS Port

This variable specifies the secure port that LiveServer listens on for HTTP/SSL requests. If the port you specify is already in use by another Web server, LiveServer won't restart. The default value is **8443**.

Realm

The realm for which the HTTP authentication applies. When users attempt to log in, the authentication prompt that appears is for the specified realm. The default value is **server1**.

Server Description

A text description of this LiveServer. The default value is **KnowNow LiveServer**.

Show server name in errors

When this is set to **true**, when LiveServer is returning an error page, it inserts the LiveServer's name and version number in the error message. The default value is **true**.

SMTP Server

LiveServer ships with a Tcl email filter that sends SMTP email in response to events. You can attach this filter to topics or routes. This parameter specifies the SMTP host for this filter. You can enter either an IP address or a host name. The default is **localhost**. For information on using the Tcl email filter, see the *KnowNow LiveServer Developer's Guide*.

Characterizing Your Workload

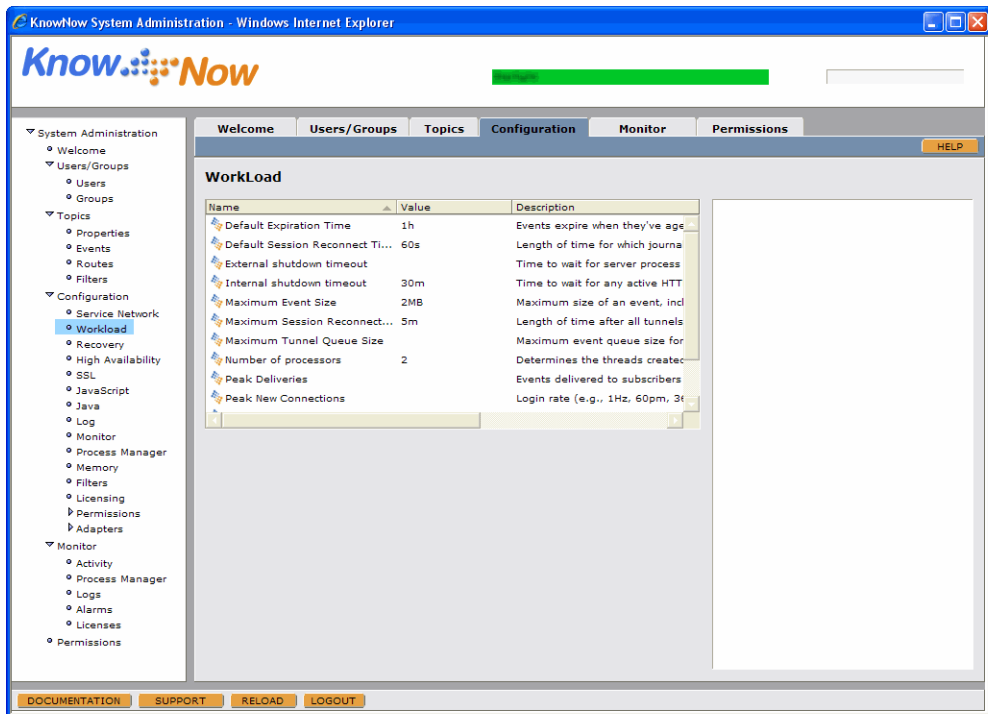
The Workload panel provides information needed to tune your system. The parameters in this section modify such things as the number of threads that will remain in a pool for servicing new connections, priorities for going through particular tasks, and the like. LiveServer performs tuning calculations based on the settings in the parameters in this panel.

Your system may well work quite fine without making settings in this panel. However, if, for example, you need 1,000 people to sign on within one minute, you may want to set the **Peak New Connections** parameter.

To configure the workload,

1. Access the KnowNow System Administration console.
2. Under **Configuration**, select **Workload**. The Workload panel is displayed.

Figure 4-2. The Workload panel.



3. For each parameter you wish to configure, select that parameter. A field appears to the right for modifying the value of that parameter. Modify the value of the parameter in the field to the right, then choose **Update**. The meanings and values of the parameters are provided under “[The Workload Parameters](#)” on page 65.
4. If you are done making changes, restart LiveServer.

The Workload Parameters

The Workload parameters are described in this section. For an explanation of the format for units in parameters, see “[Entering Data and Editing LiveServer Parameters](#)” on page 53.

Default Expiration Time

This sets the default expiration time for events within a topic. Events expire when they are the age set in this parameter. The default value is **1h** (one hour).

Events expire based on the following rules:

- `/kn_subtopics` topics **always** have an event expiration time set to **infinity**.
- `/kn_routes` topics **default** to having an event expiration time of **infinity**.
- `/kn_journal` topics default to having an event expiration time of the value set in the **Default Session Reconnect Time** parameter. To define a maximum amount of time a journal can be set to maintain events, use the **Maximum Session Reconnect Time** parameter.
- All other topics default to having an expiration time of the value set in this parameter (**Default Expiration Time**).

The event expiration time of any topic (except `/kn_subtopics` topics) or events within a topic can also be changed by using the **kn_default_kn_expires** header. For more information, see the *KnowNow LiveServer Developer's Guide*.

Default Session Reconnect Time

This parameter specifies the length of time that journal events are stored after connections close. If clients reconnect in the time period specified in this parameter, they can recover connections atomically. This time affects how long sessions remain open. For more information, see the Flat Files parameter “[Authentication policy](#)” on page 154 and the LDAP parameter “[Authentication policy](#)” on page 226.

The default value is **60s** (sixty seconds).

External shutdown timeout

This parameter and [Internal shutdown timeout](#) can be used to cause a very long shutdown time to insure that all data has been safely written to `knrouter_db`. If left blank, the default value of this parameter is two times the value of the [Internal shutdown timeout](#) parameter.

These variables can be tuned down if you insure that you first create a new snapshot just prior to shutdown in case a faster shutdown causes data corruption in your `knrouter_db`. This will allow you to restart on a snapshot that is only a few minutes old rather than from the previous day.

It is *strongly recommended* that you follow the following process prior to shutting down or restarting LiveServer or an ESS system:

1. Rename the previous day's snapshot to insure you have it saved and that it will not be overwritten in [step 2](#).
2. Create a new snapshot. (For one way to do so, see ["Restoring Your Database" on page 322](#).)
3. Shut down or restart LiveServer.

Internal shutdown timeout

See [External shutdown timeout](#) for a discussion of this parameter. The default value of this parameter is **30M** (thirty minutes).

Maximum Event Size

This parameter specifies the maximum size of an event, including header size. It also controls the maximum file upload size. The default value is **2MB** (two megabytes).

The value does not exactly correlate with the size of any single event. For example, if you use the batching API, several events get stuffed into a single POST. Other factors can affect message size as well. For more information on payloads and events, see the *KnowNow LiveServer Developer's Guide*.

Maximum Session Reconnect Time

Limits the maximum amount of time that a journal can be set to maintain events. Sets the length of time after all tunnels are closed that the journal is also closed. The default value is **5m** (five minutes). Developers can also change the amount of time that a journal stores its events by setting the **kn_default_kn_expires** property on a journal topic. For more information on **kn_default_kn_expires**, see the *KnowNow LiveServer Developer's Guide*.

Maximum Tunnel Queue Size

The value in this parameter limits the maximum size in bytes that the event queue for a slow tunnel is allowed to grow to on LiveServer before LiveServer drops the tunnel. This protects LiveServer against the scenario in which a slow tunnel and a fast publisher could cause the event backlog for that tunnel on the LiveServer side to increase the LiveServer's memory usage to a great extent and affect its functioning.

The symptoms that indicate this may be happening is that the LiveServer's memory usage grows proportionate to both of the following:

- the publish rate and
- the number of subscribers that are lagging behind the publisher significantly

The visible default value is empty, as LiveServer normally calculates this value internally. Any value set in this field overrides LiveServer's internal computation of this value.

If you are not sure you have this problem or what value you should set this parameter to, contact KnowNow Support.

Number of processors

The number of processors on this machine. Determines the threads created by LiveServer to handle connection requests and to service tunnels. 0 is not an allowed value. The default value is **2**.

Peak Deliveries

Specifies the number of events that are delivered to subscribers per a given time interval (per second, per minute, per hour...). This parameter helps to tune the number of resources that are applied to tunnels.

Peak New Connections

This parameter controls the maximum number of simultaneous connections LiveServer can handle in a specified interval (i.e., the login rate). This parameter adjusts the number of threads reserved for new connections.

You can set the rate using different units. For information on units, see [“Specifying Units in Parameters” on page 53](#).

To set this parameter, enter the largest number of new connections you would typically see over a small period of time. For example, the value **100pm** (100 per minute) would mean that you expect 1,000 new connections to be made over a ten-minute period.

You may also wish to change the number of connections in Linux as described under [“Setting the Number of Client Connections” on page 39](#).

Peak Publications

Specify the maximum number of posts you expect per a time interval (per second, per minute, or per hour).

Publishers

Specify the number of publishers you expect to be connected at once. This value is used to help tune the number of connections that will be on the system. The default value is **20**.

Subscribers

Specify the number of subscribers you expect to be connected at once. This value is used to help tune the number of connections that will be on the system. The default value is **1000**.

Recovery

LiveServer uses Sleepycat's Berkeley DB (see <http://www.sleepycat.com>) to maintain its persistent state on disk. This allows LiveServer to be stopped and then restarted in the same state; that is, all of the topics and events that exist when a LiveServer exits are available when the new LiveServer starts.

When you select one of the High Availability options, the recovery parameters are automatically enabled with reasonable default values, though you may still wish to configure recovery for your specific clustering needs.

During recovery, a clustered node will wait an accumulated total of 60 seconds (for overhead) plus the values in the following parameters:

- The High Availability parameter **Liveness wait** (for a node to be alive)
- The High Availability **Database snapshot timeout** (to snapshot that node)
- The High Availability **File transfer wait** (to transfer the files to its own database)
- The Recovery parameter **Database read timeout** (for its own database read)

For more information on clusters, and on how recovery interacts with clusters, see [Chapter 10, "Implementing Clusters."](#)

For information on moving recovery information, see ["Moving Recovery Information" on page 324.](#)

You can develop your own modules for handling recovery using the LiveServer's recovery API functions; for more information, see the information sources listed under ["Additional Sources of Information" on page xviii.](#)

Enabling and Configuring Recovery

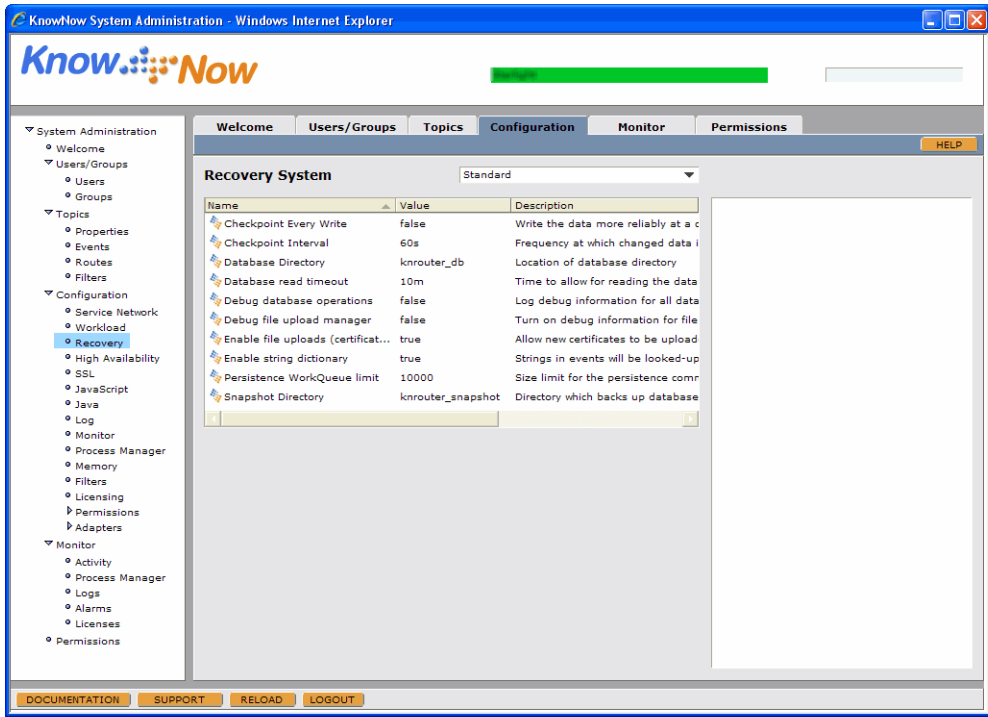
LiveServer supports data recovery through the use of a recovery module and recovery parameters. Recovery supports High Availability; in fact, when you enable one of the High Availability options, recovery is automatically enabled. The following instructions tell how to enable and configure recovery. For information on disabling recovery, see ["Disabling Recovery" on page 72.](#)

To enable and configure recovery,

1. Access the KnowNow System Administration console.

2. Under **Configuration**, select **Recovery**. The Recovery System panel is displayed.

Figure 4-3. The Recovery panel with recovery choices showing.



3. In the drop-down box, select a recovery method: **None** or **Standard**. If you select **None**, no recovery module will be used. If you select **Standard** (the default), KnowNow's recovery module will be used



Note: If you have enabled High Availability (clusters), recovery is automatically enabled and cannot be disabled. Even if you select **None**, if a cluster module has been enabled, LiveServer will still run the recovery module.

4. For each parameter you wish to configure, select that parameter. A field appears to the right for modifying the value of that parameter. Modify the value of the parameter in the field to the right, then choose **Update**. The meanings and values of the parameters are provided under "Recovery Parameters" on page 71.
5. If you are done making changes, restart LiveServer.

Recovery Parameters

This section lists the recovery parameters and describes what they do. For information on entering values, refer to [“Specifying Units in Parameters” on page 53](#). For an explanation of the format for units in parameters, see [“Entering Data and Editing LiveServer Parameters” on page 53](#).

Checkpoint Every Write

If **true**, causes LiveServer to flush all topic and event information to disk continuously (as events occur). This is called atomic mode. Atomic mode allows LiveServer to have the best chance to recover 100% of its state following a shutdown or crash, but may have a performance penalty. The default value is **false**.

If desired, developers can use the `do_method set_topic_property kn_atomic` to instruct LiveServer to continuously save a single topic’s events to disk as they occur. For additional information, see the *KnowNow LiveServer Developer’s Guide*.

Checkpoint Interval

Specifies how often, in seconds, recovery information will be saved to disk. For example, if the value is set to 600s, LiveServer flushes a snapshot of its state to disk every 600 seconds. This means that if LiveServer is abnormally terminated or crashes, it will only lose 600 seconds (at most) of information that cannot be recovered. LiveServer always tries to persist its state when it exits, so it is possible that even if LiveServer crashes, 100% may be recoverable. The default is **60s** (60 seconds).

Database Directory

This parameter specifies the location of directory in which you wish to store your database information. The default is **knrouter_db**.

Database read timeout

This parameter (the only one for single-node deployments) specifies the amount of time you want to allow LiveServer to read the database. If LiveServer cannot read the database within the specified time, it will not start up. The default is **10m** (10 minutes).

Debug database operations

When set to **true**, debugging information is logged for all database interactions. The default value is **false**.

Debug file upload manager

When set to **true**, enables debugging information when file uploads occur. The default value is **false**.

Enable file uploads (certificates, licenses, etc)

This specifies whether files can be uploaded to LiveServer. A certificate is one type of file that can be uploaded. The default value is **true**.

This parameter is also useful with clusters. When a new node is being assimilated into a cluster, you may need to enable this parameter on that new node's machine to enable files to be uploaded from the cluster to the new node.

Enable string dictionary

When enabled, strings in events will be looked up in a dictionary. The default value is **true**.

Persistence WorkQueue Limit

Specifies a size limit for the persistence command work queue. The default value is **10000**.

Snapshot Directory

This parameter specifies the location of the directory to which your database and other directories will be backed up during recovery. Especially, during cluster recovery, when a down node asks a hot node for a database, the hot node backs up its database to this directory, then transfers the database from this directory to the downed node.

The default is **knrouter_snapshot**.

Disabling Recovery

You can disable recovery if you are not using one of the High Availability options. After disabling recovery, be sure to restart your LiveServer.

To disable recovery,

1. Access the KnowNow System Administration console.
2. Under **Configuration**, select **Recovery**. The Recovery System panel is displayed.

3. In the drop-down box, select **None**.



Note: If you have enabled High Availability (clusters), recovery is automatically enabled and cannot be disabled. Even if you select **None**, if a cluster module has been enabled, LiveServer will still run the recovery module.

4. If you are done making changes, restart LiveServer.

High Availability (Clusters)

For information on configuring High Availability, see [Chapter 10](#), "Implementing Clusters."

Configuring SSL

You enable or disable basic secure sockets layer (SSL) support for LiveServer, including support for another LiveServer's (forwarding) certificate and key, by setting certain parameters as described under ["Configuring the Service Network" on page 58](#). This section provides information on configuring SSL.

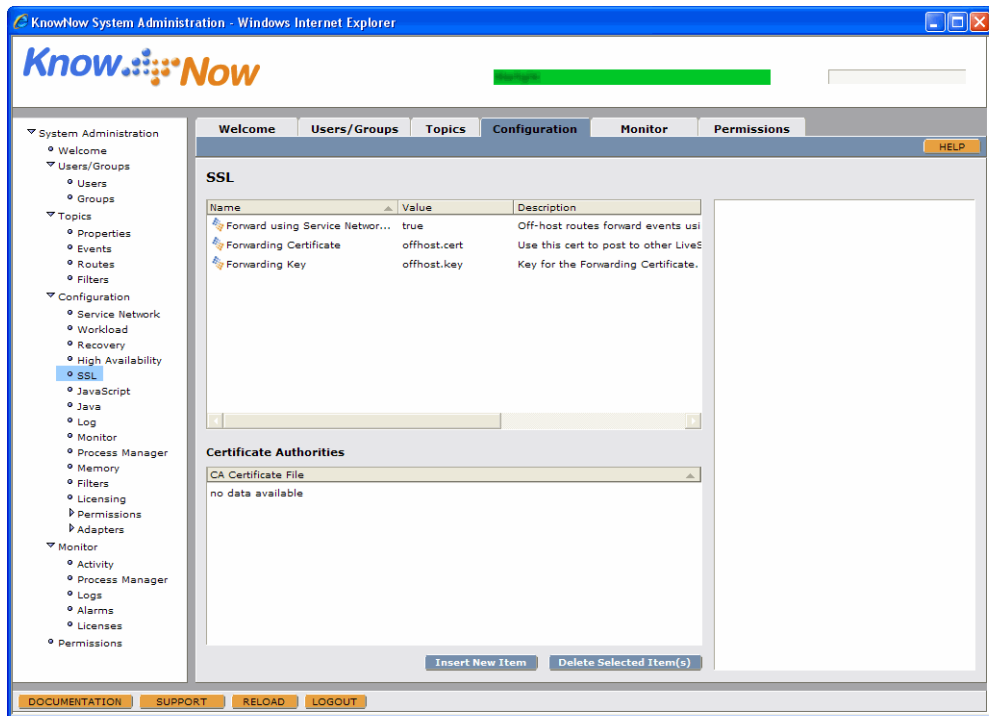
Use the following procedure to configure SSL. Depending on your needs, not all steps need to be performed.

To configure SSL,

1. In order to provide SSL support to customers, LiveServer must always have a certificate and a key. LiveServer is shipped with a default certificate and key, but KnowNow strongly recommends that you replace those immediately. You can either get the certificate and key from a trusted certificate authority (CA), or you can generate your own. The instructions under ["Using OpenSSL to Generate Certificates and Keys" on page 78](#) explain how to obtain certificates and keys from a CA.
2. Once you have your own certificate and key, upload them as described under ["Uploading Certificates and Keys" on page 80](#).
3. If you are using LiveServer to monitor a feed that is on a secure Web server and you are using a certificate that is not in LiveServer's list of trusted certificates, you will need to add that certificate to that list. For more information, see ["Adding Trusted CA Root Certificates" on page 82](#).
4. Now you are ready to configure the SSL parameters. To begin, access the KnowNow System Administration console.
5. Under **Configuration**, choose **Service Network** and configure the following Service Network parameters as needed.
 - **Domain** (if the certificate sets the domain; for more information, see [page 60](#))
 - **Enable HTTPS** (if this has been disabled; for more information, see [page 61](#))
 - **Hostname** (required for clusters; for more information, see [page 62](#))
 - **HTTPS Port** (if you need to change the port number from the default; for more information, see [page 63](#))

- Under **Configuration**, choose **SSL**. The SSL panel is displayed.

Figure 4-4. Configuring advanced SSL.



- If your LiveServer is also acting as a client to other LiveServers, or if clients are going to provide certificates to LiveServer (i.e., on a secure off-host route, when LiveServer is forwarding events), you will need to specify the certificate and key LiveServer should use when doing so. For this purpose, configure the SSL parameters **Forward using Service Network Certificate**, **Forwarding Certificate**, and **Forwarding Key**. Details on these parameters are provided under “[SSL Forwarding Parameters](#)” on page 77.
- When a client provides a client-side certificate, or when LDAP presents an SSL certificate, LiveServer needs to know which certificates to trust. In those cases, LiveServer needs to see a certificate that you have uploaded. To upload those certificates, choose **Insert New Item** and browse to (or enter) the path and name of the desired certificate(s). The new certificate(s) will be added to the list under **Certificate Authorities**.

9. To delete a certificate in the Certificate Authorities list, select it in the list in the lower panel and choose **Delete Selected Item(s)**. You can select discontinuous items by pressing and holding the control key while clicking on the items.
10. When you are done making changes, restart LiveServer.

SSL Forwarding Parameters

The SSL forwarding parameters are described in this section.

Forward using Service Network Certificate

If this parameter is set to **true** (the default value), then the **Forwarding Certificate** and **Forwarding Key** are **not** used; instead, the certificate and key that you uploaded as described under “[Uploading Certificates and Keys](#)” on page 80 will be used. If you haven’t uploaded your own certificate and key, the default certificate and key that were shipped with LiveServer will be used.

If this parameter is set to **false**, then the values in the **Forwarding Certificate** and **Forwarding Key** parameters are used.

Forwarding Certificate

Used for off-host HTTPS routes. The default value is **offhost.cert**.

This parameter should point to your LiveServer’s signed certificate file received from the certification authority (CA). LiveServer certificates need to be in PEM format, which can be generated by several public CAs, as well as by all the major CA infrastructure products.

This parameter, along with **Forwarding Key**, is only needed if you are allowing off-host routes; for example, use this parameter if you have two LiveServers, or a LiveServer and a Web server, and you want to forward events securely. This parameter is for your own LiveServer if it is acting as a client; that is, this is the certificate that your LiveServer will present to other clients.

Forwarding Key

Used for off-host HTTPS routes. The default value is **offhost.key**.

This parameter should point to the RSA private key for LiveServer. This parameter, along with **Forwarding Certificate**, is only needed if you are allowing off-host routes; for example, if you have two LiveServers, or a LiveServer and a Web server, and you want to forward events securely. This parameter is for your own LiveServer if it is acting as a client; that is, this is the key that your LiveServer will present to other clients.

Using OpenSSL to Generate Certificates and Keys

In order to provide SSL support to customers, LiveServer must always have a certificate and a key. LiveServer is shipped with a default certificate and key, but KnowNow strongly recommends that you replace those immediately. This section describes how to use the OpenSSL utility (included with LiveServer) to generate certificates and keys for your LiveServer to use, either in the service network or to present to other clients. You don't need to use the method described in this section to obtain a certificate and key; you can generate a certificate and key using another method. The important thing is to obtain your own certificate and key and replace the default ones in LiveServer as soon as you can.

Once you have obtained or created a certificate and a key, upload the certificate as described under ["Uploading Certificates and Keys" on page 80](#).



Warning: Certificates must be in PEM format, and the key that you upload cannot require a password to open it.

To generate an RSA private key and a certificate,

1. Generate an RSA private key for your LiveServer by running the OpenSSL utility, which is located in the LiveServer's `/bin/` directory. On Windows, the `/bin/` directory is usually under `C:\Program Files\KnowNow\Server\`. On Linux or UNIX, the `/bin/` directory is usually under `/opt/knserver/`. When you run the OpenSSL utility, you get a new shell or command prompt window. In that shell or window, run the following command (modifying the command to point to any local directory that is convenient for you, and modifying the file name as desired). Note that the OpenSSL command prompt is represented by `OpenSSL >`.

```
OpenSSL > genrsa -out /local_directory/yourliveserver-key.pem
```

After running the command, an RSA key in PEM format is placed in the specified directory.

2. When the OpenSSL prompt returns, generate a certificate signing request (CSR) as described in this step and in [step 3](#) and [step 4](#). If you wish, you can use an alternative way to generate a certificate request; for example, there are Web sites that generate certificate requests.

```
OpenSSL > req -new -key /local_directory/yourliveserver-key.pem -out
/local_directory/yourliveserver-request.pem
```

3. The req utility will prompt you for information for the certificate signing request. For each prompt, enter the appropriate information into the data fields. (To prevent confusion, in [Table 4-1](#) the original field labels have been retained “as-is”.)

Table 4-1. Example data for the fields for the certificate signing request.

req Prompt	Description	Example Data
Country Name (2 letter code) [AU]:	A standard two-letter code for the country your company is located in.	US
State or Province Name (full name) [Some-State]:	The full name of the state or province your company is located in.	California
Locality Name (eg, city) []:	The name of your locality; for example, the name of the city your company is located in.	Sunnyvale
Organization Name (eg, company) [Internet Widgits Pty Ltd]:	The name of your organization or company.	Widgets Inc.
Organizational Unit Name (eg, section) []:	The name of a unit within your organization; for example, a section name.	Development
Common Name (eg, YOUR name) []:	The fully qualified host name of the machine on which the certificate will be used. For clusters, enter the name of the cluster and not of a particular node.	www.widgets.com
Email Address []:	A contact email address.	sarah@widgets.com

4. When finished, the req utility prints a certificate request. Exit from OpenSSL by entering **exit** at the command prompt, then send this request to the appropriate CA and complete their approval process to receive a signed, valid certificate.
5. When you received a valid, signed certificate, place it in any convenient location on your machine, then upload the certificate and key as described under “[Uploading](#)”

Certificates and Keys” on page 80. If you are forwarding to another LiveServer, follow the instructions in [step 7 on page 76](#) to specify the forwarding certificate and key.

Uploading Certificates and Keys

Once you have obtained a certificate and key (either as described under “[Using OpenSSL to Generate Certificates and Keys](#)” on page 78 or by whatever other method you have chosen to use), the next step is to upload that certificate and key into LiveServer using the Rekey utility.

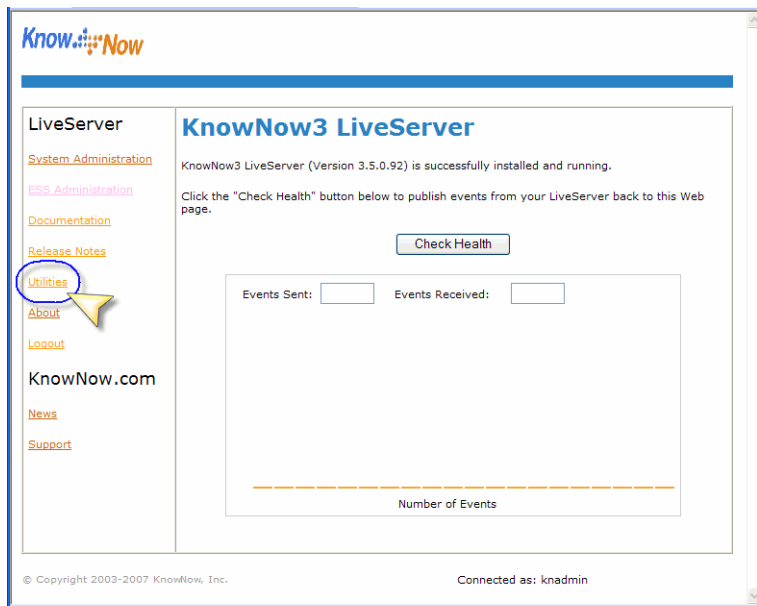
Use this same method if you have changed your certificate and key.

If your LiveServer is also acting as a client to other LiveServers, or if clients are going to provide certificates to LiveServer, then you will need to specify the forwarding certificate and key as described in [step 7 on page 76](#).

To upload a certificate and key,

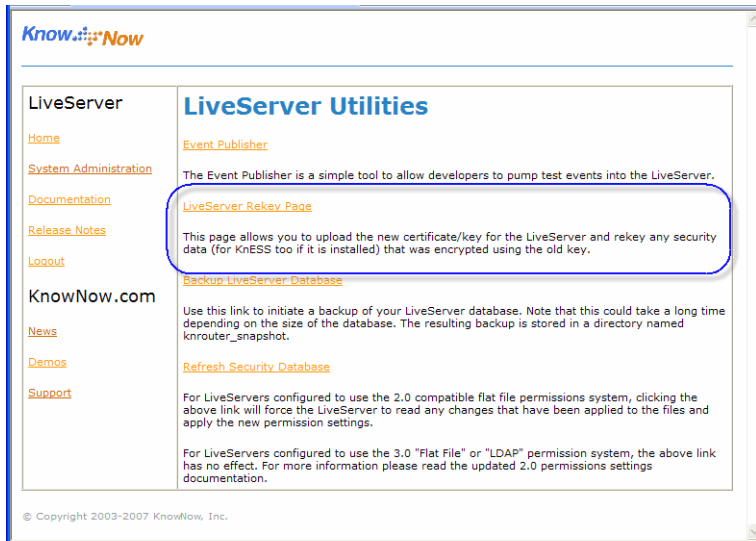
1. On the initial LiveServer page (the one you access the KnowNow System Administration console from), click on the Utilities link.

Figure 4-5. The location of the Utilities link.



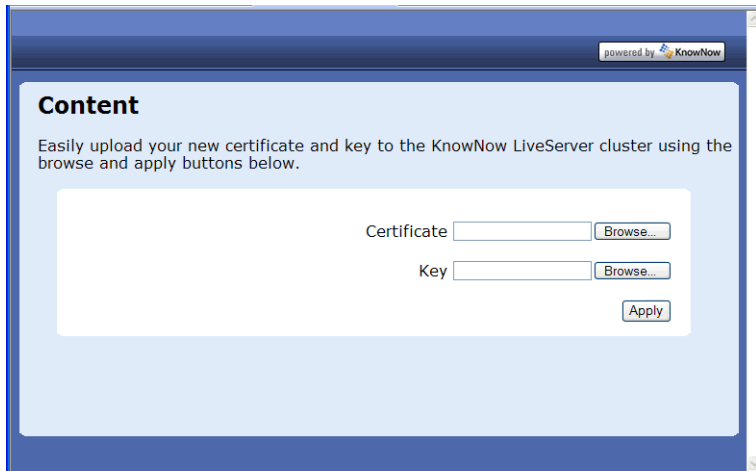
2. Click on the LiveServer Rekey Page link.

Figure 4-6. The LiveServer Rekey Page link.



3. On the rekeying page, either enter the location of the new certificate and key, or use the **Browse** button to find the new certificate and key, then choose **Apply**.

Figure 4-7. The LiveServer Rekey Page.



4. Upon success, LiveServer displays a pop-up window. Choose **OK** to close it. If there is a mis-match, then a different message appears. In that case, check to make sure you are using the matching certificate and key and try [step 3](#) again.

Figure 4-8. Certificate and key success message.



Adding Trusted CA Root Certificates

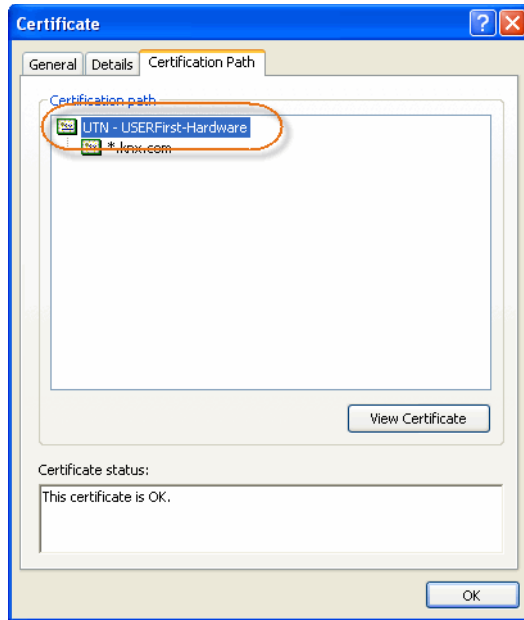
LiveServer ships with a list of trusted certificate authorities. Some certificates might have been issued by certificate authorities that are not on that list. If you are using LiveServer to monitor a feed that's on a secure Web server, and that Web server uses an SSL certificate that hasn't been issued by one of the authorities on LiveServer's list, then LiveServer will not watch the feed and may return an error. To resolve this issue, use the following instructions to add the new certificate to LiveServer's list of trusted authorities. Note that you must use Internet Explorer to accomplish the following procedure.

To add a trusted CA root certificate to LiveServer's list,

1. Using Internet Explorer, access the SSL site that uses the certificate you wish to add.
2. Double-click on the padlock icon to view the certificate. (You will need to have the status bar visible in order to see the icon.)

3. Access the Certification Path tab. You will see two certificates: The certificate authority (CA) certificate and the Web site's certificate. The CA certificate is the one you want to export.

Figure 4-9. Locating the certificate file.



4. Click on the CA certificate, then access the Details tab to export the certificate to a file.
 - a. On the Details tab, choose **Copy to File**. This starts the Certificate Export Wizard.
 - b. Choose **Next**.
 - c. Select **DER encoded binary X.509 (.CER)** as the format and choose **Next**.
 - d. Specify any desired file name and location, then choose **Next**.
 - e. Choose **Finish** to close the Certificate Export Wizard.
5. Copy the file to the machine where LiveServer is installed.
6. Import the certificate into LiveServer's list of trusted certificates. The paths are given based on the assumption that LiveServer is installed in the default location; if

it is not, replace `C:\Program Files\KnowNow\Server\` or `/opt/knserver/` with the correct string.

- In **Windows**, import the certificate by running the following command on command line:

```
C:\Program Files\KnowNow\Server\lib\JRE_directory\bin\keytool -import  
-trustcacerts -file certificate_file_including_path -keystore  
C:\Program Files\KnowNow\Server\lib\JRE_directory\lib\security\cacerts
```

- In **Linux**, import the certificate by running the following command:

```
/opt/knserver/lib/JRE_directory/bin/keytool -import -trustcacerts -file  
certificate_file_including_path -keystore /opt/knserver/lib/JRE  
directory/lib/security/cacerts
```

7. Restart LiveServer.

Configuring the JavaScript Connector

The JavaScript Connector is enabled by default. There are some parameters you can set for this module; for example, if you want to configure a cross-domain installation.

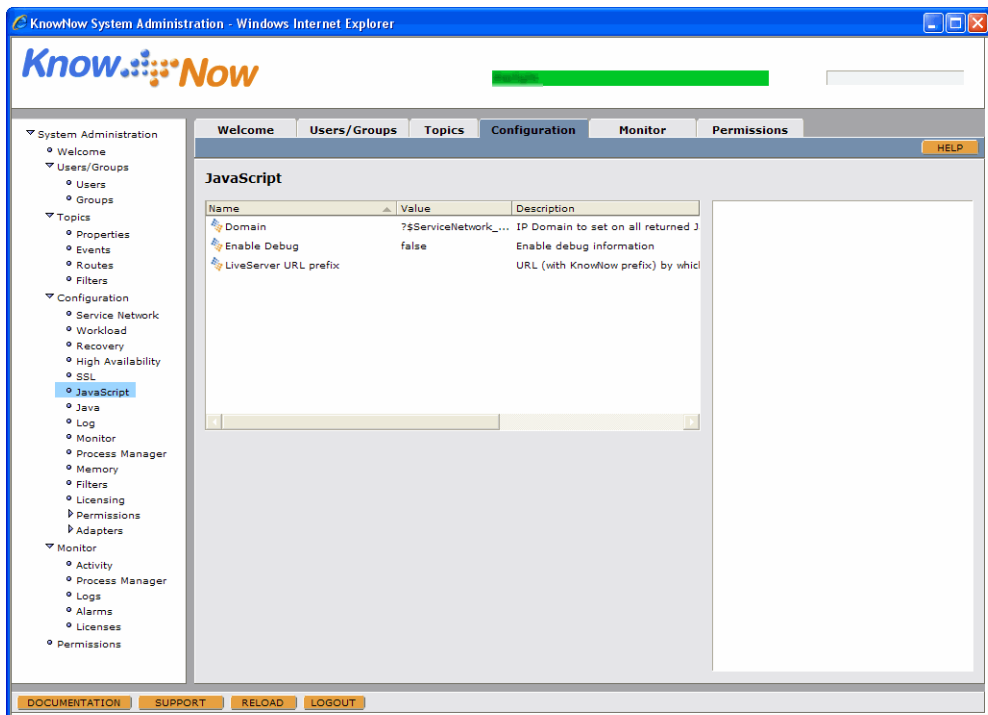


Caution: To restrict access on a hosting cross-domain server, you must use the hosting system's security system. If it's Apache, use .htaccess to control that access. In IIS, use its embedded security system. For further information on multiple Web server deployments, contact KnowNow's Support department.

To configure the JavaScript Connector,

1. Access the KnowNow System Administration console.
2. Under **Configuration**, select **JavaScript**. The JavaScript panel is displayed.

Figure 4-10. Configuring JavaScript



3. For each parameter you wish to configure, select that parameter. A field appears to the right for modifying the value of that parameter. Modify the value of the

parameter in the field to the right, then choose **Update**. The meanings and values of the parameters are provided under “[JavaScript Parameters](#)” on page 86.

4. If you are done making changes, restart LiveServer.

JavaScript Parameters

The JavaScript parameters are described in this section.

Domain

Use this parameter to set the domain for all returned JavaScript. This value is used in cross-domain LiveServer setups, such as when using the LiveBrowser on a different machine than the LiveServer’s. The default value is **?\$ServiceNetwork_Domain**, which means it uses the domain specified in the **Domain** parameter in the Service Network panel.



Caution: When configuring this parameter, do *not* use a trailing slash in the domain name. It is especially important to omit the trailing slash for cross-domain configurations.

Before configuring this parameter, determine which type of deployment you have:

1. Single-node systems where the use of fully qualified host names (for example, `myliveserver.example.com`) is required
2. Clustered systems
3. Reverse proxy usage
4. Windows SSL
5. Multi-tiered deployments where a third-party Web server serves up the Web pages for an application

If you are configuring LiveServer for one of the first four types of deployments, then do *not* use this parameter. Instead, configure the **Domain** and **Hostname** parameters as described starting on [page 60](#).

In the fourth case, *do* use this parameter, configuring it as described under “[Multi-tiered Deployments](#)” on page 87 and “[Cross-Domain Configuration Example](#)” on page 88.

For more information on...

- configuring the Service Network parameters, see [“Configuring the Service Network” on page 58](#).
- the Service Network’s **Domain** parameter, see [page 60](#).
- the JavaScript Connector, see the *KnowNow LiveServer Developer’s Guide*.
- the LiveBrowser, see the *KnowNow LiveBrowser Developer’s Guide*.

Multi-tiered Deployments

By default, the LiveBrowser’s JavaScript Connector assumes that all of the HTML pages that compose a KnowNow application will share the same host as the LiveServer’s. If you are installing LiveBrowser on a different machine than the LiveServer, or if you have a multi-tiered deployment where a third-party Web server serves up the Web pages for an application, then you will need to change this default assumption. To do so, there are three parameters you need to configure:

- The JavaScript **Domain** parameter (the one you are reading about right now). The JavaScript Connector uses the value in **Domain** to set the `document.domain` property, which signals to the browser that LiveServer is part of the same security domain as the HTML pages (loaded from the application server).
- The **LiveServer URL prefix** parameter, which provides a URL prefix by which JavaScript applications can reference LiveServer. For details on this parameter, see [page 89](#).
- The LiveBrowser **kn_config_server** parameter. The **kn_config_server** parameter identifies the machine on which LiveServer is located. This parameter is located in the LiveBrowser’s configuration file, *kn_config.js*. This file is located in the `/livebrowser/js/` directory. Details on configuring this parameter are provided under [“Cross-Domain Configuration Example” on page 88](#).

An example of configuring these three parameters is provided under [“Cross-Domain Configuration Example” on page 88](#). If you wish to use the Live.NET Connector’s ASP.NET controls, you will need to perform some additional configuration tasks, which are described in the *KnowNow LiveServer Developer’s Guide*.

In setting these three parameters, you are configuring LiveServer for use with LiveBrowser when LiveBrowser is on another machine. Essentially, you need to tell LiveServer to allow content that is served by other servers to access LiveServer, and you need to tell LiveBrowser where LiveServer is located.

Therefore, for cross-domain purposes, set the value in **Domain** to the common suffix of the application server and LiveServer domain names, or, if you are using a single IP address, use the domain or machine name. The value in **Domain** must be entirely in lowercase; for example, *example.com*. **Domain** must be set to point to the common value of the domains for LiveServer and LiveBrowser.

Cross-Domain Configuration Example

Let's say, for example, that LiveServer is hosted at *myliveserver.example.com*, while the pages that make up the application are being served from *apps.example.com*. (The two domain names must have a common suffix, such as *example.com*.)

In this case, the JavaScript **Domain** parameter would be set to *example.com*, which is the longest common suffix of the two domain names. This value will be used as a JavaScript `document.domain` setting by the JavaScript Connector.

The **LiveServer URL prefix** parameter is then set to the URL where LiveServer is found. In this example, that would be

```
http://myliveserver.example.com/kn
```

Notice that the URL includes the full protocol and host name, and ends with the `/kn` directory path but without a trailing slash.

For cross-domain configurations, modify the LiveBrowser **kn_config_server** parameter, which is located in the LiveBrowser's configuration file (`kn_config.js`, located in the `/livebrowser/js/` directory). This parameter is specified in the configuration file as follows, with comment lines (`//`) preceding the line:

```
kn_config_server = "//http://knserver.mycompany.com/kn";
```

To use this parameter, remove the comment lines and edit the URL so that it correctly identifies the machine on which LiveServer is located.

If you wish to use the Live.NET Connector's ASP.NET controls, you will need to perform some additional configuration tasks, which are described in the *KnowNow LiveServer Developer's Guide*.

Enable Debug

Enables the debugging version of the JavaScript Connector. The default value is **false**.

LiveServer URL prefix

Use this parameter to provide a URL with a KnowNow prefix by which JavaScript applications can refer to this LiveServer. This parameter is used in conjunction with the JavaScript **Domain** parameter and a LiveBrowser parameter when configuring a cross-domain setup; see “[Cross-Domain Configuration Example](#)” on page 88 for an example of setting the values of these three parameters.

The JavaScript Connector uses this parameter to determine the full URL for the LiveServer’s root event notification topic. For cross-domain purposes, this parameter needs to be set to point to the URL where LiveServer is to be found.

Set this parameter to the fully-qualified domain name for LiveServer followed by the `kn_server` path name, `/kn`. Include the port number only if you are not using the default port. This domain name must be identical to the one specified in the **src** attribute of the application’s `<script>` tag; for example,

```
http://myliveserver.example.com/kn
```



Note: It is important **not** to include a trailing slash or any double slashes (other than the one following **http:**) when modifying this parameter, because the value of this parameter is also used to resolve topic names, which use slashes.

It is also very important to write the host name entirely in lower case, because Web browsers internally represent host names in lower case for security-checking purposes. The LiveServer’s domain name must be written in fully qualified form, not as an IP address or abbreviated domain name, and Web browsers must be referred to that same fully qualified name.

Enabling Java Support

LiveServer ships with a Java module. This module provides a Java runtime environment for compiled Java modules to run inside. If you wish to write filter modules in Java, you must load the LiveServer's Java module so that your Java filter modules will run. By default, the Java module is not loaded. You can also provide command-line arguments for the LiveServer's Java interpreter.



Note: In order to use the Java module, you will need to have at least the Java Runtime Environment (JRE) or the Java Development Kit (JDK) installed. The JRE or JDK may be installed in any location on the machine. To communicate its location to LiveServer, set the `JAVA_HOME` environment variable to the full path name of the JRE or JDK before starting the LiveServer.

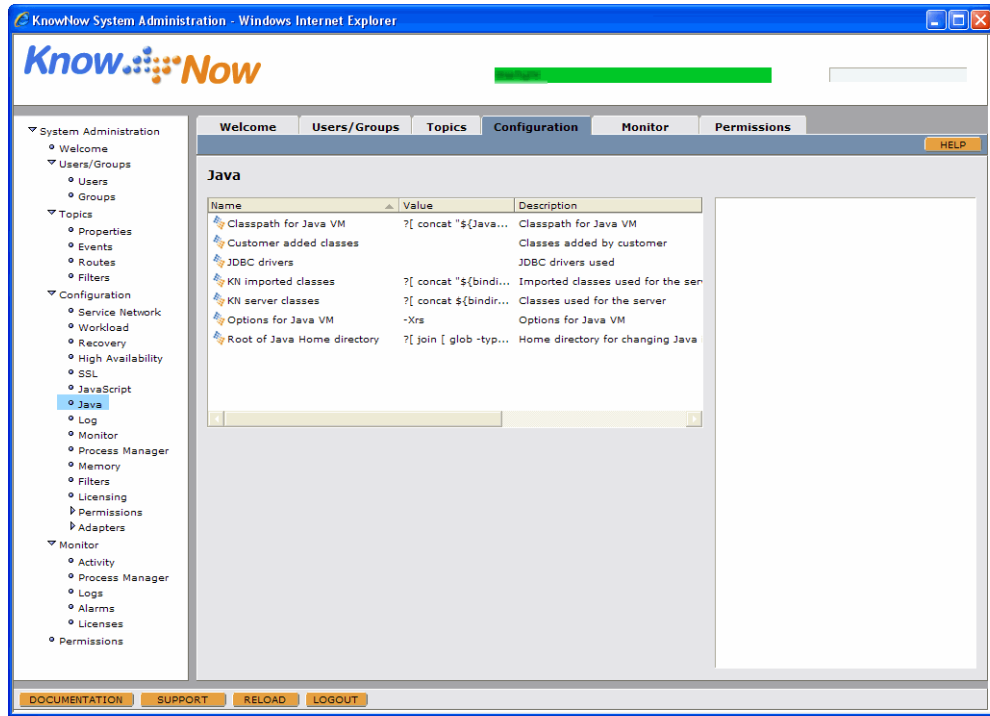
Also note that the Java module works only with 32-bit LiveServers.

To enable Java,

1. Access the KnowNow System Administration console.

2. Under **Configuration**, select **Java**. The Java panel is displayed.

Figure 4-11. The Java panel.



3. For each parameter you wish to configure, select that parameter. A field appears to the right for modifying the value of that parameter. Modify the value of the parameter in the field to the right, then choose **Update**. The meanings and values of the parameters are provided under “Java Parameters” on page 92.
4. If you are done making changes, restart LiveServer.

Java Parameters

The Java parameters are described in this section.

Classpath for Java VM

Sets the CLASSPATH for the Java VM. The default value is the following as a single line:

```
?[ concat "${Java_KNClasses}  
${pathseparator}  
${Java_ImportedClasses}  
${pathseparator}${bindir}  
${pathseparator}  
${Java_CustomClasses}  
${pathseparator}  
${Java_JDBCClasses}" ]
```

Customer added classes

Add your own classes using this parameter. To do so, click on this parameter and enter the class information in the open box to the right. Choose **Update** to add the class.

Enable Java-based filters?

If you wish to use the Java filter module, set this parameter to **true**. If true, the Java VM must be installed on the LiveServer's machine. The default value is **false**.

JDBC drivers

Use this parameter to specify the locations of your JDBC drivers.

KN imported classes

This parameter specifies classes imported into LiveServer. The default value is the following as a single line:

```
?[ concat "${bindir}/xalan.jar${pathseparator}  
${bindir}/activation.jar${pathseparator}  
${bindir}/axis.jar${pathseparator}  
${bindir}/commons-discovery.jar${pathseparator}  
${bindir}/commons-logging.jar${pathseparator}  
${bindir}/jaxrpc.jar${pathseparator}
```

```

${bindir}/mailapi.jar${pathseparator}
${bindir}/saaj.jar${pathseparator}
${bindir}/wsdl4j.jar${pathseparator}
${bindir}/../lib/livejava/lib/log4j-1.2.8.jar${pathseparator}
${bindir}/../lib/livejava/lib/jcert.jar${pathseparator}
${bindir}/../lib/livejava/lib/microserver.jar${pathseparator}
${bindir}/../lib/livejava/lib/jsse.jar" ]
```

KN server classes

This parameter specifies LiveServer classes. The default value is **?[concat \${bindir}/kn-router.jar]**.

Options for Java VM

Sets options to pass to the Java VM. The default value is **-Xrs**.

Root of Java Home directory

Sets the home directory for changing Java implementations. The default value is **?[join [glob -type d "\${homedir}/lib/j?re*"]]**.

Log, Monitor, Process Manager, and Memory

For information on configuring the log, monitor, Process Manager, and memory parameters, see [Chapter 11](#) , "Performance Monitoring and Process Management."

Filter Modules

Filter modules perform actions on topics and routes. Aside from certain filter modules that come with LiveServer (which are documented in the *KnowNow LiveServer Developer's Guide*), if you want to perform filtering operations, you will need to write your own custom filter modules to perform the actions you desire.

Your module can read configuration settings from the LiveServer's configuration files, such as host names and login information for external systems the module will use. The LiveServer's API provides support for writing your own filter modules. For more information, see the *KnowNow LiveServer Developer's Guide*.

You can have more than one filter module. A filter module can be attached to one or more topics or routes; once attached, it filters events based on the topic or route's `kn_payload` content (using a regular expression that has been assigned to the filter). Parameters can be set differently for each topic or route that a filter is attached to.

Once written, filter modules are loaded into LiveServer using the KnowNow System Administration console (as described in this section), after which they can be configured, attached, and detached, as described under [“Topics and Filter Modules” on page 128](#). You can also unload a filter module from LiveServer, which then makes it unavailable for any LiveServer operations.

This section provides information on custom filter modules under the following headings:

- [“Loading Custom Filter Modules”](#)
- [“Unregistering Custom Filter Modules” on page 97](#)
- [“Restrictions on Using Filter Modules” on page 97](#)
- [“Filter Module Debugging Options” on page 98](#)

Loading Custom Filter Modules

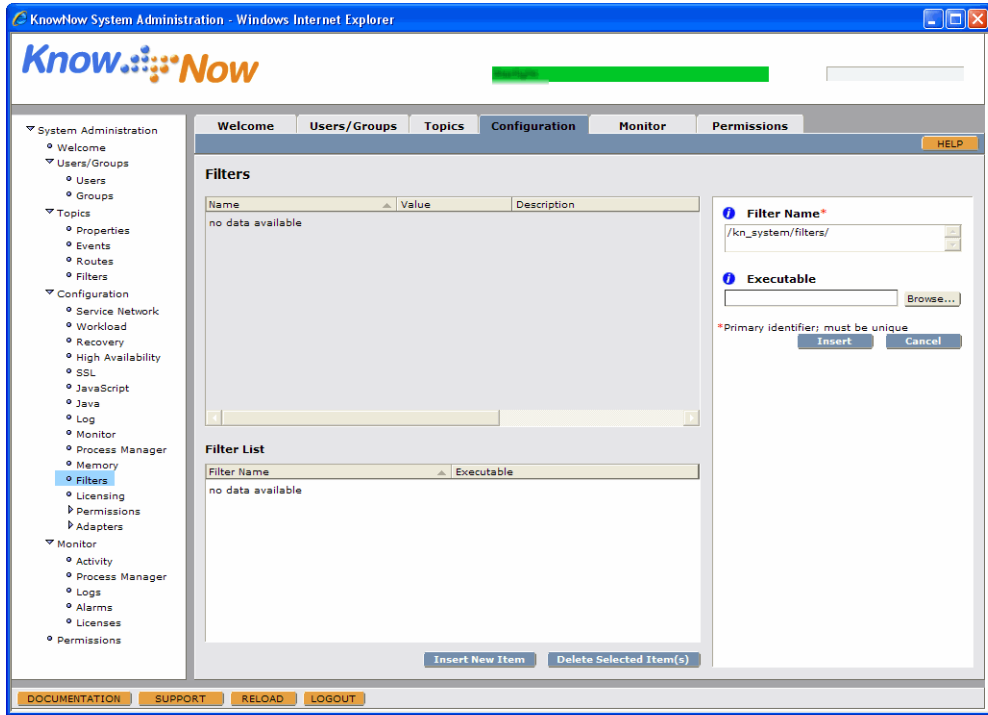
Once you have created one or more filter modules, you need to load them into LiveServer so they can be made available for use.

To load a filter module into LiveServer,

1. Access the KnowNow System Administration console.
2. Under **Configuration**, select **Filters**. The Filters panel is displayed.

3. Choose **Insert New Item**. To the right, two fields appear for specifying the filter name and for pointing to the location of the new filter.

Figure 4-12. Adding a new filter.



4. In the **Filter Name** field, enter the name of the new filter. The name must be in the form of a path and is required. The first part of the path, /kn_system/filters/ (which you must use), is already supplied for you.
5. In the **Executable** field, enter the path to the new filter, or use the **Browse** button to navigate to the location of the new filter. This information is also required.
6. Choose **Insert** to add the new filter. This command causes LiveServer to copy the file to the correct location in the LiveServer's installation directory, making the filter available for use as desired. The filter is added to the Filter List in the lower panel.

For information on attaching filters to topics and routes, see [Chapter 5 , "Managing Topics."](#) For information on configuring filters globally, see "Enabling Global Filter Properties" on page 132.

7. If you are done making changes, restart LiveServer.

Unregistering Custom Filter Modules

You can detach a filter from one or more topics or routes, or you can unregister it from LiveServer.

- If you **detach a filter from a topic or route**, it is still loaded into LiveServer and is still available to be attached to any topic or route again. Detaching a filter from a topic or route is described under [“Detaching a Filter From a Topic” on page 132](#).
- If you **unregister a filter** from LiveServer, the filter file itself still exists, but is no longer available for use anywhere in LiveServer. If you had created a route between two topics and had attached a filter module to that route, and then you later deleted that filter module from LiveServer, the created route will fail when it is next traversed. Of course, you can always re-load a filter at any time following the instructions under [“Loading Custom Filter Modules” on page 95](#), though you will then need to re-attach that filter module as desired.

To unregister a filter from LiveServer,

1. Access the KnowNow System Administration console.
2. Under **Configuration**, select **Filters**. The Filters panel is displayed.
3. In the Filter List, select the filter you wish to unregister. You can select more than one item; if you want to select discontinuous items, use Ctrl-click to select each one.
4. Choose **Delete Selected Item(s)**.

Restrictions on Using Filter Modules

Once a filter module is loaded into LiveServer, it can be added to any topics with the following restrictions:

- A filter module cannot be added to any topic containing /kn_subtopics.
- Filter modules can be attached to special topics, such as /kn_config and /kn_statistics, during the runtime of a LiveServer; however, the modules will not be persisted.

For more information on filters and topics, see [Chapter 5 , “Managing Topics.”](#)

Filter Module Debugging Options

To assist filter module developers, LiveServer comes with a built-in tool for verifying that reference counted objects (that is, anything inheriting from `RefCount::`, including events and topics) are correctly de-allocated by LiveServer. To use this tool, set the **Enable Leak Check** parameter (available under **Configuration > Log**) as follows:

- On all platforms, set **Enable Leak Check** to **1-Simple**. This will cause the tool to verify deletion upon LiveServer shutdown.
- On Windows platforms, you may set this value to **2-Expensive**. This will provide a complete stack trace for every increment and decrement call for any objects that were “leaked” at termination of the program, though with enormous performance penalties.

Configuring and Managing Licenses

There are several different types of LiveServer licenses available. Using the KnowNow System Administration console, you can specify where license files are stored as well as specify how to manage FLEXIm. You can also view your licenses.

You'll need to have a FLEXIm license manager available for LiveServer to contact. If you already have a FLEXIm license manager managing other licenses in your system, you can have it manage your LiveServer license as well. Otherwise, configure LiveServer to start the FLEXIm license manager from the KnowNow System Administration console's licensing screen as described here. If you have questions about the FLEXIm license manager or licensing that are not answered in this document, please contact KnowNow support.

If you wish to use your own licensing software, you may wish to disable the LiveServer's licensing software. That can be done using the Process Manager as described under ["Configuring and Using the Process Manager" on page 292](#).

This section covers licenses under the following headings:

- ["Configuring Licenses and Licensing"](#)
- ["Monitoring Licenses" on page 102](#)

Configuring Licenses and Licensing

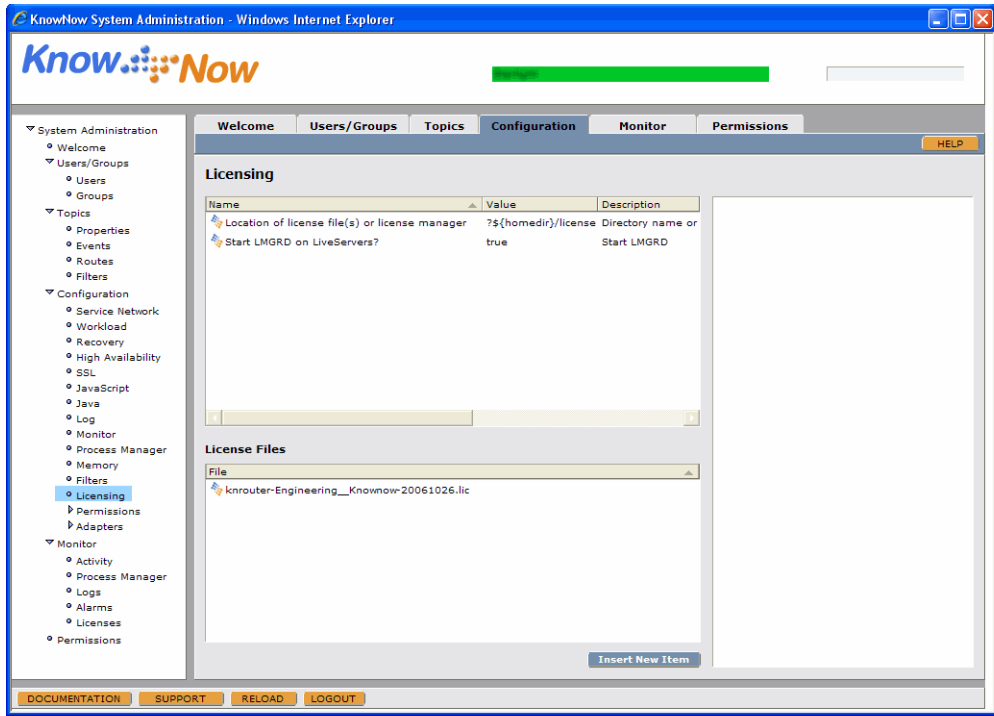
Configuring licenses and licensing is performed under the Configuration tab.

To configure your licenses,

1. Access the KnowNow System Administration console.

- Under **Configuration**, select **Licensing**. The Licensing panel is displayed.

Figure 4-13. The Licensing panel.



- For each parameter you wish to configure, select that parameter. A field appears to the right for modifying the value of that parameter. Modify the value of the parameter in the field to the right, then choose **Update**. The meanings and values of the parameters are provided under “[License Parameters](#)” on page 100.
- To insert new licenses, select **Insert New Item** at the bottom of the window. A **File** field appears to the right. Enter a file name (with path), or browse to the selected file, then choose **Insert**. The new license will appear in the list in the License Files panel.
- If you are done making changes, restart LiveServer.

License Parameters

This section describes the licensing parameters. For additional information about license managers, see “[Configuring and Using the Process Manager](#)” on page 292.

Location of license file(s) or license manager

This parameter specifies the directory in which you will store your license files. You can use a path or topic name. The default is a directory named `/license/` under your LiveServer installation directory (specified by `?${homedir}/license`).

You can set this parameter to one of the following values:

- The directory where the license files for LiveServer are physically located. This is the default. This is the only option if you chosen to have LiveServer start the FLEXlm license manager.
- **port@host** (of the FLEXlm license manager; this should be same as the license file's SERVER line) **or** **@host** if the FLEXlm license manager is on its default port.
- **@localhost** if the license manager is running on this host.
- The actual text of the license file, with **START_LICENSE\n** as the prefix and **\nEND_LICENSE** as the suffix.

The default FLEXlm port number is 27000.

Start LMGRD on LiveServers?

The options for this parameter are **true** or **false**, with the default value being **true**. LiveServer is licensed using Macrovision's FLEXlm. FLEXlm gives you many options for deployment.

- If LiveServer is the only product you license that uses FLEXlm, or if you want to manage your LiveServer licenses separately from your other FLEXlm-licensed products, leave this value **true**.
- On the other hand, if your company centralizes FLEXlm licenses in a single license management server (or cluster), you'll need to refer to your FLEXlm documentation to have that license manager manage the LiveServer licenses (and set this option to **false**).

Monitoring Licenses

License monitoring is accessed using the Monitor tab. Here you can view licenses for all nodes and for each specific node. For each license for all nodes, you can view the total number of licenses in use, the total number of floating licenses in use, the number of licenses available, and the overdraft. For individual nodes, you can view the actual number of licenses of each feature being used by that node as against the totals in the main license monitoring screen, which indicate the licenses checked out by all nodes from the license manager that this node is talking to.

To monitor how your licenses are being used,

1. Access the KnowNow System Administration console.
2. Under **Monitor**, select **Licenses**. The License panel is displayed.

Figure 4-14. The Licenses panel.

The screenshot shows the KnowNow System Administration console in a Windows Internet Explorer browser. The 'Monitor' tab is selected, and the 'Licenses' panel is displayed. The panel is divided into two sections: 'Licenses - All Nodes' and 'Licenses - Specific'.

Licenses - All Nodes

License Name	Total Lic in Use	Floating Lic in Use	Number of Licenses	Overdraft
kn_license_conn-KNLanguage	uncounted	uncounted	uncounted	uncounted
kn_license_conn-LiveBrowser	uncounted	uncounted	uncounted	uncounted
kn_license_conn-Other	uncounted	uncounted	uncounted	uncounted
kn_license_module-knperm	uncounted	uncounted	uncounted	uncounted
kn_license_module-knpersistence	uncounted	uncounted	uncounted	uncounted
kn_license_module-nskn	uncounted	uncounted	uncounted	uncounted
kn_license_node-CPU	uncounted	uncounted	uncounted	uncounted
kn_license_node-NIC	uncounted	uncounted	uncounted	uncounted
kn_license_node-RAM	uncounted	uncounted	uncounted	uncounted
kn_license_node-VM	uncounted	uncounted	uncounted	uncounted

Licenses - Specific (Starlight)

Statistic Name	Value
kn_license_conn-KNLanguage	7
kn_license_conn-LiveBrowser	1
kn_license_conn-Other	1
kn_license_module-knperm	1
kn_license_module-knpersistence	1
kn_license_module-nskn	1
kn_license_node-CPU	2
kn_license_node-NIC	2
kn_license_node-RAM	16
kn_license_node-VM	16

3. To view the information for a specific node, choose the node in the drop-down box next to the **Licenses - Specific** label. The information for that node is shown.

Permissions (Flat Files, LDAP)

For information on configuring permissions, see [Chapter 6](#), "Selecting a Permissions System."

Configuring LiveAdapters

KnowNow's LiveAdapters perform specific types of tasks. For example, the LiveAdapter for Microsoft Exchange, which is incorporated with KnowNow's Enterprise Syndication Solution (ESS) performs tasks making it possible to exchange information with Microsoft Exchange and to receive alerts based on items originating in Microsoft Exchange. And the LiveAdapter for RDBMS makes it possible to use ESS to generate business methods as events, including as RSS events. The LiveAdapter for Sphere makes it possible to create Sphere search channels in ESS.

Before using LiveAdapters for the first time, you *may* need to configure some parameters using the KnowNow System Administration console, though in most cases you don't need to do so (because the default settings enable and configure the LiveAdapters in ways that suit most needs).

However, if you are creating a LiveAdapters for RDBMS cluster (as described under "[LiveAdapters Clusters](#)" on page 282), you will need to set at least the **Shared Filesystem Path** parameter (as described under [step 3 on page 105](#)).

In addition, one important task is to use the KnowNow System Administration console to set permissions for LiveAdapters Framework and LiveAdapter for RDBMS users and for the topics used by those LiveAdapters.

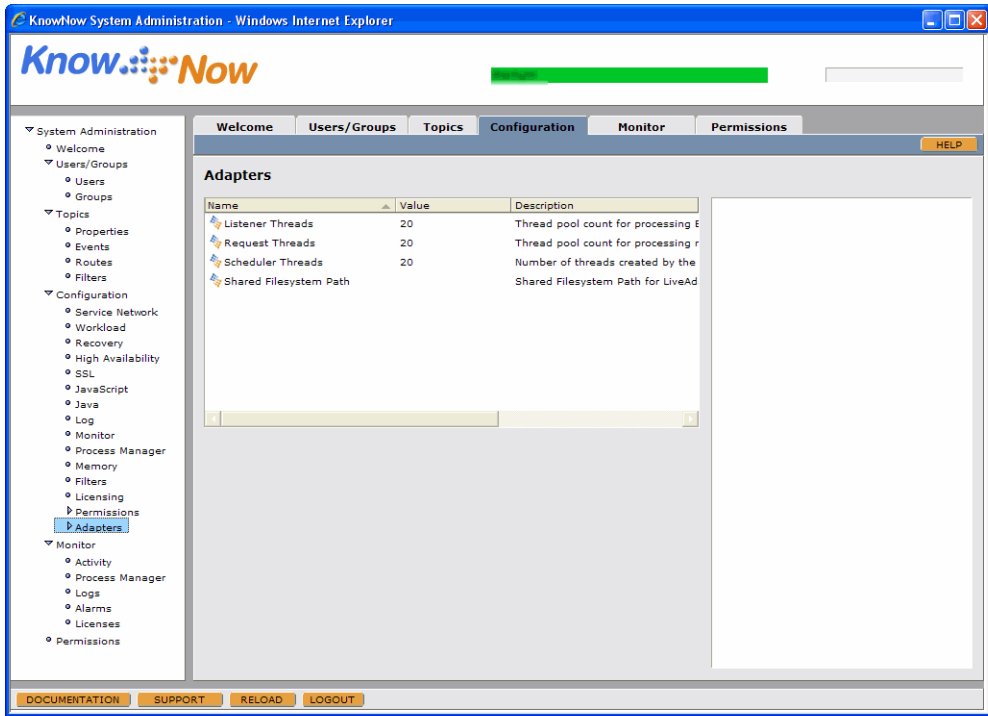
For additional information on LiveAdapters, see the *KnowNow LiveAdapters User's Guide* or the *KnowNow Enterprise Syndication Solution Administration and Developer's Guide*.

To configure LiveAdapters parameters,

1. Access the KnowNow System Administration console.

2. Under **Configuration**, select **Adapters**. The Adapters panel is displayed.

Figure 4-15. Configuring LiveAdapters.

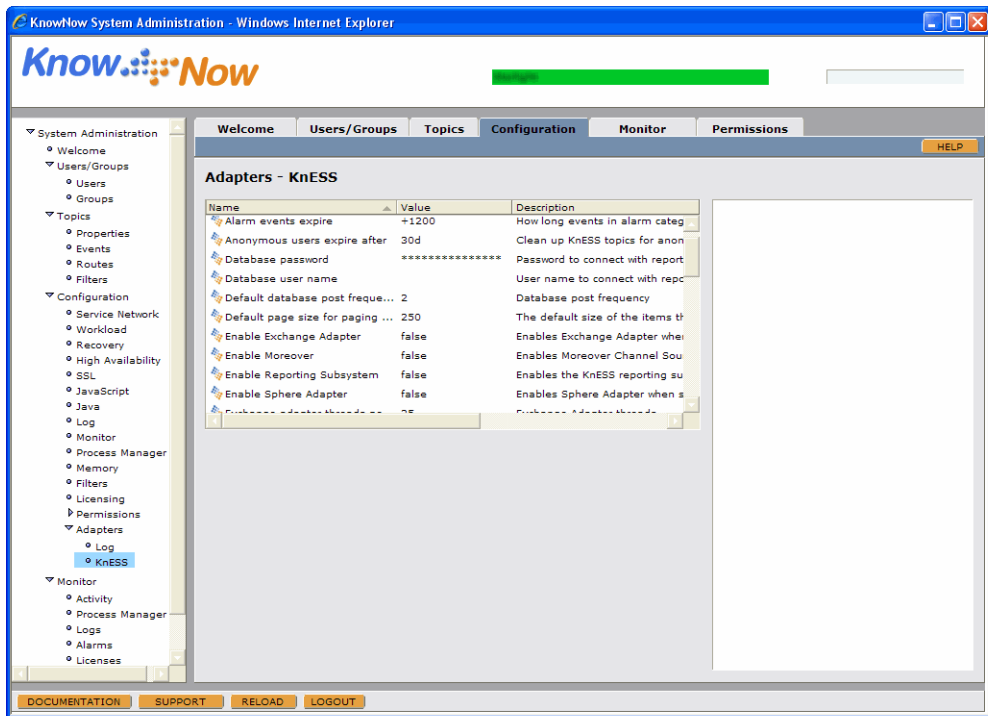


3. Configure the LiveAdapter for RDBMS parameters as follows:
 - **Listener Threads.** This specifies the thread pool count for pushed EIS events. The default value is **20**.
 - **Request Threads.** This specifies the number of dispatcher threads to be used for servicing inbound business methods (in both inbound modes). The default value is **20**.
 - **Scheduler Threads.** This specifies the number of dispatcher threads to be used for servicing scheduled methods. The default value is **20**.
 - **Shared Filesystem Path.** This specifies the path for a shared directory for a LiveAdapters cluster. For information on creating a LiveAdapters cluster, see [“LiveAdapters Clusters” on page 282](#). Provide the path without any trailing slashes. For example, If you create **k:** as mapped directory, then you need to set this parameter to **k:** only, not to **k:**. **For Windows 2003**, set this parameter

as `//machine_name/share_folder`. **Even though this is Windows, use forward slashes in the path.**

4. Next, navigate to **Configuration > Adapters > Logs**. Configure each of the parameters in this panel as follows:
 - **Debug**. Specifies whether to turn on logging for LiveAdapters. If set to **true**, then log entries for LiveAdapters are sent to the LiveServer’s log file. Normally, this value should be set to **false**.
 - **Log all request headers**. Specifies whether to log all the headers for every request. The default value is **false**.
5. Navigate to **Configuration > Adapters > KnESS** to configure ESS, the LiveAdapter for Microsoft Exchange, and the LiveAdapter for Sphere. For each parameter you wish to configure, select that parameter. A field appears to the right for modifying the value of that parameter. Modify the value of the parameter in the field to the right, then choose **Update**. The meanings and values of the parameters are provided under “KnESS Parameters” on page 107.

Figure 4-16. Configuring ESS.



6. In the **Permissions** panel (and possibly using the **User/Groups** panel), set permissions for LiveAdapters for RDBMS users and groups as needed. When you install LiveAdapters FrameWork and LiveAdapters for RDBMS, two LiveServer groups are created. Access rights for individual users can be configured by simply creating users in LiveServer and adding those users to one of these two groups. For information on how to do so, see [Chapter 9, "Managing Users and Groups."](#) The two groups are as follows:
 - **LiveAdapters Business Analyst.** These users have access to the Business Method Management tab.
 - **LiveAdapters Admin.** These users have access to both the Business Method Management tab and the Setup tab.
7. Set permissions for LiveAdapters for RDBMS topics by creating access patterns and assigning users and groups to those patterns as described in [Chapter 8, "Managing Permissions."](#)
8. If you are done making changes, restart LiveServer.

KnESS Parameters

The **Adapters > KnESS** configuration parameters control a number of aspects of KnowNow's Enterprise Syndication Solution (ESS). For more information on ESS, including information on using the LiveAdapter for Microsoft Exchange and the LiveAdapter for Sphere, see the *KnowNow Enterprise Syndication Solution Administration and Developer's Guide*.

Alarm events expire

This specifies how long events that have been specified as alerts will last. The default value is **+1200** (1200 seconds after the alert was received).

Anonymous users expire after

This specifies the length of time of inactivity in an anonymous user's topics, after which those topics are deleted. The default value is **30d** (30 days).

Database password

Password for the user who accesses the ESS Reporting database.

Database user name

Login name of the user who accesses the ESS Reporting database.

Default database post frequency

The default value for how frequently the ESS Reporting statistics are posted to the Reporting database. The default value is **2** (hours). ESS Reporting needs to be enabled for this parameter to have any effect. This value can be configured here or through the ESS Administration console when configuring ESS Reporting.

Default page size for paging support

The default size of the items that will be fetched in one iteration. The default value is **250**.

Enable Exchange Adapter

Enables the LiveAdapter for Microsoft Exchange, so that KnowNow's ESS can be used to create channels that exchange information with Microsoft Exchange. The default value is **true**.

Enable Moreover

Enables the ability to configure and create Moreover channels in ESS. The default value is **true**. For more information on Moreover, see the *KnowNow Enterprise Syndication Solution Administration and Developer's Guide*.

Enable Reporting Subsystem

When set to **true**, the ESS Reporting subsystem is enabled. The default value is **false**.

Enable Sphere Adapter

Enables the ability to configure and create Sphere channels in ESS. The default value is **true**. For more information on Sphere, see the *KnowNow Enterprise Syndication Solution Administration and Developer's Guide*.

Exchange adapter threads per processor

This specifies the number of threads per processor for the LiveAdapter for Microsoft Exchange. The default value is **25**.

Exchange Adapter UDP Port for receiving updates

This specifies the port on which the LiveAdapter for Microsoft Exchange will be listening for updates from the Microsoft Exchange server. The default value is **3501**.

Exchange Server Domain Name

This specifies the domain name for the Microsoft Exchange server. The default value is **KnowNow**.

Exchange Server Host Name

This specifies the host name for the Microsoft Exchange server. The default value is **exchange.knownow.com**.

Exchange Server Password

This specifies the password for the Microsoft Exchange server. The user (signin account) is specified using the [Exchange Server Sign-in Account](#).

Exchange Server Port

This specifies the port number for the Microsoft Exchange server. The default value is **443**.

Exchange Server Protocol

This specifies the protocol (HTTP or HTTPS) used by the Microsoft Exchange server. The default value is **https**.

Exchange Server Public folder Base URI

This specifies the base URI for the Microsoft Exchange server's Public folder. The default value is **/public**.

Exchange Server Sign-in Account

This specifies the sign-in account (user) for the Microsoft Exchange server. The password is specified using the [Exchange Server Password](#).

Feed threads per processor

This specifies the number of feed threads to allow per processor. The default value is **25**.

JDBC connection URL

URL to use for connecting to the ESS Reporting database. This value can also be set using the ESS Administration console when configuring ESS Reporting.

JDBC Driver name

Name of the JDBC driver for the ESS Reporting database. This value can also be set using the ESS Administration console when configuring ESS Reporting.

List Management Debug

This enables or disables debugging for summary event formation. The default value is **false**.

Log Level for Exchange Adapter

This enables or disables reporting of advanced support diagnostics for the LiveAdapter for Microsoft Exchange. The default value is **false**.

Log Level for LiveAdapter for Reporting

This enables or disables reporting of advanced support diagnostics for the LiveAdapter for Reporting. The default value is **false**.

Log Level for LiveAdapter for RSS

This enables or disables reporting of advanced support diagnostics for the LiveAdapter for RSS. The default value is **false**.

Log Level for LiveAdapter for Sphere

This enables or disables reporting of advanced support diagnostics for the LiveAdapter for Sphere. The default value is **false**.

Login Style

This specifies the style of login for ESS: **Enterprise Syndication Solution** (the default value) or **Web Syndication Solution**.

Login user expire after

This specifies the length of time to pass after which the ESS user's login expires and they must log in again. The default value is **infinity**. For information on specifying units in parameters, see ["Entering Data and Editing LiveServer Parameters"](#) on page 53.

Moreover Metabase URL

This specifies the metabase URL for Moreover configuration. The default value is **http://p.moreover.com/cgi-local/page?client_id=KnowNow**.

Partition Period for Reporting Database

Specifies the default value for the partition period for ESS database partitioning. This value can be changed through the ESS Administration console.

RDBMS Used

The type of the RDBMS to be used for ESS Reporting. This value can be configured here or through the ESS Administration console when configuring ESS Reporting.

Reporting adapter threads per processor

Reporting adapter threads per processor

This specifies the number of threads per processor for ESS Reporting. The default value is **25**.

Sphere adapter threads per processor

This specifies the number of threads per processor for the LiveAdapter for Sphere. The default value is **25**.

Sphere Customer ID

This specifies the ID for the Sphere customer; for example, **example.com**.

Sphere Partner ID

This specifies the ID for the Sphere partner; for example, **example**.

Sphere Server URL

This specifies the URL for Sphere; for example, **www.sphere.com/api/query**.

Sphere URL for Find Related Articles

This specifies the URL for Sphere to use for the “find related articles” option when creating Sphere channels; for example, **www.sphere.com/sphereit/**.

Sphere URL for Search Histogram

This specifies the URL for Sphere to use for the search results histogram; for example, **www.sphere.com/hdsproxy**.

Threshold for feedwatchdog warnings

This specifies the maximum percentage difference between number of refresh events and the number of Feedwatcher command events before a warning is generated. The default value is **10** (10%).

Chapter 5 Managing Topics

Using the Topics tab in the LiveServer's KnowNow System Administration console, you can view, edit, delete, and create topics, routes, and events. You can also manage topics, events, routes, access control, and add or remove filter modules to or from topics and routes. All of these capabilities can also be performed using `do_method` commands. For more information on `do_method` commands, see the *KnowNow LiveServer Developer's Guide*.

In this chapter, you will find the following information about topics:

- [“Understanding and Managing Topic Space” on page 114](#)
- [“Navigating Topic Space” on page 115](#)
- [“Working with Topics” on page 117](#)
- [“Topics and Topic Properties” on page 119](#)
- [“Topics and Filter Modules” on page 128](#)
- [“Working With Events and Event Properties” on page 133](#)
- [“Managing Routes and Route Properties” on page 140](#)

Understanding and Managing Topic Space

The hierarchical structure of topics within a KnowNow LiveServer is referred to as topic space. Using the Topics tab in the KnowNow System Administration console, you can navigate topic space and examine topic properties and events. For an overview of what topics are, see [“Topics” on page 6](#). You can also find additional information on topics and events in the *KnowNow LiveServer Developer’s Guide*.

Best Practices in Creating Topics

KnowNow follows certain rules for creating and managing topics with applications. These rules include the following:

1. Topics are easy to create and do not cost much more to create than to post an event. They come and go as needed. Because of this, you can, for example, create topics for a short amount of time to use for one specific purpose. Therefore, we encourage you to feel free to create topics as needed, even if you only need them briefly.
2. Understand how to create a temporary topic. Temporary topics can be very useful. For information on creating temporary topics, see [“Temporary Topic Expires” on page 315](#) and [“Temporary Topic Recognize” on page 316](#).
3. You can use topics to logically group events and to keep the topic tree clean. Take a look at how KnowNow organizes and names topics for some ideas. For example, you can use a topic tree to contain all your system topics/events, then set the permissions on that tree to protect that entire set of topics and subtopics.
4. Topics have a number of properties, such as atomic, that you can set. These properties enable you to get special behaviors out of topics. The definition and use of these properties are discussed in this chapter. Familiarizing yourself with these properties can help you take advantage of the power and flexibility of topics.

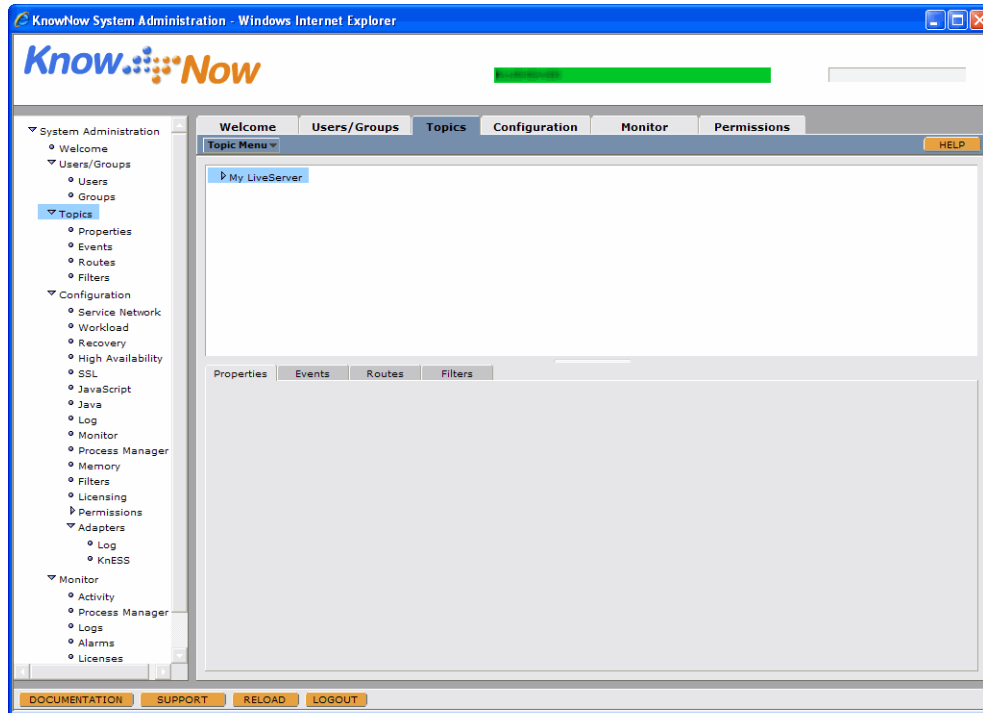
Events

One of the basic building blocks in the KnowNow LiveServer platform is the event. Events essentially consist of a name and a value. LiveServer comes with a large number of events you can put to good use. For example, an event may be named `kn_from` and may contain the topic from which that event was routed. The *KnowNow LiveServer Developer’s Guide* describes events in detail and lists and defines each of the events that are provided with LiveServer. See that book for more information.

Navigating Topic Space

You can access the Topics tab by opening LiveServer in a Web browser and clicking on the Topics tab.

Figure 5-1. The Topics tab.



The list of topics on LiveServer is displayed in the top panel of the Topics tab. This list represents the topic space of a LiveServer as an expandable tree of nodes, with LiveServer as the topmost node.

- To select a topic, choose the topic name in the tree (in the top panel). The topic is highlighted in the tree, indicating it is the currently selected topic.
- To switch to a different topic, select the name of the new topic. Only one topic can be selected at a time.
- To display a topic's subtopics, click on the topic's arrow. To hide the subtopics, click on the topic's arrow again.

The lower panel in the Topics tab displays a set of tabs providing different views of topics. These tabs are described starting on [page 117](#).

Usable /kn_system Subtopics

You are free to subscribe to any of the subtopics under /kn_system. For example, you can subscribe to the /heartbeat topic to receive an event every ten seconds, or to the /filters topic to collect your filters, or to the /permissions topic to use events to set permissions. However, be wary of posting, deleting, or performing other such operations on subtopics under /kn_system; the majority of those topics should not be written to.

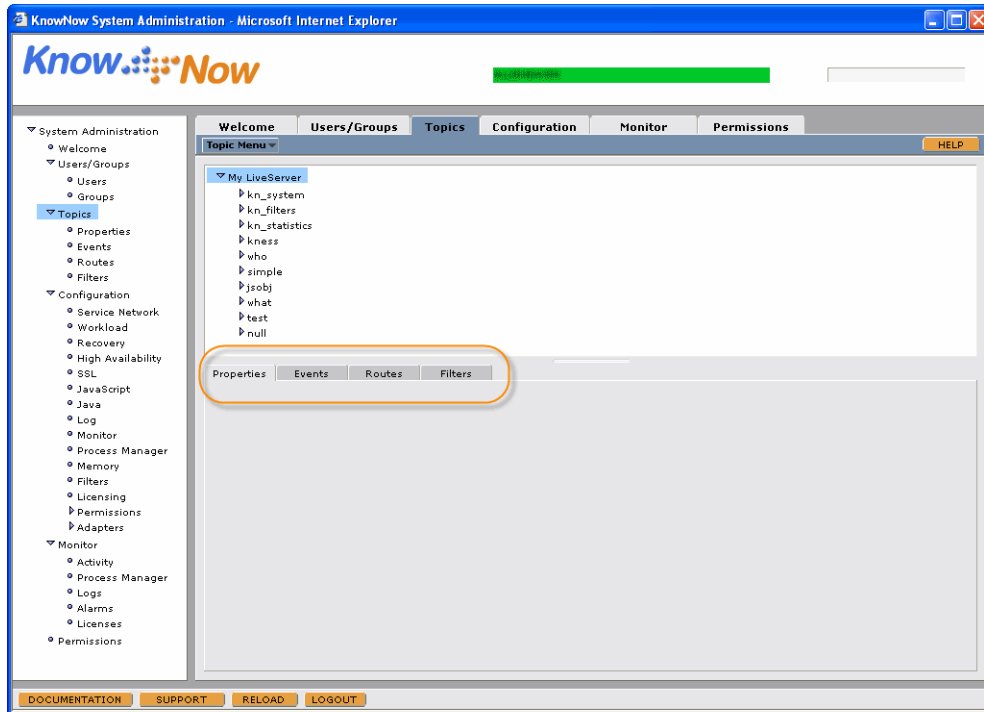
Working with Topics

The Topics tab provides four modes for viewing and working with topics, accessed by choosing tabs in the lower pane of the Topics page:

- **Properties.** Use the functions on the Properties tab to view, edit, and delete the selected topic and its properties. The Properties tab is discussed starting on [page 119](#).
- **Events.** Use the functions on the Events tab to view, edit, and delete events within the selected topic. The Events tab is discussed starting on [page 133](#).
- **Routes.** Use the functions on the Routes tab to manage the routes from this topic to other locations. You also use the Routes tab to manage route filters. The Routes tab is discussed starting on [page 140](#).
- **Filters.** Use the functions on the Filters tab to manage topic filters. The Filters tab is discussed starting on [page 128](#).

These modes operate in the context of the currently selected topic. To access one of these modes, select a topic in the upper panel, then select the desired tab in the lower panel. For three of these modes, there is also a Property Editor available for editing the characteristics of the properties, events, and routes. A small **f** next to a topic indicates that a filter is attached to that topic.

Figure 5-2. Topic modes.



Topics and Topic Properties

Using the functions available on the Properties tab, you can view, edit, and delete topic properties. You can also create new subtopics and delete topics or subtopics. More information on performing these actions can be found under the following headings:

- [“Viewing Topic Properties”](#)
- [“The Property Editor” on page 121](#)
- [“Editing a Topic Property” on page 124](#)
- [“Adding a New Topic Property” on page 124](#)
- [“Deleting a Topic Property” on page 125](#)
- [“Creating a Subtopic” on page 125](#)
- [“Deleting a Topic or Subtopic” on page 127](#)

If you wish to restrict access to a topic, assign the topic to an access pattern, then assign that access pattern to users or groups. For more information, see [Chapter 8 , “Managing Permissions.”](#)

Viewing Topic Properties

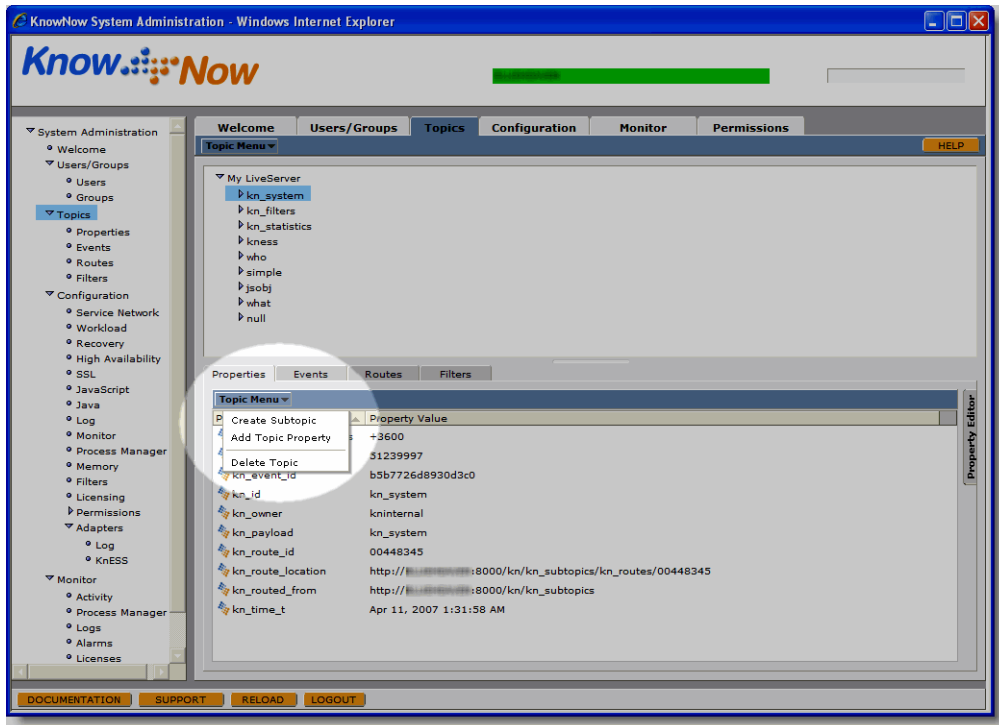
The properties of a topic can be viewed and managed using the Properties tab. The Properties tab displays the names and values for each property associated with the selected topic.

To view the properties of a topic,

1. Access the KnowNow System Administration console.
2. Select the Topics tab.
3. In the top panel, navigate to the desired topic.

- In the bottom panel, select the Properties tab. The properties of the selected topic are displayed. This list of properties is also called the properties table.

Figure 5-3. Topic properties.



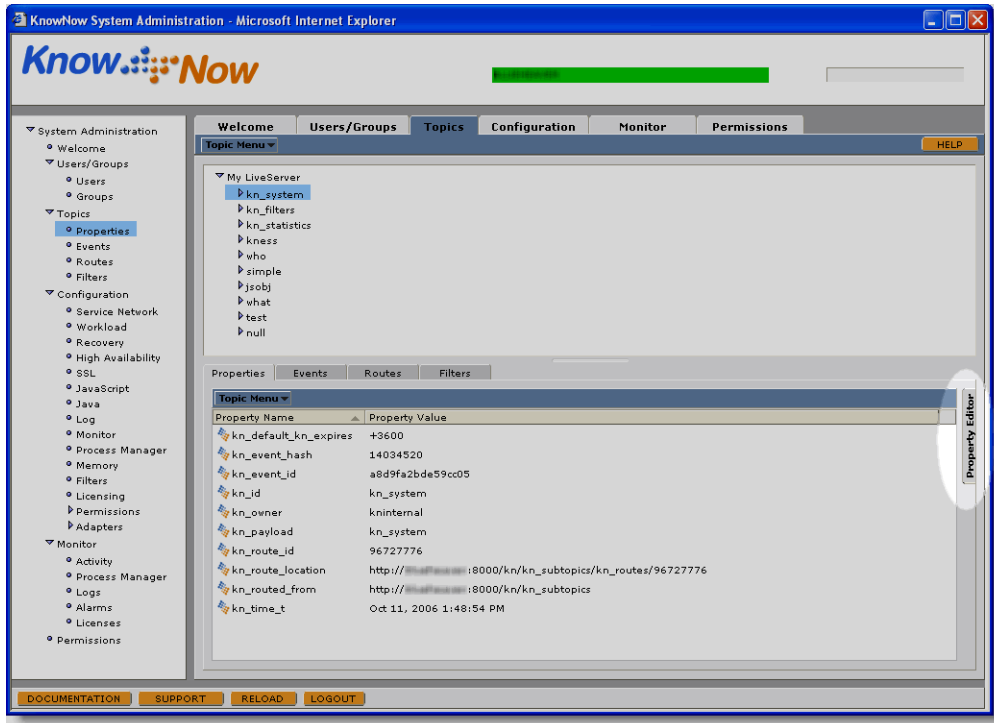
A **Topic Menu** on the left in the lower panel provides access to additional capabilities:

- **Create Subtopic**, which opens a dialog box for creating a new subtopic. Creating a subtopic is described under [“Creating a Subtopic”](#) on page 125.
- **Add Topic Property**, which opens the Property Editor. Adding a topic property is described under [“Adding a New Topic Property”](#) on page 124.
- **Delete Topic**, which deletes the entire topic. Deleting a topic is described under [“Deleting a Topic or Subtopic”](#) on page 127.

The Property Editor

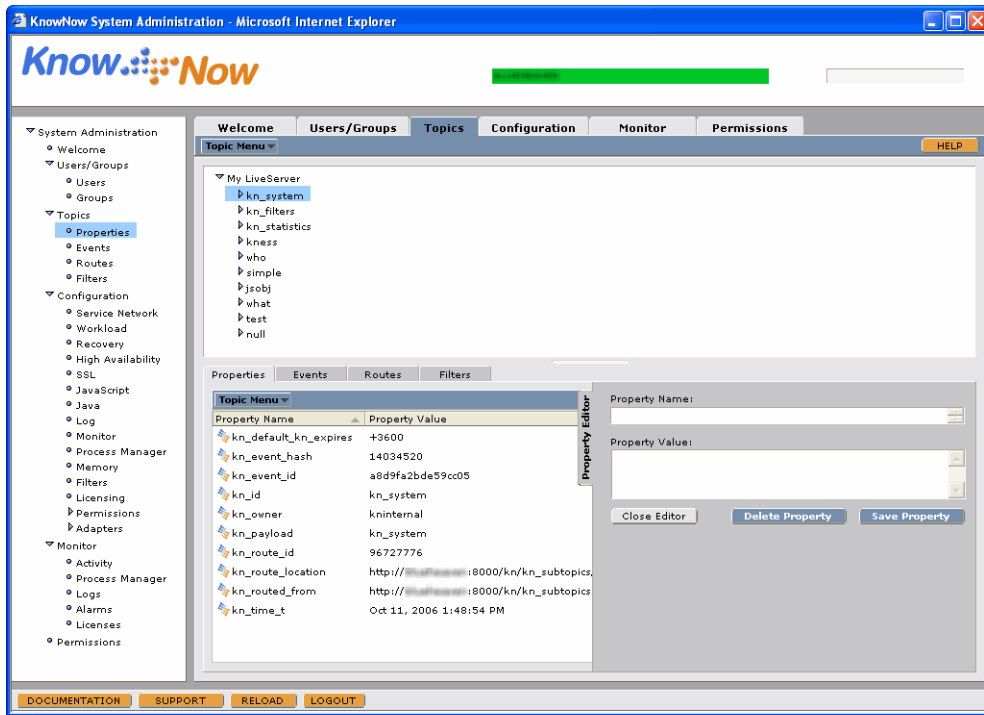
To the right of the lower panel is a tab labeled **Property Editor**. The Property Editor can be used to edit topic, event, and route properties.

Figure 5-4. Topic properties editor.



Clicking on that tab expands the editor so you can modify the characteristics of a topic, event, or route's properties, add a new property, or delete a property.

Figure 5-5. The Property Editor.



The Property Editor has the following fields:

- **Property Name**, which displays the name of a selected property, or in which you can enter the name of a new property.
- **Property Value**, which displays the value for a selected property, or in which you can enter the value for a new property.

Topic Properties and Topic Behaviors

As discussed under “Topics” on page 6, there are different types of topics: Topics, sub-topics, routes, and journals. Some topics have special behaviors which are controlled by setting their properties. Those topic properties can be set in a number of different ways.

- Some properties are set by using the `do_method=set_topic_property`. For information on `do_methods`, see the *KnowNow LiveServer Developer's Guide*.

- Some properties are set by using header/value pairs and using one of the set of KnowNow-specific headers whose names start with kn_ (such as kn_atomic). The kn_ headers are also described in the *KnowNow LiveServer Developer's Guide*; again, see that guide for more information. You can either send these headers as commands from your program or set them using the Property Editor.
- Your custom properties can be set using either your own custom headers or using the Property Editor.

A normal named topic is not temporary; the topic itself will last until its expiration time.

There are two types of topic behaviors:

- Journals. Think of journals as sessions. They last a certain amount of time. If you disconnect and reconnect and the session still exists (i.e., the session has not timed out), you can recover the state of your session.
- Request/reply (also called request/response). LiveServer supports two kinds of request/responses:
 - A LiveBrowser form called LWWS. With LWWS, if you have some rendezvous topic with one subscriber to that topic (you could have more than one, but they must obey certain rules), there can be any number of posters to that topic. When a publisher (which is called a client in this pattern) posts a service request to that topic, the client is required to do two things: it must have a reply-to header, which must point to some topic specific to that client. The client publishes that request to the well-known service. The service receives the request, performs whatever action is required by that request, then posts a reply to the client's reply-to topic. If the client provides a reply URI, the service is required to stuff that value into kn_route_location and kn_uri. For more information on LWWS, see the *KnowNow LiveBrowser Developer's Guide*.
 - Another form that is very similar to the LiveBrowser form, except that this model has some additional features. For example, it uses different headers than the LWWS method. In addition, it has two kinds of subscribers: a manager (of which there can be only one), and providers, of which there can be one or more. The manager decides which provider should handle any given request. This model can guarantee that the event got delivered to only one provider, so there is no possibility of a duplicate service responding to a request. (Though of course, duplicate squashing will take care of duplicate events.) For more on request/response and duplicate event squashing, see the *KnowNow LiveServer Developer's Guide*.

Editing a Topic Property

You can use the Property Editor to change the values of a topic property or to add or delete a topic property.



Caution: Editing and adding topic properties can have unforeseen consequences. For example, editing the value of the `kn_id` property will create a new topic with the new `kn_id` value and leave the original topic intact. Other properties, such as `kn_route_location`, are used by the Connectors and should not be edited without a full understanding of how they operate. Therefore, it is recommended that you use caution when editing and adding topic properties.

To edit an existing topic property,

1. Access the KnowNow System Administration console.
2. Select the Topics tab.
3. In the top panel, navigate to the desired topic.
4. In the bottom panel, select a topic property in the Properties tab and click on **Property Editor** to expand the editor.
5. Edit the name and value of the property as needed.
6. Choose **Save Property** to save your changes. The properties table is updated with the new value.

Adding a New Topic Property

You have the option of adding a new property to the selected topic. Before proceeding, be sure to read the caution (under [“Editing a Topic Property” on page 124](#)) regarding editing or adding properties. You can add any valid `kn_` header topic property using the Property Editor; for a description of the `kn_` headers, see the *KnowNow LiveServer Developer’s Guide*.

To add a new topic property,

1. Select the Topics tab.
2. Make sure you do not have a topic property already selected. If you do, navigate away from the topic and then back again to clear the topic property selection.

3. Click on **Property Editor** to expand the editor or choose **Topic Menu > Add Topic Property**, which also opens the editor.
4. In the Property Editor, enter the name and value of the new property as needed.
5. Choose **Save Property**. The properties table is updated with the new property.

Deleting a Topic Property

Before deleting a topic's property, make sure that you understand what the consequences of deleting that property will be.

To delete a topic property,

1. Access the KnowNow System Administration console.
2. Select the Topics tab.
3. In the top panel, navigate to the desired topic.
4. In the bottom panel, select a topic property in the Properties tab and click on **Property Editor** to expand the editor.
5. Choose **Delete Property**. The properties table is updated with the new property.

Creating a Subtopic

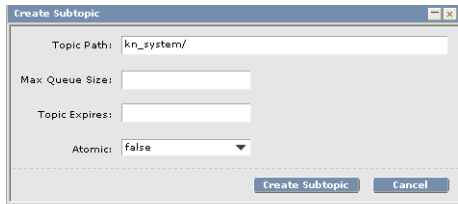
Use the Properties tab to create a subtopic under the currently selected topic.

To create a subtopic,

1. Access the KnowNow System Administration console.
2. Select the Topics tab.
3. In the top panel, navigate to the desired topic.

4. In the bottom panel, in the Properties tab, choose **Topic Menu > Create Subtopic**. The **Create Subtopic** dialog box appears.

Figure 5-6. Creating a subtopic.



5. In the **Topic Path** field, enter the name of the subtopic you want to create. The name of the selected topic will already be filled in as part of that topic name.
6. If desired, enter a value for the maximum queue size for the new subtopic in the **Max Queue Size** field. This value must be an integer. If you do not wish to specify a maximum queue size, leave this field blank. This value is associated with the header `kn_max_queue_size`. For more information on this header, see the *KnowNow LiveServer Developer's Guide*.
7. Enter a value for topic expiration in the **Topic Expires** field. By default, the expiration value for the new subtopic is **infinity**, meaning the topic never expires.
8. In the **Atomic** field, choose **True** or **False**. When **Atomic** is **True**, LiveServer will **not** send status events on an event sent to that topic until the event is on disk and recoverable. So, for example, if you receive a post when **Atomic** is set to **True**, then you can be assured that the event was routed properly and stored on disk. This state is also called “durable” in some systems. For more information on this setting, see the information on the `kn_atomic` header in the *KnowNow LiveServer Developer's Guide*.
9. Choose **Create Subtopic** to create the new subtopic. The new subtopic is added to the topic tree under the currently selected topic. (Or choose **Cancel** to cancel the creation of the subtopic.)

Deleting a Topic or Subtopic

Before deleting a topic or subtopic, make sure that you understand what the consequences of deleting that topic will be.



Caution: There is no way to undo the deletion of a topic, so please use caution when deleting topics. Deleting a topic also deletes all events and subtopics within that topic. If you want the topic again, you will need to re-create it, then post new events to it.

To delete a topic,

1. Select the Topics tab.
2. Select the topic you wish to delete.
3. In the Properties tab, choose **Topic Menu > Delete Topic**. A confirmation dialog box appears, asking if you are sure you want to delete the selected topic. Choose **OK** in the confirmation dialog box to complete deletion of the topic, or choose **Cancel** to leave the topic intact.

Topics and Filter Modules

Filter modules are installed on LiveServer to provide event filtering capabilities for topics and routes. LiveServer ships with some internal filter modules for its own purposes, as well as some useful filter modules that you can use. You can also write your own filter modules to perform whatever kinds of filtering tasks you wish on topics and routes. For more information on filter modules—what they are (including the KnowNow filter modules that you can use) and how to create them—see the *KnowNow LiveServer Developer's Guide*.

Filters must first be loaded into LiveServer in order to be available through the Topics tab. For more information on loading a custom filter, see [“Loading Custom Filter Modules” on page 95](#).

Instructions on working with topic filters are provided under the following headings. For information on route filters, see [“Attaching a Filter to a Route” on page 146](#) and [“Detaching a Filter from a Route” on page 147](#).

- [“Viewing Filters on a Topic”](#)
- [“Attaching a Filter to a Topic” on page 130](#)
- [“Editing Filter Parameters” on page 131](#)
- [“Enabling Global Filter Properties” on page 132](#)
- [“Detaching a Filter From a Topic” on page 132](#)

Viewing Filters on a Topic

You can view the filter or filters associated with a topic.

To view the filters associated with a topic,

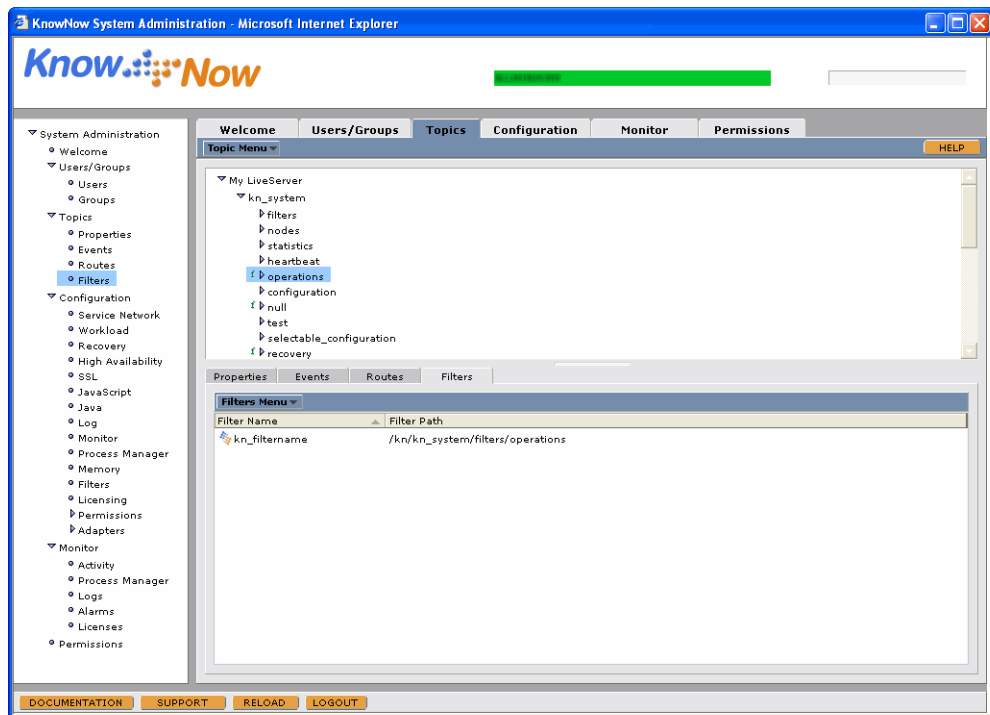
1. Access the KnowNow System Administration console.
2. Select the Topics tab.
3. In the top panel, navigate to the desired topic. The topic must have a small **f** to the left of its name (indicating that it has a filter attached).
4. Select the Filters tab. In the lower pane is a list of the filters attached to that topic and the parameter values for the attached filters. You can sort the list by clicking on the column headings. The columns are labeled as follows:
 - **Filter Name**, which contains the names of the filters and filter parameters.

- **Filter Path**, which contains the value of each filter item in **Filter Name**.

Above the list of filter parameters is a button labeled **Filters Menu** that provides the following commands:

- **Attach Filter**, which provides access to the capability of attaching a filter to the topic as described under “Attaching a Filter to a Topic” on page 130.
- **Update Filter Parameters**, which provides access to editing the filter parameters as described under “Editing Filter Parameters” on page 131.
- **Detach Filter**, which you use to detach a filter as described under “Detaching a Filter From a Topic” on page 132.

Figure 5-7. Viewing filters.



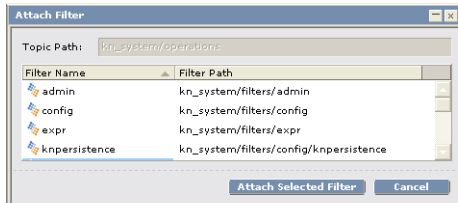
Attaching a Filter to a Topic

You can attach one or more filter modules to any given topic. This section tells how to attach a single filter module to a topic using the KnowNow System Administration console. You can also use headers to attach multiple filter modules (and assign parameters for those filter modules) as described in the *KnowNow LiveServer Developer's Guide*.

To attach a filter to a topic,

1. Access the KnowNow System Administration console.
2. Select the Topics tab.
3. In the top panel, navigate to the desired topic.
4. Select the Filters tab.
5. Choose **Filters Menu > Attach Filter**. The Attach Filter dialog box is displayed. The selected topic is displayed in the **Topic Path** field. In this dialog box, a list of filters appears. Not all the filters in this list are available for use; some are internal LiveServer filters. The LiveServer filters that you can use are documented in the *KnowNow LiveServer Developer's Guide*. You can also use any custom filters that have been uploaded using the procedure described under “[Loading Custom Filter Modules](#)” on page 95.

Figure 5-8. Attaching a filter.



6. Select the desired filter in the list and choose **Attach Selected Filter**. The filter is attached to the topic, and the filter parameters table is updated. Once a filter is attached to the topic, the filter properties are available for editing as described under “[Editing Filter Parameters](#)” on page 131.

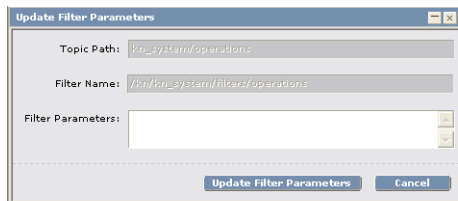
Editing Filter Parameters

You can edit many filter parameters, though not all filter parameters are editable. When you edit the parameters for a filter that is attached to a topic, you are editing those parameters for the instance of the filter that is attached to that topic only. Other topics that have an instance of the same filter attached can have different parameters.

To edit the parameters of a filter,

1. Access the KnowNow System Administration console.
2. Select the Topics tab.
3. In the top panel, navigate to the desired topic (it must have a filter attached).
4. Select the Filters tab.
5. In the lower pane, select a filter parameter that you want to edit.
6. Choose **Filters Menu > Update Filter Parameters**. The Update Filter Parameters dialog box is displayed. The topic path is displayed in the **Topic Path** field. The name of the filter is displayed in the **Filter Name** field. Neither of these two fields is editable. Finally, the parameters for this filter are displayed in the **Filter Parameters** field, which you can edit.

Figure 5-9. Editing a filter parameter's properties.



7. Edit the value of the parameter in the **Filter Parameters** field.
8. Choose **Update Filter Parameters** to save your changes. The filter parameters table is updated with the new value.

Enabling Global Filter Properties

You can upload your own custom filter modules as described under [“Loading Custom Filter Modules” on page 95](#). Once a module has been loaded, you may need to configure it. When a filter is attached to a topic, you normally set parameters for the instance of the filter that is attached to that topic. Another instance of the same filter module that is attached to a different topic might use different parameters.

However, some filters require global configuration. A typical way for such a filter (i.e., for a filter that requires global configuration) to get those configuration entries is to add those entries as properties on the filter’s own topic. To configure a filter globally, you can assign properties to the `/kn_system/filters/filtername` topic using the Property Editor as described under [“Adding a New Topic Property” on page 124](#).

Detaching a Filter From a Topic

You can detach a filter module from a topic. When you do so, you are not deleting that filter from the system, but are instead just detaching it from the specific topic. Other topics that have the same filter attached will still have that filter attached, and you can also attach that filter to other topics and routes. You can use the KnowNow System Administration console or headers to detach a filter from a topic. You can also edit a route or subscription’s properties to detach filters. The filter is detached and the filter properties table is updated in real time.



Caution: Detaching a filter can have unexpected results, and you will **not** be given a confirmation prompt for this action. Be sure you know what you are doing and what the effects will be before detaching a filter.

If you want to permanently remove a filter module from LiveServer, you will need to follow the instructions provided under [“Unregistering Custom Filter Modules” on page 97](#).

To detach a filter,

1. Access the KnowNow System Administration console.
2. Select the Topics tab.
3. In the top panel, navigate to the desired topic.
4. Select the Filters tab.
5. In the lower pane, choose **Filters Menu > Detach Filter**. The filter is detached from the topic.

Working With Events and Event Properties

Topics can contain events which can be viewed and edited using the Topics tab. Events are discussed in more detail in the *KnowNow LiveServer Developer's Guide*. Using the functions on the Topics tab, you can view, edit, and delete events and the properties of those events. The instructions for performing these actions are provided under the following headings:

- [“Viewing Events and Event Properties”](#)
- [“Sending Test Events and Deleting Events” on page 135](#)
- [“Editing, Adding, and Deleting Event Properties” on page 138](#)

Viewing Events and Event Properties

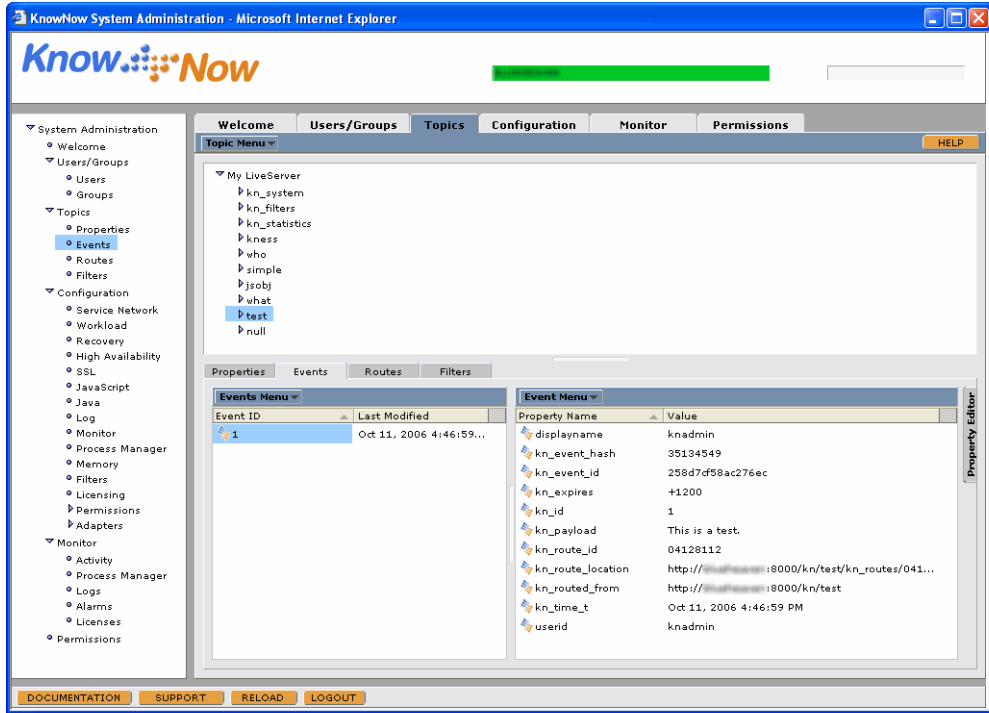
When viewing events in a topic or subtopic, you can also view the event's properties and can access to the Property Editor to edit those events and their properties.

To view events and event properties in a topic,

1. Access the KnowNow System Administration console.
2. Select the Topics tab.
3. In the top panel, navigate to the desired topic.
4. In the bottom panel, select the Events tab. Any events present within the topic are displayed. The table of events displays the following:
 - **Event ID**, which contains the event's unique ID.

- **Last Modified**, which shows the date and time when the event was last changed.

Figure 5-10. Viewing events.



5. The event table can be sorted by either the event ID or the event last modified date by clicking on the appropriate column head.



Note: The table of events will automatically update in real time when new events appear or are deleted from the selected topic. Be aware that there is no way to “lock” an event to prevent it from being edited, or to prevent more than one user from editing the same event simultaneously.

Above the event table is a button labeled **Events Menu**. This menu is for working with all events. It provides the commands **Send Test Event** and **Delete All Events**. For more on these commands, see “[Sending a Test Event](#)” on page 135 and “[Deleting All Events in a Topic](#)” on page 137.

6. To view the properties of a particular event, simply select an event. A table containing the properties of that event appears to the right. The event you selected for viewing is highlighted in the events table.
7. Above the table of the selected event properties is a button labeled **Event Menu**. This menu is for working with individual events. It provides the commands **Add Property to Selected Event** and **Delete Selected Event**. For more on these commands, see [“Deleting an Event” on page 137](#) and [“Adding Event Properties” on page 139](#).

The table of event properties is organized into two columns.

- **Property Name**, which lists the names of the event properties.
- **Value**, which shows the values in each event property.

For information on editing event properties, see [“Editing and Deleting Event Properties” on page 138](#).

Sending Test Events and Deleting Events

Using the Events tab, you can perform several actions on events as described under the following headings:

- [“Sending a Test Event”](#)
- [“Deleting an Event” on page 137](#)
- [“Deleting All Events in a Topic” on page 137](#)

For information on editing event properties, see [“Editing, Adding, and Deleting Event Properties” on page 138](#).

Sending a Test Event

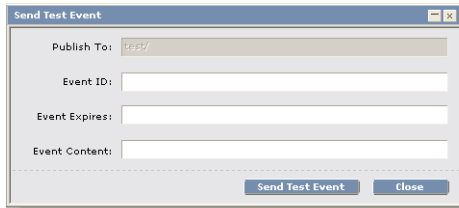
You can publish a test event to a topic to check a connection.

To send a test event to a topic,

1. Access the KnowNow System Administration console.
2. Select the Topics tab.
3. In the top panel, navigate to the desired topic.
4. In the bottom panel, select the Events tab.

5. From the **Events Menu**, choose **Send Test Event**. The Send Test Event dialog box pops up.

Figure 5-11. Sending a test event.



6. Fill in the fields as desired.
 - The **Publish To** field is already filled in with the topic you selected.
 - In the **Event ID** field, enter a unique ID for this event.
 - In the **Event Expires** field, enter an expiration time for this event. For information on event expiration times, see the description of **kn_expires** in the *KnowNow LiveServer Developer's Guide*.
 - In the **Event Content** field, enter the content you wish the event to hold.
7. Choose **Send Test Event** to send the event to the selected topic. When you choose **Send Test Event**, the new event appears in the event table, along with any other events that might exist for this topic.
8. To send another test event, repeat [step 6](#) and [step 7](#).
9. When you are done sending test events, choose **Close** to close the dialog box.
10. You can view the data for the test event just as you might for any other by clicking on the event, as described under [“Viewing Events and Event Properties”](#) on [page 133](#).

Deleting an Event

You can delete an event from a topic using the KnowNow System Administration console or using a browser.



Caution: Once you delete an event, it is gone forever. You cannot retrieve it; you can only recreate it by posting a new event to the given topic.

To delete an event from a topic using the KnowNow System Administration console,

1. Access the KnowNow System Administration console.
2. Select the Topics tab.
3. In the top panel, navigate to the desired topic.
4. In the bottom panel, select the Events tab.
5. Select the desired event on the left in the Events tab.
6. From the **Event Menu**, choose **Delete Selected Event**. The event is deleted.

You can also delete an event using an HTTP command sent from a Web browser.

To delete an event using a Web browser,

1. Publish a new event with the same `kn_id` as the event you wish to delete, an empty `kn_payload`, and `kn_expires` of "+0" as shown in the following example.

```
// delete event with kn_id "1234" from topic "/x/y/z"  
kn.publish( "x/y/z", { kn_id: "1234", kn_expires: "+0", kn_payload: "" } );
```

Deleting All Events in a Topic

In addition to deleting individual events, a topic can be cleared of all events with one command. Clearing events is useful when a topic becomes overcrowded with events, or during testing when you need to start with an empty topic.



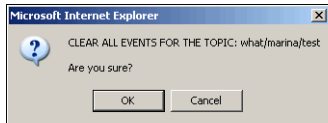
Caution: Once you delete all events in a topic, those events are gone and cannot be recovered; you can only recreate them by posting new events to the given topic.

To clear all events from a topic,

1. Access the KnowNow System Administration console.

2. Select the Topics tab.
3. In the top panel, navigate to the desired topic.
4. In the bottom panel, select the Events tab.
5. From the **Events Menu**, choose **Delete All Events**. A confirmation window appears.

Figure 5-12. Clear events confirmation prompt.



6. If you wish to proceed, choose **OK** to complete the action. The topic will then be emptied of events. To cancel the action, choose **Cancel** in the confirmation dialog box.

Editing, Adding, and Deleting Event Properties

In addition to working with events, you can also work with the individual properties of each event. The actions you can perform on event properties using the KnowNow System Administration console are described under the following headings:

- [“Editing and Deleting Event Properties”](#)
- [“Adding Event Properties” on page 139](#)

Editing and Deleting Event Properties

Working with event properties is similar to working with topic properties. For more on topic properties, see [“Editing a Topic Property” on page 124](#).

To edit or delete an event property,

1. Access the KnowNow System Administration console.
2. Select the Topics tab.
3. In the top panel, navigate to the desired topic.
4. In the bottom panel, select the Events tab.
5. In the lower left pane, select an event whose properties you wish to edit or delete. The properties associated with that event appear in the right pane.

6. In the lower right pane, select the event property you wish to edit or delete.
7. Choose **Property Editor**. The Property Editor, which is introduced under “[The Property Editor](#)” on page 121, slides open, displaying the name and value of the selected event property.
8. If you are editing this event property, modify the name and value of the event property in the proper fields. If you are deleting the event property, skip to [step 9](#).
 - a. In the **Property Name** field, enter a new name for the property.
 - b. In the **Property Value** field, enter a valid new value for the property.
 - c. Choose **Save Property** to save your changes. The event properties table is updated with the new value.
9. If you are instead deleting the property, choose **Delete Property** to delete this property.

Adding Event Properties

You can use the KnowNow System Administration console to add a property to an event. You can also add properties using the kn_ headers as described in the *KnowNow LiveServer Developer's Guide*.

To add a new event property,

1. Access the KnowNow System Administration console.
2. Select the Topics tab.
3. In the top panel, navigate to the desired topic.
4. In the bottom panel, select the Events tab.
5. Select an event in the lower left pane.
6. In the lower right pane, choose **Event Menu > Add Property to Selected Event**. The Property Editor (described under “[The Property Editor](#)” on page 121) slides open.
7. Enter a name and value for the property you wish to add.
8. Choose **Save Property** to save your changes. The property table is updated with the new value.

Managing Routes and Route Properties

Routes are the paths between topics along which events flow. Each topic can have several routes over which events flow from that topic to different locations, including KnowNow-powered applications, Connectors, and other topics. Routes also have their own properties.

Using the Routes tab, you can view routes, view and edit a route's properties, add new properties, add new routes, and delete routes. The instructions for performing these actions are provided under the following headings:

- ["Working with Routes"](#)
- ["Working with Route Properties" on page 144](#)

Working with Routes

Working at the route level, you can view routes, create routes, and clear routes as described under the following headings:

- ["Viewing a Topic's Routes"](#)
- ["Creating a New Route for a Topic" on page 142](#)
- ["Deleting All Routes from a Topic" on page 143](#)
- ["Deleting a Single Route from a Topic" on page 144](#)

When viewing routes for a topic or subtopic, you can also view the route's properties and can access to the Property Editor to edit those routes and their properties as described under ["Working with Route Properties" on page 144](#).

Viewing a Topic's Routes

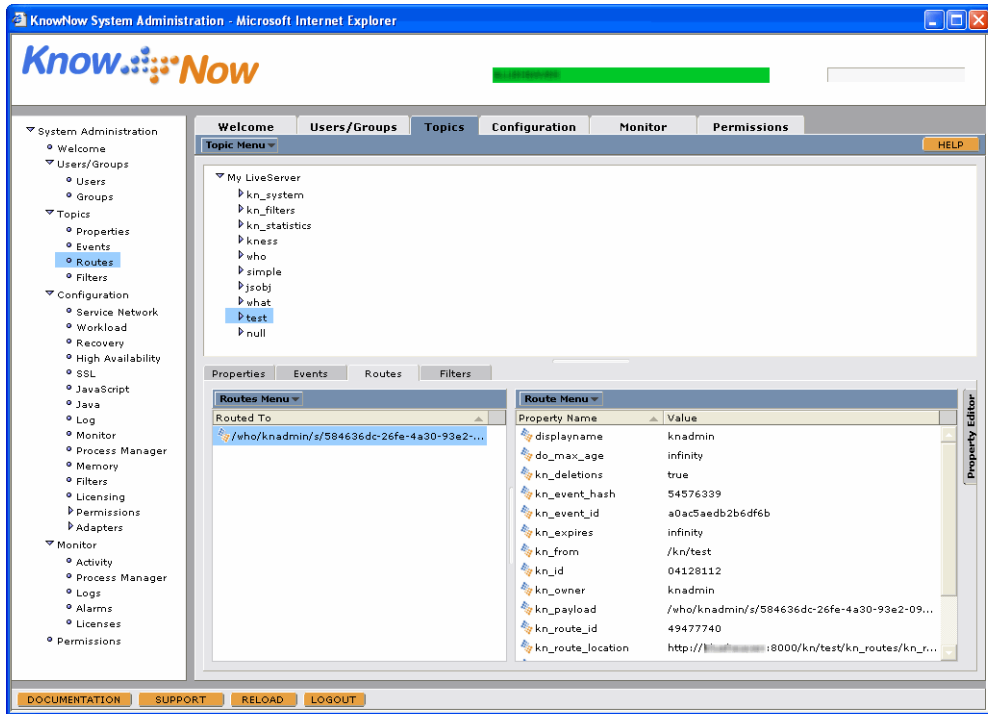
The Routes tab shows all routes for a topic, and also displays the properties for each route.

To view the routes for a topic,

1. Access the KnowNow System Administration console.
2. Select the Topics tab.
3. In the top panel, navigate to the desired topic.

- In the bottom panel, select the Routes tab. Any routes associated with that topic are displayed. Using this information, you can view and work with routes and route properties.

Figure 5-13. Viewing a topic's routes



- The lower left pane presents a list of the topic's existing routes under a column labeled **Routed To**. Events that are sent into the selected topic are routed to each of the routes in this list.
- Above the **Routed To** list is a **Routes Menu** button. This gives access to two commands for managing routes: **Create Route** and **Delete All Routes**. The use of these two commands is described under "Creating a New Route for a Topic" on page 142 and "Deleting All Routes from a Topic" on page 143, respectively.
- In the lower right pane is a list of properties for any given route that is selected on the left. This list provides the property name and value for each property for that route. For more information, see "Working with Route Properties" on page 144.

- Above the list of route properties is a **Route Menu** button, which gives access to a menu of several commands for managing individual routes, including **Delete Selected Route**. **Delete Selected Route** is discussed under [“Deleting a Single Route from a Topic” on page 144](#). The remaining commands from this menu are described under [“Working with Route Properties” on page 144](#).

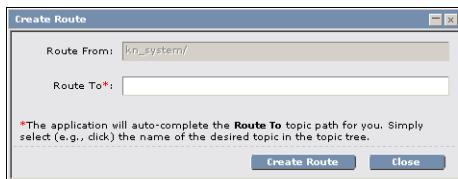
Creating a New Route for a Topic

While viewing the list of routes for a topic, you can add a new route for that topic. By definition, a route goes from one place to another, so you will need to know where you want your new route to start and where you want it to go to. Once the new route is established, events entering the source topic will be sent to the target topic.

To create a new route,

1. Access the KnowNow System Administration console.
2. Select the Topics tab.
3. In the top panel, navigate to the desired topic.
4. In the bottom panel, select the Routes tab.
5. Select a topic, then select the Routes tab. The selected topic becomes the source of your new route. Events sent into this topic will be sent to the topic you select in [step 7](#).
6. Choose **Routes Menu > Create Route**. The **Create Route** dialog box appears, with the selected topic in the **Routed From** field as the source of the route.

Figure 5-14. Create Route dialog box,



7. To fill in the **Routed To** field, select the endpoint for your route by simply navigating to and selecting a topic. Events from the topic you selected in [step 3](#) will be routed to this topic.
8. Once you have selected the target topic, choose **Create Route** to create the route. The route list table is updated in real time with the new route information. The

Create Route dialog box stays open in case you want to create another route. To cancel the creation of a route, choose the **Close** button in the **Create Route** dialog box instead of choosing **Create Route**.

9. To create another route, repeat [step 7](#).
10. When you are done creating routes to the selected topic, choose **Close**. The **Create Route** dialog box closes.

Deleting All Routes from a Topic

You can delete all routes from a topic with a single command, as described here, or you can delete one route at a time, as described under [“Deleting a Single Route from a Topic”](#) on page 144.

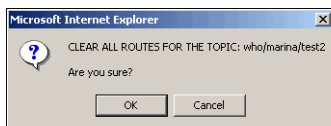


Caution: Once you delete all routes, you cannot undo this action. You will need to re-create the routes if you want them established again.

To delete all routes from a topic,

1. Access the KnowNow System Administration console.
2. Select the Topics tab.
3. In the top panel, navigate to the desired topic.
4. In the bottom panel, select the Routes tab.
5. Choose **Routes Menu > Delete All Routes**. A dialog box prompts you to confirm that you want to delete all routes for that topic.

Figure 5-15. Clear all routes confirmation window.



6. Choose **OK** in the confirmation dialog box to complete the action, or choose **Cancel** to stop the action and leave the routes as they are.

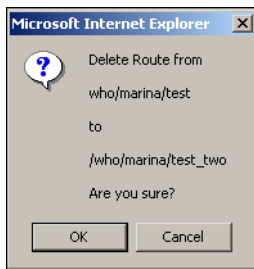
Deleting a Single Route from a Topic

You can delete a single route from a topic, as described here, or you can delete all routes from a topic with one command, as described under [“Deleting All Routes from a Topic”](#) on page 143.

To delete one route from a topic,

1. Access the KnowNow System Administration console.
2. Select the Topics tab.
3. In the top panel, navigate to the desired topic.
4. In the bottom panel, select the Routes tab.
5. Select a route in the route list in the lower left pane.
6. In the lower right pane, choose **Route Menu > Delete Selected Route**. A dialog box prompts you for confirmation.

Figure 5-16. Delete route confirmation window.



7. Choose **OK** in the confirmation dialog box to complete the action, or choose **Cancel** to cancel the deletion of that route.
8. Repeat [step 3](#) through [step 7](#) for each route you wish to delete. If you wish to delete all routes for a topic, you can do it faster by following the directions under [“Deleting All Routes from a Topic”](#) on page 143.

Working with Route Properties

In addition to working with routes, you can also work with an individual route’s properties as described under the following headings:

- [“Viewing Route Properties”](#)
- [“Editing, Adding, or Deleting a Route Property”](#) on page 145

Viewing Route Properties

While viewing the routes for a topic (as shown in [Figure 5-13 on page 141](#)), you can also view the properties for a single route.

To view the properties of a single route,

1. Access the KnowNow System Administration console.
2. Select the Topics tab.
3. In the top panel, navigate to the desired topic.
4. In the bottom panel, select the Routes tab.
5. In the lower right pane is a list of properties for any given route that is selected on the left. This list provides the property name and value for each property for that route. You can sort the list by clicking on the column headings.
 - To edit, add, or delete a property, see [“Editing, Adding, or Deleting a Route Property.”](#)
6. Above the list of route properties is a **Route Menu** button, which gives access to a menu of several commands. These commands are
 - **Add Property to Route**, which is described under [“Editing, Adding, or Deleting a Route Property.”](#)
 - **Attach Route Filter**, which is described under [“Attaching a Filter to a Route” on page 146.](#)
 - **Detach Route Filter**, which is described under [“Detaching a Filter from a Route” on page 147.](#)
 - **Delete Selected Route**, which is discussed under [“Deleting a Single Route from a Topic” on page 144.](#)

Editing, Adding, or Deleting a Route Property

You can change the values of each route property. You can also add a new route property or delete a route property. These actions can also be performed using `kn_` headers as described in the *KnowNow LiveServer Developer’s Guide*.

To edit, add, or delete a route property,

1. Access the KnowNow System Administration console.
2. Select the Topics tab.

3. In the top panel, navigate to the desired topic.
4. In the bottom panel, select the Routes tab.
5. In the lower left pane, select a route. The properties for that route are displayed in the lower right pane.
6. Choose **Property Editor** next to the route property. The Property Editor (described under [“The Property Editor” on page 121](#)) slides open with the current route property name and value in it. (If you are adding a route property, you can also choose **Route Menu > Add Property to Route**.)
7. To change or add a property, edit or enter the name and value of the property as needed. Choose **Save Property** to save your changes or to add a new property. If you don’t wish to save your changes, simply click on **Property Editor** again to close the editor. If you wish to delete the property instead, skip to [step 8](#).
8. To delete the property, choose **Delete Property**. The property is immediately deleted.

Attaching a Filter to a Route

You can attach one or more filter modules to any given route. This section tells how to attach a single filter module to a route using the KnowNow System Administration console. You can also use headers to attach multiple filter modules (and assign parameters for those filter modules) as described in the *KnowNow LiveServer Developer’s Guide*.

Filter properties can be set individually for each instance of a filter, or they can be set globally for a filter so that they apply to all topics and routes that they are attached to. For more information, see [“Enabling Global Filter Properties” on page 132](#).

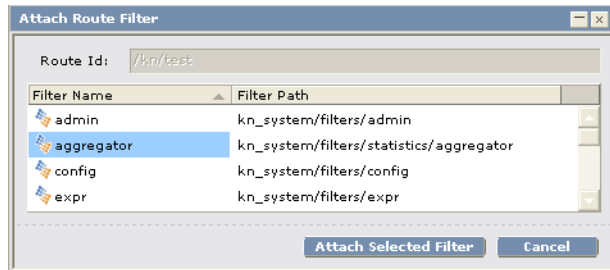
In addition to creating and using your own custom filter modules, you can also use filter modules that ship with LiveServer. These filter modules are discussed in more detail in the *KnowNow LiveServer Developer’s Guide*.

To attach a filter to a route,

1. Access the KnowNow System Administration console.
2. Select the Topics tab.
3. In the top panel, navigate to the desired topic.
4. In the bottom panel, select the Routes tab.
5. In the lower left pane, select a route.

- In the lower right pane, choose **Route Menu > Attach Route Filter**. A dialog box for attaching a filter to that route appears. In this dialog box, a list of filters appears. Not all the filters in this list are available for use; some are internal LiveServer filters. The LiveServer filters that you can use are documented in the *KnowNow LiveServer Developer's Guide*. You can also use any custom filters that have been uploaded using the procedure described under “[Loading Custom Filter Modules](#)” on page 95.

Figure 5-17. Attaching a filter to a route.



- Select a filter in the list and choose **Attach Selected Filter**. The selected filter is attached to the route. If the route is not found, you may get an Error 404 - Route Not Found message. For more information on applying filters, see “[Topics and Filter Modules](#)” on page 128.

Detaching a Filter from a Route

You can detach a filter module from a route. When you do so, you are not deleting that filter from the system, but are instead just detaching it from the specific route. Other routes that have the same filter attached will still have that filter attached, and you can also attach that filter to other topics and routes. (You can also edit a route or subscription's properties to detach filters.) The filter is detached and the filter properties table is updated in real time.



Caution: Detaching a filter can have unexpected results, and you will **not** be given a confirmation prompt for this action. Be sure you know what you are doing and what the effects will be before detaching a filter.

To detach a filter from a route,

- Access the KnowNow System Administration console.
- Select the Topics tab.

3. In the top panel, navigate to the desired topic.
4. In the bottom panel, select the Routes tab.
5. In the lower left pane, select a route.
6. In the lower right pane, choose **Route Menu > Detach Route Filter**. The filter is detached.

Chapter 6 Selecting a Permissions System

This chapter describes the LiveServer's permissions options and how to select a permissions module. Includes information on configuring the Flat File and 2.0 Compatible permissions modules. The LDAP and LDAP Test permissions modules are covered in [Chapter 7 , "Implementing Security with LDAP"](#)

The topics covered in this chapter are:

- "Selecting a Permissions Module" on page 150
- "Configuring Flat Files Permissions" on page 153

Selecting a Permissions Module

LiveServer offers a number of choices for configuring permissions, which involves authenticating and authorizing users and groups. Authorization modules verify access rights, check to see if a particular set of credentials can access a particular URL on LiveServer, and so on. When you make and configure a permissions selection on the **Configuration > Permissions** panel, you are selecting and configuring an authorization module.

The selected authorization module also controls authentication. Authentication is configured by parameters you set when configuring your chosen authorization module. Authentication modules are called to verify a user, usually based on a user ID and password. LiveServer does not call the authentication module directly; it is up to the authorization layer to call it using the API call `knAuthenticateUser`.

Some security files are associated with many of the permissions options you choose. These files are described in [Chapter 8](#), “[Managing Permissions](#).”

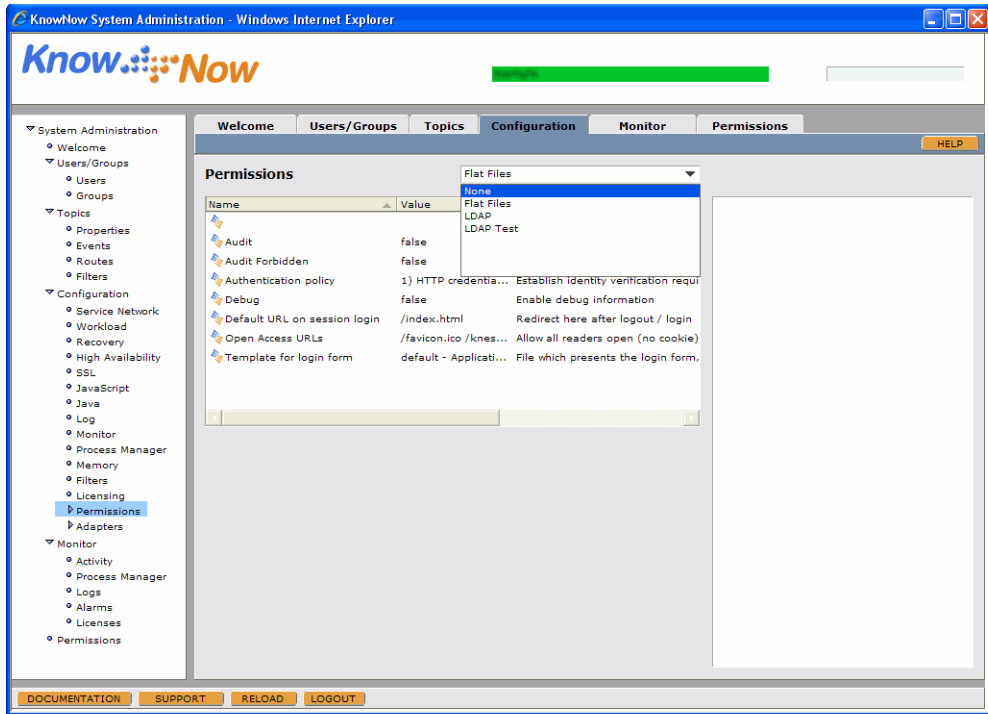
The first step in configuring permissions is to select a permissions module. After making that choice, you can configure the selected module. If you wish to use the Flat Files permission system, you can skip this procedure and move ahead to configuring this system as described under “[Configuring Flat Files Permissions](#)” on page 153.

To select a permissions module,

1. Access the KnowNow System Administration console.
2. Under **Configuration**, select **Permissions**. The Permissions configuration panel is displayed.
3. From the drop-down list, select one of the permissions modules. When you do so, the parameters for configuring the selected permissions module appear.
 - **None** specifies that no permissions module is to be used.
 - **Flat Files** (the default selection) specifies using a flat-file-based system.
 - **LDAP** specifies using the LiveServer’s LDAP permissions module.
 - **LDAP Test** specifies using a version of the LDAP module that you can use for testing to see if you can talk with your LDAP back end without having to exit the system.

With both the Flat Files module and the LDAP modules, you can add and drop users from your database without having to restart LiveServer.

Figure 6-1. Choosing an authorization module.



4. Configure the selected permissions module.

- For Flat Files, see [“Configuring Flat Files Permissions”](#) on page 153 and [“Flat Files Parameters”](#) on page 154
- For LDAP, see [Chapter 7, “Implementing Security with LDAP.”](#) That chapter provides
 - vital information on integrating your LiveServer with your LDAP environment ([“Integrating ESS With Your LDAP Environment”](#) on page 161)
 - specifics on selecting and configuring the KnowNow LDAP modules ([“Selecting and Configuring an LDAP Module”](#) on page 223)
 - a description of the KnowNow System Administration console’s parameters for LDAP ([“LDAP and LDAP Test Parameters”](#) on page 225)
 - additional LDAP configuration information ([“Advanced LDAP Configuration”](#) on page 229)

5. After configuring your selected permissions module, if you are done making changes, restart LiveServer.

Configuring Flat Files Permissions

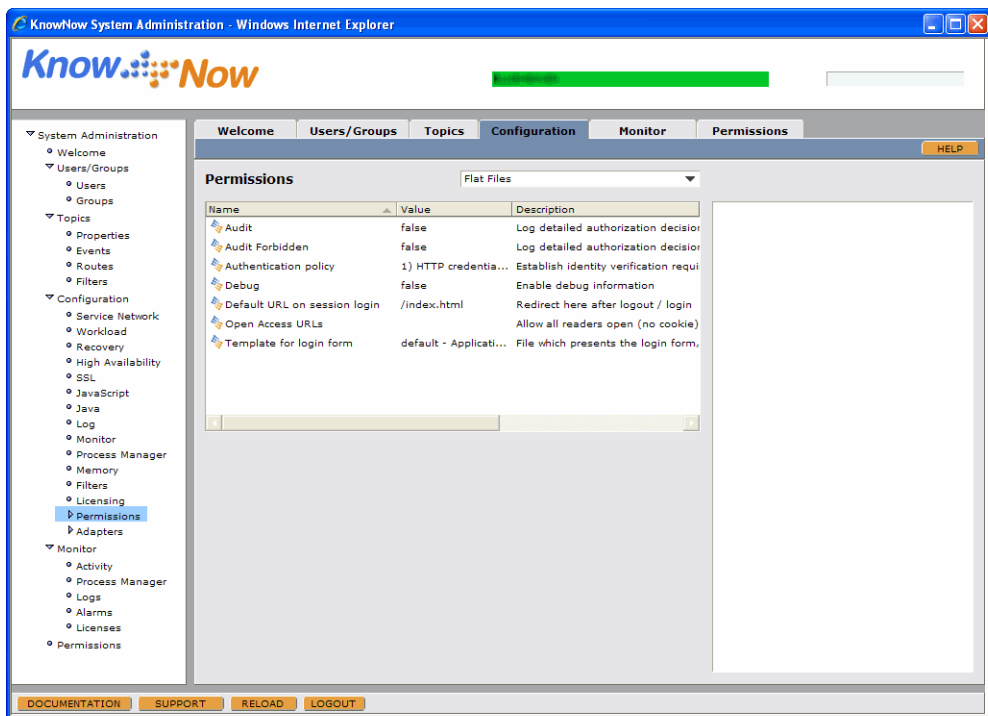
The Flat Files permissions system is the default security module. The Flat Files module offers a number of capabilities, including the ability to create and use permissions patterns as described under [Chapter 8 , “Managing Permissions.”](#)

You can instead use LDAP; if you wish to do so, select one of the LDAP options as described under [“Selecting a Permissions Module”](#) on page 150, then refer to [Chapter 7 , “Implementing Security with LDAP”](#)

To configure Flat Files permissions,

1. Under **Configuration**, select **Permissions**. The Permissions configuration panel is displayed.
2. If you have changed your permission module to something else, and want to revert back to Flat Files, select **Flat Files**.

Figure 6-2. The Flat Files module parameters.



3. Configure the Flat Files parameters, which are described under [“Flat Files Parameters”](#) on page 154.

4. If desired, configure permissions patterns as described under [Chapter 8 , "Managing Permissions."](#)
5. If desired, configure parameters for internal users as described under ["Configuring Internal Users" on page 257.](#)
6. When you are done making changes, restart LiveServer.

Flat Files Parameters

This permissions module has the following parameters to configure.

Audit

When set to **true**, detailed authorization decisions are entered as log entries in the LiveServer's log file, *server.log*. The default value is **false**.

Audit Forbidden

When set to **true**, authorization decisions that result in a "forbidden" response are entered as log entries in the LiveServer's log file, *server.log*. The default value is **false**.

Authentication policy

This field provides a number of different authentication policies as described below. For a summary of this information, see [Table 6-1 on page 156.](#)

1. When the **HTTP credentials required** policy is selected, every connection requires a login with password. No anonymous logins are allowed. If someone tries to log in without presenting the proper credentials, they will be denied access.

This policy relies on HTTP basic authentication. This means that the browser, not any specific application (such as LiveServer), will prompt for the user name and password.

2. When the **Credentials required - no anonymous** policy is selected, browser users are presented with an application-specific login form (by default, LiveServer presents this form). This form can be customized as desired; see ["Template for login form" on page 158.](#) No anonymous logins are allowed. If someone tries to log in without presenting the proper credentials, they will be denied access.

Non-browser users (for example, LiveServer Connectors) are required to provide HTTP basic authentication headers.

This policy is useful in a number of situations. For example, in this situation: You are using **HTTP credentials required**, you change a user's password, and that user is logged into a complicated LiveBrowser application, then that user would see a number of login prompts. To simplify the login process in such a situation, use this policy (**Credentials required - no anonymous**).

This is the default value for the **Authentication Policy** parameter. See also the **Default URL on session login** parameter.

3. The **Credentials, anonymous on demand** policy is very similar to the **Credentials required - no anonymous** policy, except that it also allows anonymous logins. When attempting access using a browser, users are presented with an application-specific login form that prompts them to log in (by default, LiveServer presents this form). However, users can elect to remain anonymous when they log in.

The login form that is presented must be coded with conditional logic to provide the anonymous login option; that conditional logic is provided in the default (sample) login.adp form. This form can be customized as desired; see ["Template for login form" on page 158](#). For additional information, see also the **Default URL on session login** parameter.

Note that non-browser users are still required to provide HTTP basic authentication credentials in this case, and are *not* allowed anonymous access.

4. When you select the **Open Access** policy, anonymous logins are allowed to unrestricted areas of LiveServer. Any HTTP client can access LiveServer without presenting credentials. Therefore, normally no login form is presented to users. However, when a user attempts to access a restricted part of the LiveServer's URL (and thus, topic) space, LiveServer responds with an HTTP "401 Unauthorized" response, which causes the browser to prompt for a login (as with the **HTTP credentials required** option). No application-specific login form is provided with this option.



Caution: This policy is *not* recommended for ESS.

5. The **Anonymous, credentials on demand** policy is similar to the **Open access** policy. When users access LiveServer, they gain access immediately and are automatically anonymous. the difference between this policy and policy 4 is that when this policy is selected, LiveServer retains the information about which

anonymous user a person is across browser sessions, and users can also log in as a different user by accessing the login screen

Table 6-1. Authentication policies summary table.

Policy	Good Choice for Clusters and Cross-Domain Deployment?	Anonymous Logins Allowed?	Experience of browser-based users	Experience of non-browser-based client applications to LiveServer
1 - HTTP credentials required	Can be used, but other policies recommended	No	All user authentication is performed using basic authentication. Browser-based users are presented with the browser's login dialog box requesting the user's ID and password.	KnowNow Alerts Desktop (and other applications that use KnowNow Connectors) must provide credentials using basic authentication.
2 - Credentials required - no anonymous	Yes	No	Browser-based users are presented with an HTML form where the user can input credentials.	KnowNow Alerts Desktop (and other applications that use the KnowNow Connectors) must provide credentials using basic authentication.
3 - Credentials, anonymous on demand	Yes	Yes	An HTML form is presented to users so they can specify credentials. This form includes a checkbox for anonymous login (logging in without credentials).	KnowNow Alerts Desktop (and other applications that use the KnowNow Connectors) must provide credentials using basic authentication.
4 - Open Access	Can be used, but other policies recommended	Yes	Users can access any unrestricted LiveServer area without having to log in. However, if users try to access a restricted area, they are presented with a login form and must provide valid credentials.	All applications are allowed to connect without having to present credentials. Credentials must be presented to access restricted areas.

Table 6-1. Authentication policies summary table. (continued)

Policy	Good Choice for Clusters and Cross-Domain Deployment?	Anonymous Logins Allowed?	Experience of browser-based users	Experience of non-browser-based client applications to LiveServer
5 - Anonymous, credentials on demand	Yes	Yes	The same as policy 4, except that certain settings a particular user might have set are retained across sessions. Also, users can access the login screen to log in as a different user.	The same as policy 4.

Debug

Turns on debugging statements in the log. Enable this parameter only when asked to do so by KnowNow Support. The default value is **false**.

Default URL on session login

If you're using certain options in the **Authentication policy** parameter, and a user logs out, then logs in again, the browser is directed to the URL specified in this parameter. This URL is also used in any other case in which a user logs in without having previously presented LiveServer with a non-login URL. (For example, when a user goes directly to `http://example.com/login/login.html` and logs in.) The default value is `/index.html`.

Open Access URLs

When any URLs are set in this parameter, and when policy 2, 3, or 5 is in effect, then all readers are allowed open access to those URLs (no cookies are required) unless that access is forbidden by an access pattern. The default value is for this parameter to have no URLs specified. For information on access patterns, see [Chapter 8, "Managing Permissions."](#)

The URLs here must use the syntax known as a "glob pattern." That is, it looks like file-name patterns in command-line systems, *not* like the pattern language KnowNow uses in the Permissions tab. For example, to specify "everything that starts with `/kness/channels`"; use `"/kness/channels/*"`, *not* `"/kness/channels/!"`

To separate values, use spaces.

Template for login form

When selecting certain **Authentication policy** options, users are prompted to log in using a custom login form. LiveServer ships with a default login form; you can customize that form or supply your own custom login form. If you choose to do so, use this parameter to specify the location of that custom form. The default location and file name is either specified by a custom application or is **`${homedir}/runtime/pages/login/login.adp`**.

Chapter 7 Implementing Security with LDAP

LiveServer is shipped with a lightweight directory access protocol (LDAP) module for managing users and groups and to mediate authentication with production-level security inside corporations and across firewalls. LDAP is useful when you wish to have a single repository for users and groups in your company.

This chapter presents information on integrating ESS with LDAP using several common scenarios. Although the necessary steps are provided, we assume that you are already familiar with LDAP. For an excellent work on LDAP, see *Understanding and Deploying LDAP Directory Services*, Second Edition, by Timothy Howes, Mark G. Smith, and Gordon S. Good.

LDAP deployment and configuration information is described under the following headings:

- [“LiveServer LDAP Overview” on page 160](#)
- [“Integrating ESS With Your LDAP Environment” on page 161](#)
- [“Deploying ESS in a Dual LDAP Environment: AD With OpenLDAP” on page 170](#)
- [“Deploying ESS in a Dual LDAP Environment: AD With ADAM” on page 179](#)
- [“Deploying ESS in a Dual OpenLDAP Environment” on page 188](#)
- [“Deploying ESS in a Standalone OpenLDAP Environment” on page 197](#)
- [“Deploying ESS in a Standalone AD Environment” on page 205](#)
- [“Reducing the LSadmin User’s Privileges” on page 215](#)
- [“Selecting and Configuring an LDAP Module” on page 223](#)
- [“Advanced LDAP Configuration” on page 229](#)

LiveServer LDAP Overview

The LiveServer's LDAP implementation is primarily for discovering security information and for storing application-specific security information. LiveServer does not use LDAP to store configuration information or to store actual application data.

When LiveServer authenticates a user using LDAP, it tries to connect to that LDAP server as that user using the password that user provided. For more on managing users and groups with the KnowNow System Administration console, see [Chapter 9, "Managing Users and Groups."](#)

If you prefer using your existing LDAP infrastructure, you can configure LiveServer to use that existing LDAP infrastructure

By implementing an LDAP solution, you can have the same password for LiveServer as you do for other systems in your company's infrastructure (if you are already using LDAP for your password management).

If during system operation the connection to the directory server is lost, the LDAP module tries to reconnect to the directory server. During this disconnected period, the module will return the Allow or Deny values according to the cache. There is a user-configurable timeout period after which, if no reconnection to the directory server has been made, the module returns all "denies" until reconnect and cache refresh.

For further information, see the following sections:

- For LiveServer/LDAP integration information, see ["Integrating ESS With Your LDAP Environment" on page 161.](#)
- For specifics on selecting and configuring the KnowNow LDAP module, see ["Selecting and Configuring an LDAP Module" on page 223.](#) The KnowNow System Administration console's parameters for LDAP are described under ["LDAP and LDAP Test Parameters" on page 225.](#)
- For additional KnowNow System Administration console configuration information, see ["Advanced LDAP Configuration" on page 229.](#)

Integrating ESS With Your LDAP Environment

Integrating ESS with an LDAP environment is a complex, multi-task process. An overview of the possible tasks is provided in this section.

In most cases, you will be deploying LiveServer/ESS and LDAP on separate servers.

Note that LiveServer implements its own authorization scheme; it does not use the access control scheme provided in your LDAP server. However, it must be able to access this authorization data as an LDAP user (actually, two different LDAP users), so you may need to set up access control instructions (ACIs) in your LDAP server to allow it access to the appropriate information.



Caution: If you are going to create clusters and use LDAP, create clusters **before** integrating LDAP.

Integration is described under the following headings:

- [“Prerequisites for LDAP Installation”](#)
- [“Background Information on LDAP Integration Tasks” on page 164](#)
- [“Supporting Files” on page 169](#)

For assistance and the latest integration information, please contact KnowNow Support.

Prerequisites for LDAP Installation

You will need the following tools and configurations before you attempt to integrate LiveServer with LDAP:

1. **Required:** A directory-specific tool for managing LDAP users and groups. In AD and ADAM, this is ADSI Edit.
2. **Required** if the directory service requires it (such as AD and ADAM): An LDAP browser that knows how to log in via SSPI. AD and ADAM use ldp.exe.
3. **Required:** A system prepared to run Perl scripts as described under [“Perl Environment Requirements” on page 162](#).
4. Ideally, you will also need distinguished names (DNs) for the following users:
 - An LDAP user with the appropriate credentials for extending your LDAP schema.

- An LDAP user with the appropriate credentials for reading and writing to selected areas of your directory. This user may be used in production, but depending on your work flow, you can stop using this user after the integration is complete (and start using normal LDAP users instead).
- An LDAP user with only read capabilities for selected areas of your directory.

In the AD and ADAM integration examples, a single user (a single DN) is created to perform the functions of the first two users mentioned above.

5. It is also recommended that you configure LiveServer for cross-domain in order to propagate your well-known name when loading the LiveServer objects. For more information, see [“Cross-Domain Configuration Example” on page 88](#).

Perl Environment Requirements

The LDAP libraries needed by KnowNow’s Perl script are in the */LiveServer_root/bin/* directory. To prepare to run the Perl script, you’ll need to establish a command-line environment that has the following settings.

- On Windows, add the */LiveServer_root/bin/* directory to your PATH environment variable; on Linux or UNIX, add it to both your PATH and your LD_LIBRARY_PATH environment variables **before** any system library directories.
- The LiveServer Perl installation must be the first Perl installation in the PATH environment variable. The KnowNow Perl binary is located in *../lib/perl/knperl/bin/* in the LiveServer installation directory.
- The LiveServer */bin/* directory (*LiveServer_root/bin/*) must be placed right after the setting of the location of the Perl binary.
- On Linux and UNIX, you also need to set your PERLLIB environment variable. See the file */LiveServer_root/lib/perl/config-perl.txt* for the settings.

Setting Up the Perl Environment

Before you add users and groups, set up the Perl environment variables on LiveServer as follows. The paths are given based on the assumption that LiveServer is installed in the default location; if it is not, replace C:\Program Files\KnowNow\Server\ or /opt/kn-server/ with the correct string.



Warning: If you choose to copy and paste commands from this documentation, first, copy and paste *only from the help files*, not from the PDF. Second, be careful: when copying from HTML, certain editors may embed trailing blanks, carriage returns, or line breaks, rendering the command ineffective. In addition, apostrophes may be incorrectly transcribed as beginning single quotes.

On Windows,

1. Set the environment variable **perlroot** to

```
C:\Program Files\KnowNow\Server\lib\perl
```

2. Add the following to the PATH variable:

```
C:\Program Files\KnowNow\Server\lib\perl\knperl\bin;
C:\Program Files\KnowNow\Server\bin;
C:\Program Files\KnowNow\Server\lib\perl\knperl\lib\CORE;
```

3. To test, run the following command from a command prompt. If the Perl environment variables are correct, you will see OK on the screen.

```
perl -MKnowNow::Utilities -e "{ print 'OK'; }"
```

On Linux,

1. Enter the following commands:

```
export perlroot=/opt/knserver/lib/perl
export knroot=/opt/knserver
export SCHEMADIR=/opt/knserver/lib/ldap/schema
export
PERLLIB="${perlroot}/knperl/lib/site_perl/5.8.3/KNOWNOW${perlroot}/knperl/lib/5.8.3/i686-linux-thread-multi:${perlroot}/knperl/lib/5.8.3:${perlroot}/knperl/lib/site_perl/5.8.3/i686-linux-thread-multi:${perlroot}/knperl/lib/site_perl/5.8.3:${perlroot}/knperl/lib/site_perl"
export PATH="/opt/knserver/lib/perl/knperl/bin:/opt/knserver/bin:${PERLLIB}:${PATH}"
export LD_LIBRARY_PATH=$PATH
```

2. To test, run the following command. If the Perl environment variables are correct, you will see OK on the screen.

```
perl -MKnowNow::Utilities -e '{ print "OK\n"; }'
```

Background Information on LDAP Integration Tasks

After ensuring that your system meets the prerequisites described under [“Prerequisites for LDAP Installation” on page 161](#), you will be performing a set of tasks for integrating LiveServer/ESS with your LDAP environment. As needed, background information for these tasks is provided under the following headings in this section; the specific tasks for different environments are described under the specific integration scenarios presented later in this chapter.

- [“Create an ESS Container and ESS Admin User”](#)
- [“Load LiveServer Schema Extensions” on page 164](#)
- [“Create the LiveServer Users and Groups” on page 167](#)
- [“Load Permission Patterns” on page 168](#)
- [“Configure LiveServer for LDAP” on page 168](#)
- [“Reduce the LSadmin User’s Privileges \(In AD\)” on page 169](#)

Create an ESS Container and ESS Admin User

Using KnowNow’s tools, identify an ESS admin user with the proper credentials to modify the directory server’s schema. If deploying a LiveServer for the first time, you may need to create that user.

The user you identify or create is to extend the schema and query the LDAP server.



Note: You only need to create an ESS admin user with the power to modify the schema if you don’t already have access to such a user.

Load LiveServer Schema Extensions

When you implement LDAP for ESS in your enterprise, you must extend your LDAP schema. LDAP will need a definition of the objects that ESS will request. The procedure for doing this differs for each directory service vendor.

We assume you will be using the same directory information tree (DIT) for each LiveServer in your enterprise, in which case you would only perform this step once. Different directory servers use different mechanisms for extending their schemas. KnowNow provides two sets of directory extension scripts: one intended for directory servers which support definition of new schema objects in RFC 2252 format, and one intended for Microsoft AD and ADAM, which do not. (All LDAPv3 DSAs support *export* of schemas in RFC 2252).

The schema extensions will provide two new object classes, and eighteen new attribute types. These objects and attributes define the rules that govern event publications, topic routes, and topic subscriptions.

The two object classes are

- comKnowNowAccessPattern
- comKnowNowAuthorizationEntry

The attribute types are

- comKnowNowAccessPatternDN
- comKnowNowAllowChange
- comKnowNowAllowGet
- comKnowNowAllowPost
- comKnowNowAllowPublish
- comKnowNowAllowSubscribe
- comKnowNowAuthorizationDN
- comKnowNowDefault
- comKnowNowGroup
- comKnowNowOriginator
- comKnowNowOwner
- comKnowNowPeer
- comKnowNowPeerPort
- comKnowNowPeerUrl
- comKnowNowProtocol
- comKnowNowServerPort

- comKnowNowServerUrl
- comKnowNowUser

None of the schema or object files will import directly as LDIF files without some changes. To reflect this, the extension of these files is .ldff (LDIF fragment). The various installation scripts we provide can help turn these into legitimate .ldif files for your specific installation; in many cases, they also import it directly to your directory. For both AD and ADAM, you'll be using the same ldif extension file.

eDirectory Example: Perl-based Schema Load from RFC2252-compliant Files

For Novell eDirectory, you extend the schema by executing a Perl script which loads RFC2252-compliant schema definitions. The procedure presented here should work for any LDAP server that supports schema loading from RFC2252-compliant files, and that does so by allowing you to modify the returned value from a query for subschemaSubentry.

To load the schema,

1. Set your Perl environment variables as described under [“Perl Environment Requirements”](#) on page 162.
2. From a command shell, go to the KnowNow LDAP directory on your system. This directory is in the following locations by default (schemas are located in /schema/ below the /ldap/ directory):
 - **Windows:** C:\Program Files\KnowNow\Server\lib\ldap\
 - **Linux:** /opt/knserver/lib/ldap/
3. Execute the schema population script using this command:

```
$ perl ../../bin/loadschema \  
--ldap_url ldaps://yourserver \  
--bind_dn admin_login \  
--root_dn good_DN \  
--password password schema
```

For example,

```
$ perl ../../bin/loadschema \  
--ldap_url ldaps://edirectory \  
--bind_dn cn=Lsadmin,o=knownow \  
--root_dn o=knownow \  
--password mypassword schema
```

Notes:

- *Loadschema* takes a number of command line options; use the following command to get a list of those options:

```
perl ../../../../bin/loadschema --help
```

- The `root_dn` is the element that will be searched for the `subschemaSubentry` attribute; that value will then be the entry modified by each schema file.
- The schema is expressed in RFC2252 format; each attribute and class has its own `.ldiff` file in the `/LiveServer_root/lib/ldap/schema/` directory.

Create the LiveServer Users and Groups

You will need to add a set of default LiveServer users and groups. (For important information on these users and groups, see [“The LiveServer’s Default Users and Groups” on page 252](#).) These users and groups can be added manually by the AD administrator. Alternatively, if desired, they can be added using a Perl script provided by KnowNow.

The users are:

- **kndaemon**
- **kninternal**
- **knadmin**
- **~KnDummy**

The groups are:

- **KnowNow_Administrators**
- **KnowNow_Internal**

KnowNow_Internal will contain **kndaemon** and **kninternal**.

KnowNow_Administrators will contain **knadmin**. All LiveServer administrators should be members of the **KnowNow_Administrators** group.

Load Permission Patterns

In this step, you load the permission patterns and authorizations. You'll be running a Perl script.



Note: Permissions changes can take some time (from minutes to hours) to propagate and synchronize within Microsoft Active Directory and other LDAP systems. The amount of time required is not controlled by LiveServer. Please allow sufficient time for these changes to propagate.

Note that if you are integrating with a LiveServer cluster, you must provide the `--name` option; for example,

```
--name knserver.example.com
```

The name value should match the host name in the **Configuration > Service Network > Hostname** parameter in the KnowNow System Administration console (the name of your load balancer, typically).

Configure LiveServer for LDAP

Once you've extended the schema and added the objects, you are ready to choose the LDAP permission module. Here is an overview of the required steps; the sections on each specific installation provide more details.

To select and configure the LDAP module,

1. Access the **Configuration > Permissions** tab.
2. Select **LDAP**. Configure the parameters as described for each type of integration later in this chapter.
3. Restart LiveServer.

Once you restart your LiveServer, the new LDAP permissions will be enforced.

- You will now have to log in as a user that is in your LDAP database.
- If you cannot log into your LiveServer once the reboot has occurred, make sure you're logging in using the same attribute as the LiveServer is using for your user ID.
- If you cannot log in and need to get your LiveServer operational again, edit the file `/LiveServer_root/conf/knsettings.conf` to return to the flat file subsystem. Change the line that reads

```
set Permissions_Module {knperm}
```

to read

```
set Permissions_Module {knperm_files}
```

Then please call KnowNow Customer Support for assistance.

Reduce the LSadmin User's Privileges (In AD)

For the purposes of performing the LDAP integration tasks, the **LSadmin** user must have certain privileges. After the integration tasks have been completed, those privileges can be reduced if desired. However, those privileges must be reduced in specific ways so that the **LSadmin** user can continue to use LiveServer. When integrated with AD, LiveServer stores permission patterns in an AD container. This container is associated with the access control list (ACL) that defines the access privileges of AD users, groups, and roles. You can use Microsoft's ADSI Edit tool to edit the ACL to control operations toward AD containers/objects for AD users, groups, and other AD objects. The **LSadmin** user must have full control over the LiveServer container. Instructions are provided under "[Reducing the LSadmin User's Privileges](#)" on page 215.

Supporting Files

KnowNow provides a number of files to support LDAP integration; these files are located in the KnowNow LDAP directory, which in the default installation is in the following locations:

- **Windows:** C:\Program Files\KnowNow\Server\lib\ldap\
- **Linux:** /opt/knserver/lib/ldap/

Deploying ESS in a Dual LDAP Environment: AD With OpenLDAP

The tasks in this section deploy ESS in a dual LDAP environment that uses AD for authentication and OpenLDAP for authorization. **This deployment is recommended for ESS deployed on Linux servers.**

The tasks to perform for this environment are as follows. For background information on some of the tasks provided in this section, see [“Background Information on LDAP Integration Tasks”](#) on page 164.

1. [“Task 1: Install and Validate the OpenLDAP Server”](#)
2. [“Task 2: Create an ESS Container and ESS Admin User in OpenLDAP”](#) on page 171
3. [“Task 3: Load LiveServer Schema Extensions in OpenLDAP”](#) on page 172
4. [“Task 4: Load Permission Patterns in OpenLDAP”](#) on page 172
5. [“Task 5: Load the LiveServer Users and Groups in AD”](#) on page 173
6. [“Task 6: Configure the LiveServer’s LDAP Module”](#) on page 174
7. [“Task 7: Validate Your Configuration”](#) on page 177

Task 1: Install and Validate the OpenLDAP Server

Install the OpenLDAP server on the LiveServer machine (on Linux) according to the instructions for that product. Then validate the OpenLDAP installation by running the following commands on the same machine.

To validate the OpenLDAP installation,

1. Validate the LDAP installation and change the run levels using the following commands:

```
>chkconfig --list ldap  
>chkconfig --levels 235 ldap on
```

2. Start the OpenLDAP server by running the following command:

```
>service ldap start
```

3. Check the OpenLDAP server health by running the following command:

```
ldapsearch -x -b '' -s base '(objectclass=*)' namingContexts
```

4. In `/etc/openldap/`, edit `ldap.conf` to comment out the default settings.

5. In `/etc/openldap/`, edit `slapd.conf`, replacing the following with the declarations below:

```
suffix          "dc=kn-openldap,dc=com"
rootdn          "cn=admin,dc=kn-openldap,dc=com"
rootpw         password
```

6. Restart OpenLDAP.

```
# service ldap restart
```

Task 2: Create an ESS Container and ESS Admin User in OpenLDAP

In this task, you are adding ESS entries to the OpenLDAP directory. Use the `ldapadd` program to perform these steps. `ldapadd` expects input in LDIF form, so you will first create a file named `kn_init_objects.ldif`, run `ldapadd`, and change the password. The user created in this example will be able to administer the permission patterns in the authorizing database. The DN of this user, which in this example is **cn=LSadmin,ou=liveserver,dc=kn-openldap,dc=com**, is used in the **LiveServer Authoring DN** parameter.

To create an ESS container and admin user in OpenLDAP,

1. Use your favorite editor to create an LDIF file named `kn_init_objects.ldif` that contains the following:

```
dn: dc=kn-openldap,dc=com
objectclass: dcObject
objectclass: organization
o: KnowNow
dc: kn-openldap

dn: ou=liveserver,dc=kn-openldap,dc=com
objectclass: organizationalUnit
ou: liveserver

dn: cn=LSadmin,ou=liveserver,dc=kn-openldap,dc=com
objectclass: person
sn: LSadmin
cn: LSadmin
```

2. Run the following command to add the object and user. OpenLDAP prompts for the rootdn password you specified in `slapd.conf`.

```
>ldapadd -x -D "cn=admin,dc=kn-openldap,dc=com" -W -f kn_init_objects.ldif
```

3. Change the password:

```
>ldappasswd -x -v -S -W -D cn=admin,dc=kn-openldap,dc=com \  
cn=LSadmin,ou=liveserver,dc=kn-openldap,dc=com
```

Task 3: Load LiveServer Schema Extensions in OpenLDAP

LDAP will need a definition of the objects that ESS will request.

To load the LiveServer schema extensions,

1. Stop the OpenLDAP daemon.

```
>service ldap stop
```

2. Copy the file named *knpermission.schema* to the OpenLDAP schema directory:

```
>cp /opt/knserver/lib/ldap/schema/knpermission.schema /etc/openldap/schema/
```

3. Edit *slapd.conf* to extend the schema and to give the needed permissions as follows.

- Insert the following line for the schema extensions:

```
include /etc/openldap/schema/knpermission.schema
```

- Insert the following lines for permissions. The indentation for the second and third lines is important.

```
access to *  
  by users write  
  by * read
```

4. Restart the OpenLDAP daemon:

```
>service ldap restart
```

Task 4: Load Permission Patterns in OpenLDAP

Now run a Perl script to load the permission patterns and authorizations.

```
> cd /opt/knserver/lib/ldap  
  
perl ../../bin/knloadldif \  
--doit \  
--ldap_url ldap://OpenLDAP_server:389/ \  
--bind_dn cn=LSadmin,ou=liveserver,dc=kn-openldap,dc=com \  
--password password \  
--kn_container ou=liveserver,dc=kn-openldap,dc=com \  
--group_container "OU=LiveServer - knownow.example.com, ou=liveserver,dc=kn-  
openldap,dc=com" \  
--user_class user \  

```

```
--group_other "groupType: 2147483650" \  
--name ESS_server_name.example.com \  
data/knobjects_openldap.ldif
```



Warning: If you choose to copy and paste commands from this documentation, first, copy and paste *only from the help files*, not from the PDF. Second, be careful: when copying from HTML, certain editors may embed trailing blanks, carriage returns, or line breaks, rendering the command ineffective. In addition, apostrophes may be incorrectly transcribed as beginning single quotes.

Task 5: Load the LiveServer Users and Groups in AD

In order to function, ESS requires four specific LiveServer users belonging to two specific LiveServer groups (as discussed under [“Create the LiveServer Users and Groups” on page 167](#)).

In AD, create a container to hold those users and groups. This task can should be performed by the AD administrator; it can also be done by using the Perl script provided by KnowNow. The following instructions tell how to perform this task using the Perl script provided by KnowNow.

On Windows,

1. In AD, add a user whose DN will be used to bind to Active Directory to query the authentication database. In this example, we call this user **DSadmin**.
2. Set your Perl environment variables as described under [“Perl Environment Requirements” on page 162](#).
3. Change to the LDAP directory, which is the following by default:

```
C:\Program Files\KnowNow\Server\lib\ldap\
```

4. Execute the Perl command to add users and groups to AD.

```
perl ../../bin/knloadldif ^  
--doit ^  
--ldap_url ldap://AD_server_name ^  
--bind_dn DSadmin_distinguished_name ^  
--password password ^  
--kn_container OU=LiveServer_container,DC=example,DC=com ^  
--user_class user --group_other "groupType: 2147483650" data/knusers_ad.ldff
```

Note the following:

- If you’re connecting in the clear, specify **ldap** instead of **ldaps** for the protocol.

- An initial password is pre-loaded for the **kndaemon** user, and matches one in the default configuration files for LiveServer.
 - *knloadldif* has a lot of options. You can provide them on the command line or by setting them in your environment (without the leading double dashes (--)).
 - The AD security model does not permit password alteration or group configuration in the same script as for user addition. If you see errors on setting passwords and enabling users/groups, you should be able to enable users/groups and reset passwords by using the [Microsoft Active Directory Computers and Users MMC snap-in](#).
5. The LiveServer users and groups should now be visible in ADSI Edit under the LiveServer container.
 6. Additionally, use AD's *Active Directory Users and Computers* program to validate the following. To validate, for each group, right-click on the group name, choose **Properties**, and edit the **member** attribute. The user names should be displayed. If all is well, cancel out of the window.
 - Validate that **kndaemon** and **kninternal** are members of **KnowNow_Internal**.
 - Validate that **knadmin** is a member of **KnowNow_Administrators**.
 - Validate **kndaemon**, **kninternal**, and **knadmin** to make sure they have attributes for a sAMAccountName and password. You'll need a snap-in for this.

Task 6: Configure the LiveServer's LDAP Module

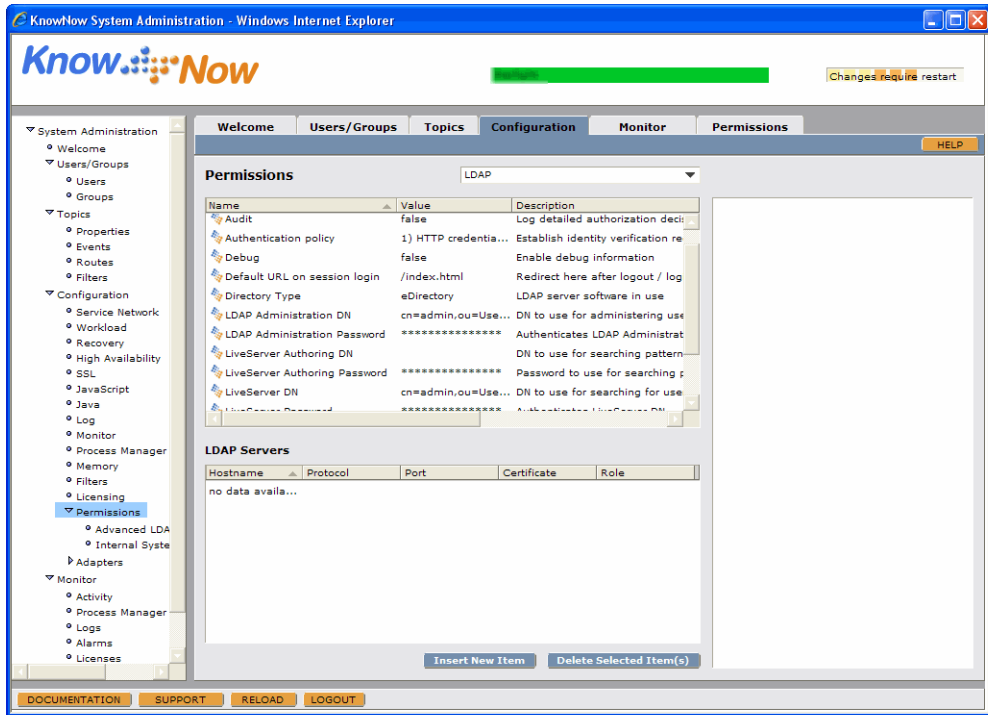
Once you've populated the schema and the objects, you are ready to configure the LiveServer's LDAP module.

To select and configure the LDAP module,

1. Access the KnowNow System Administration console.
2. Under **Configuration**, select **Permissions**. The Permissions configuration panel is displayed.

- From the drop-down list, select **LDAP**. When you do so, the parameters for configuring LDAP appear.

Figure 7-1. The LDAP parameters.



- For each parameter you wish to configure, select that parameter. A field appears to the right for that parameter. Modify the value of the parameter in that field, then choose **Update**. Set the parameters as shown below. The meanings and values of the parameters are provided under “LDAP and LDAP Test Parameters” on page 225.
 - Set **Directory Type** to **Active Directory**
 - Set **LDAP Administration DN** to the **DSadmin** DN
 - Set **LDAP Administration Password** to the password for the **DSadmin**
 - Set **LiveServer Authoring DN** to the DN for **LSadmin**
 - Set **LiveServer DN** to the **DSadmin** DN
 - Set **LiveServer Password** to the password for the **LSadmin**
 - Set **Root Distinguished Name** to the base container for searching in AD

5. Add an AD server. Choose **Insert New Item** in the lower pane. A set of fields appears to the right. Set the parameters as shown here, then choose **Insert**.

For the AD server,

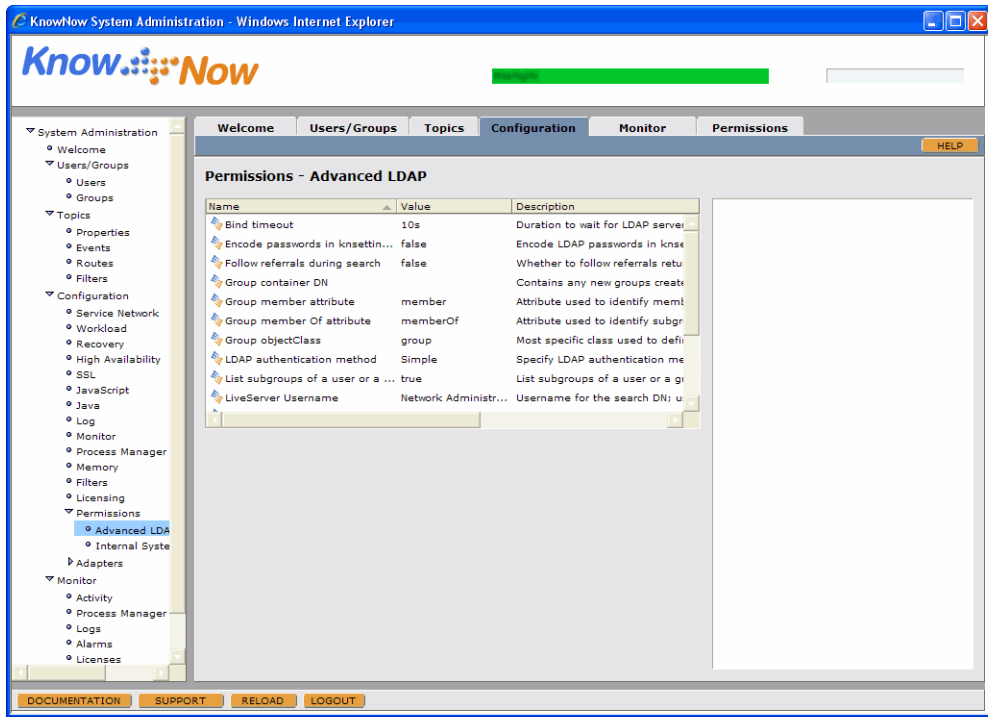
- Set **Hostname** to *AD_server_name (or IP address)*
 - Set **Protocol** to **LDAP**
 - Set **Port** to *AD_server_port*
 - In the **Role** field, enter **authentication** (you will need to manually enter this; make sure you enter it exactly like this, all lowercase, as it is case sensitive)
6. Add an OpenLDAP server. Choose **Insert New Item** in the lower pane. A set of fields appears to the right. Set the parameters as shown here, then choose **Insert**.

For the OpenLDAP server,

- Set **Hostname** to *OpenLDAP_server_name (or IP address)*
- Set **Protocol** to **LDAP**
- Set **Port** to *OpenLDAP_server_port*
- In the **Role** field, enter **authorization** (you will need to manually enter this; make sure you enter it exactly like this, all lowercase, as it is case sensitive)

7. Expand the Permissions category and select **Advanced LDAP Configuration**.

Figure 7-2. The Advanced LDAP panel.



8. Set the **User identifier attribute** parameter to **sAMAccountName**.
9. To make your values effective, restart LiveServer.
10. Once LiveServer has restarted, close down the browser and re-access the LiveServer opening screen. You will be prompted for network authentication. Enter **LSadmin** and the **LSadmin** user's password, after which you will see the opening screen.

Task 7: Validate Your Configuration

The following steps describe how to validate your configuration.

To validate your configuration,

1. Connect to LiveServer and perform a Check Health test from the LiveServer opening screen.

2. Using the KnowNow System Administration console's Users/Groups tab, you should be able to browse for LDAP users and groups and check their permissions.

Deploying ESS in a Dual LDAP Environment: AD With ADAM

The tasks in this section are for deploying ESS in a dual LDAP environment that uses AD for authentication and ADAM for authorization. **This deployment is recommended for ESS deployed on Windows servers.**

The tasks to perform for this environment are as follows. For background information on each of the tasks provided in this section, see “Background Information on LDAP Integration Tasks” on page 164.

1. “Task 1: Install ADAM with the Partition Name O=ess,C=knadam”
2. “Task 2: Create an ESS Container and ESS Admin User in ADAM” on page 180
3. “Task 3: Load LiveServer Schema Extensions in ADAM” on page 181
4. “Task 4: Load the LiveServer Users and Groups in AD” on page 181
5. “Task 5: Load the Permission Patterns in ADAM” on page 183
6. “Task 6: Configure LiveServer’s LDAP Module” on page 184
7. “Task 7: Validate Configuration” on page 186

Task 1: Install ADAM with the Partition Name O=ess,C=knadam

The ADAM examples in this configuration use the sample ADAM installation provided in the *Microsoft Windows Server 2003 R2 Step-by-Step Guide to Deploying ADAM*.

If you have not already done so, install ADAM now. In your ADAM installation, for this example, it is assumed that you will do the following:

- For Windows 2003 Server and Windows XP, install Windows Service Pack 2.
- Install *MS-User.ldf*, *Ms-InetOrgPerson.ldf*, and *Ms-UserProxy.ldf*.
- For lab configurations: Use a standalone server. Other useful tips for lab configurations are available at Microsoft’s ADAM site.
- When you install ADAM, create an instance with an application directory partition for ESS. This example uses **o=ess,c=knadam** as the partition name in ADAM.

Task 2: Create an ESS Container and ESS Admin User in ADAM

In this task, you are adding a user that will be able to administer the permission patterns in the authorizing database, which is **o=ess,c=knadam** for this example. The DN of this user is used in the **LiveServer Authoring DN** parameter.

To create an ESS admin user in ADAM,

1. To relax the SSL, in ADAM ADSI Edit,
 - a. Navigate to
CN=Configuration > CN=Services > CN=Windows NT > CN=Directory Service
 - b. Right-click on **CN=Directory Service** and choose **Properties**.
 - c. Edit **DSHeuristics**. Enter **000000001001**, then choose **OK**.
2. Under the application directory partition (**o=ess,c=knadam**), create an organizational unit (OU) named **liveserver**.
3. Create a user in the **liveserver** OU named **LSadmin**.
4. Give the **LSadmin** user a password by right-clicking on the user's entry in the right pane in ADSI Edit and selecting **Reset Password**.
5. Copy the complete DN for this user.
6. Give the **LSadmin** user administrative privileges. Click **CN=Roles**, right-click **CN=Administrators**, choose **Properties**, then edit the **member** attribute. Choose **Add ADAM Account**, then enter the complete distinguished name:
CN=LSadmin,OU=liveserver,o=ess,c=knadam
7. Enable the user. Select **OU=liveserver**, then right-click on **CN=LSadmin**. Select **Properties** and set the value of **msDS-UserAccountDisabled** to **False**.
8. Use the *ldp* tool from the ADAM Tools command prompt to validate that you can connect and bind with the user. You must enter the complete distinguished name in the **User** field when binding:
CN=LSadmin,OU=liveserver,o=ess,c=knadam

Task 3: Load LiveServer Schema Extensions in ADAM

In this step, you will need to log in using a DN with administration rights on the configuration partition of the server. In ADAM, this is easiest if you're logged into Windows with the same credentials as when the ADAM instance was created.

The schema fragment for ADAM is provided in the LiveServer installation in the file named *adam_schema_extension.ldf*, which is located in C:\Program Files\KnowNow\Server\lib\ldap\schema\ (the location of the LiveServer installation directory may differ).

This file must be copied from the LiveServer installation to the \Windows\ADAM\ directory on the ADAM server.

To load the schema, execute the following command from the ADAM Tools command prompt.

```
C:\WINDOWS\ADAM> ldifde -i -u -f adam_schema_extension.ldf -s localhost:389 -k -j
. -c "cn=Configuration,dc=X" "#configurationNamingContext"
```

The **-c** option and its two arguments are very important. They will take care of changing occurrences of **dc=X** in the file into the appropriate root DN for your directory.

You will use ADAM's native *ldifde* tool for the import.

Typical output from *ldifde* for a successful load:

```
C:\WINDOWS\ADAM> ldifde -i -u -f adam_schema_extension.ldf -s localhost:389 -k -j
. -c "cn=Configuration,dc=X" "#configurationNamingContext"
Connecting to "localhost:389"
Logging in as current user using SSPI
Importing directory from file "c:\Program
Files\KnowNow\Server\lib\ldap\schema\adam_schema_extension_example.ldf"
Loading entries.....
22 entries modified successfully.
```

```
The command has completed successfully
C:\WINDOWS\ADAM>
```

Task 4: Load the LiveServer Users and Groups in AD

In order to function, ESS requires four specific LiveServer users belonging to two specific LiveServer groups (as discussed under [“Create the LiveServer Users and Groups” on page 167](#)).

In AD, create a container to hold those users and groups. This task can should be performed by the AD administrator; it can also be done by using the Perl script provided by KnowNow.

The following instructions are for creating the container using the Perl script. This example uses **OU=liveserver,DC=example,DC=com**.

On the AD server,

1. In AD, add a user whose DN will be used to bind to Active Directory to query the authentication database. In this example, we call this user **DSadmin**.
2. Set your Perl environment variables as described under “[Perl Environment Requirements](#)” on page 162.
3. Change to the LDAP directory, which is the following by default:

```
C:\Program Files\KnowNow\Server\lib\ldap\
```

4. Execute the Perl command to add users and groups to AD.



Warning: If you choose to copy and paste commands from this documentation, first, *copy and paste only from the help files*, not from the PDF. Second, be careful: when copying from HTML, certain editors may embed trailing blanks, carriage returns, or line breaks, rendering the command ineffective. In addition, apostrophes may be incorrectly transcribed as beginning single quotes.

```
perl ../../bin/knloadldif ^
--doit ^
--ldap_url ldap://AD_server_name ^
--bind_dn AD_administrative_DN ^
--password password ^
--user_class user ^
--group_other "groupType: 2147483650" data/knusers_ad.ldff -kn_groupcontainer
OU=liveserver,dc=example,dc=com
```

Note the following:

- If you're connecting in the clear, specify **ldap** instead of **ldaps** for the protocol.
- An initial password is pre-loaded for the **kndaemon** user, and matches one in the default configuration files for LiveServer.
- *knloadldif* has a lot of options. You can provide them on the command line or by setting them in your environment (without the leading double dashes (--)).

- The AD security model does not permit password alteration or group configuration in the same script as for user addition. If you see errors on setting passwords and enabling users/groups, you should be able to enable users/groups and reset passwords by using the [Microsoft Active Directory Computers and Users MMC snap-in](#).
5. The LiveServer users and groups should now be visible in ADSI Edit under the LiveServer container.
 6. Additionally, use AD's *Active Directory Users and Computers* program to validate the following. To validate, for each group, right-click on the group name, choose **Properties**, and edit the **member** attribute. The user names should be displayed. If all is well, cancel out of the window.
 - Validate that **kndaemon** and **kninternal** are members of **KnowNow_Internal**.
 - Validate that **knadmin** is a member of **KnowNow_Administrators**.
 - Validate **kndaemon**, **kninternal**, and **knadmin** to make sure they have attributes for a sAMAccountName and password. You'll need a snap-in for this.

Note: The AD security model does not permit password alteration or group configuration in the same script as for user addition.

Task 5: Load the Permission Patterns in ADAM

Now use a Perl script to load the permission patterns and authorizations. Using the same directories used in [Task 4: Load the LiveServer Users and Groups in AD](#), enter the following Perl command:



Warning: If you choose to copy and paste commands from this documentation, first, *copy and paste only from the help files*, not from the PDF. Second, be careful: when copying from HTML, certain editors may embed trailing blanks, carriage returns, or line breaks, rendering the command ineffective. In addition, apostrophes may be incorrectly transcribed as beginning single quotes.

```
perl ../../bin/knloadldif --doit ^
--doit --ldap_url ldap://localhost:389 ^
--bind_dn CN=Lsadmin,OU=liveserver,o=ess,c=knadam ^
--password password ^
--kn_container OU=liveserver,o=ess,c=knadam ^
--group_container distinguished_name_in_AD ^
--user_class user ^
--group_other "groupType: 2147483650" data/knobjects.ldff ^
```

```
-name server.example.com
```

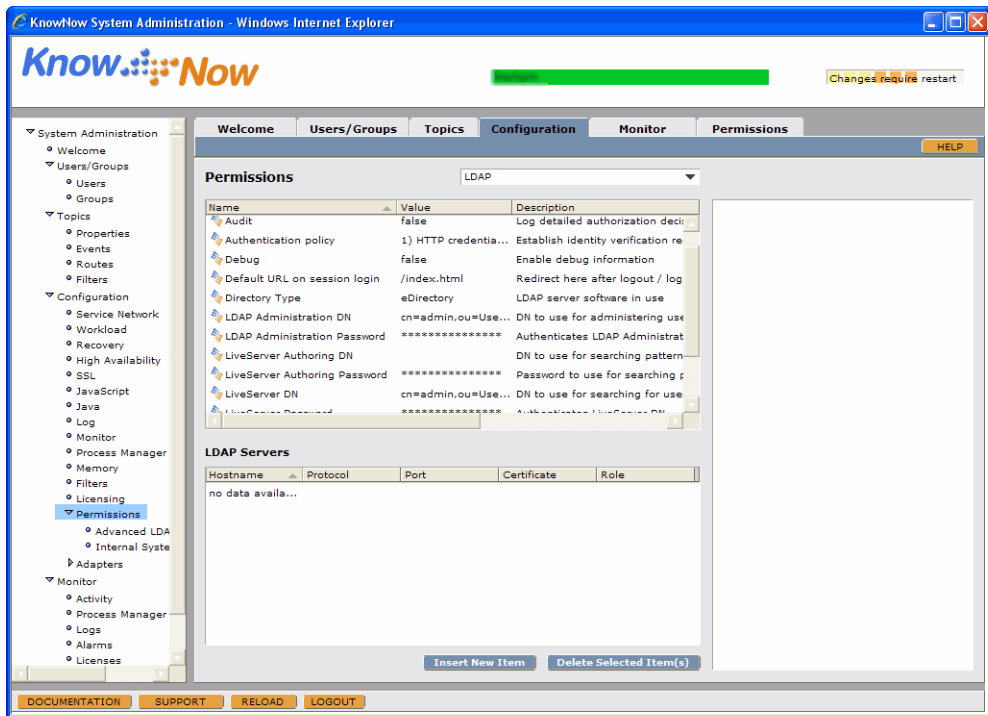
Task 6: Configure LiveServer's LDAP Module

Once you've populated the schema and the objects, you are ready to configure the LiveServer's LDAP module.

To select and configure the LDAP module,

1. Access the KnowNow System Administration console.
2. Under **Configuration**, select **Permissions**. The Permissions configuration panel is displayed.
3. From the drop-down list, select **LDAP**. When you do so, the parameters for configuring LDAP appear.

Figure 7-3. The LDAP parameters.



4. For each parameter you wish to configure, select that parameter. A field appears to the right of that parameter. Modify the value of the parameter in that field, then

choose **Update**. Set the parameters as shown below. The meanings and values of the parameters are provided under “LDAP and LDAP Test Parameters” on page 225.

- Set **Directory Type** to **Active Directory**
- Set **LDAP Administration DN** to the **DSadmin** DN
- Set **LDAP Administration Password** to the password for the **DSadmin**
- Set **LiveServer Authoring DN** to the DN created in [step 3 on page 180](#)
- Set **LiveServer DN** to the **DSadmin** DN
- Set **LiveServer Password** to the password for the **DSadmin**
- Set **Root Distinguished Name** to the base container for searching in AD

5. Add an AD server. Choose **Insert New Item** in the lower pane. A set of fields appears to the right. Set the parameters as shown here, then choose **Insert**.

For the AD server,

- Set **Hostname** to *AD_server_name (or IP address)*
- Set **Protocol** to **LDAP**
- Set **Port** to *AD_server_port*
- In the **Role** field, enter **authentication** (you will need to manually enter this; make sure you enter it exactly like this, all lowercase, as it is case sensitive)

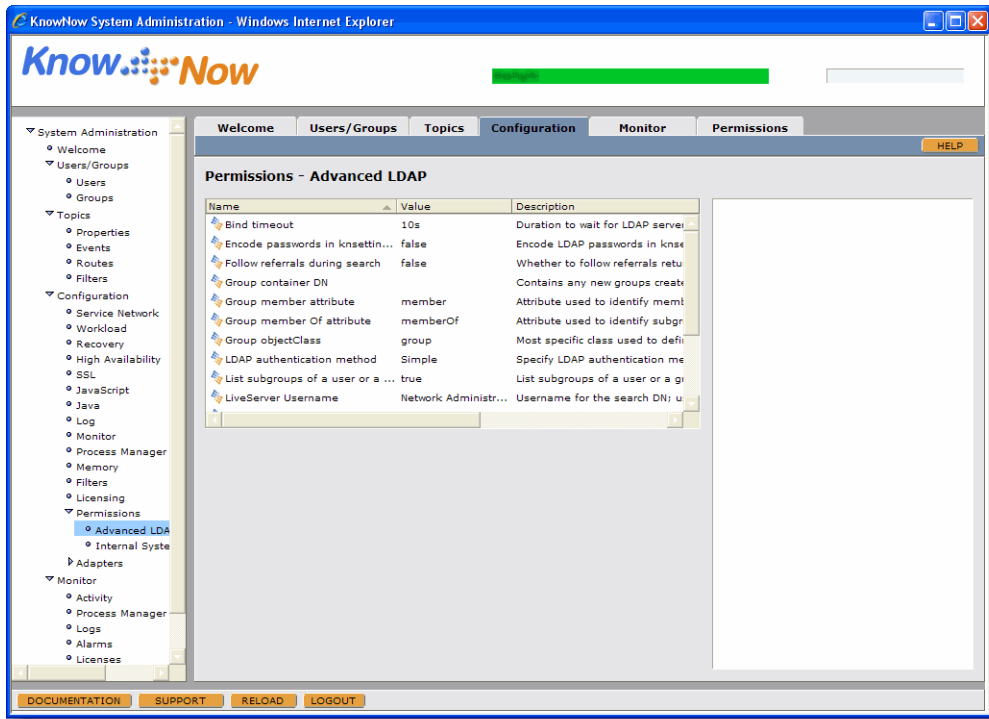
6. Add an ADAM server. Choose **Insert New Item** in the lower pane. A set of fields appears to the right. Set the parameters as shown here, then choose **Insert**.

For the ADAM server,

- Set **Hostname** to *ADAM_server_name (or IP address)*
- Set **Protocol** to **LDAP**
- Set **Port** to *ADAM_server_port*
- In the **Role** field, enter **authorization** (you will need to manually enter this; make sure you enter it exactly like this, all lowercase, as it is case sensitive)

7. Expand the Permissions category and select **Advanced LDAP Configuration**.

Figure 7-4. The Advanced LDAP panel.



8. Set the **User identifier attribute** parameter to **sAMAccountName**.
9. To make your values effective, restart LiveServer.
10. Once LiveServer has restarted, close down the browser and re-access the LiveServer opening screen. You will be prompted for network authentication. Enter **LSadmin** and the **LSadmin** user's password, after which you will see the opening screen.

Task 7: Validate Configuration

The following steps describe how to validate your configuration.

To validate your configuration,

1. Connect to LiveServer and perform a Check Health test from the LiveServer opening screen.

2. Using the KnowNow System Administration console's Users/Groups tab, you should be able to browse for LDAP users and groups and check their permissions.

Deploying ESS in a Dual OpenLDAP Environment

The tasks in this section deploy ESS in a dual OpenLDAP environment that uses one OpenLDAP server for authentication and a second OpenLDAP server for authorization.

The tasks to perform for this environment are as follows. For background information on some of the tasks provided in this section, see “Background Information on LDAP Integration Tasks” on page 164.

1. “Task 1: Install and Validate the OpenLDAP Servers”
2. “Task 2: Create an ESS Container and ESS Admin User in the OpenLDAP Authorization Server” on page 189
3. “Task 3: Load LiveServer Schema Extensions in the OpenLDAP Authorization Server” on page 190
4. “Task 4: Load Permission Patterns in the OpenLDAP Authorization Server” on page 190
5. “Task 5: Load the LiveServer Users and Groups in the OpenLDAP Authentication Server” on page 191
6. “Task 6: Configure the LiveServer’s LDAP Module” on page 192
7. “Task 7: Validate Your Configuration” on page 196

Task 1: Install and Validate the OpenLDAP Servers

Install the OpenLDAP servers according to the instructions for that product. Then validate each OpenLDAP installation by running the following commands on each OpenLDAP machine. The commands given apply to a linux installation.

To validate the OpenLDAP installation on Linux,

1. Validate the LDAP installation and change the run levels using the following commands:

```
>chkconfig --list ldap
>chkconfig --levels 235 ldap on
```

2. Start the OpenLDAP server by running the following command:

```
>service ldap start
```

3. Check the OpenLDAP server health by running the following command:

```
ldapsearch -x -b '' -s base '(objectclass=*)' namingContexts
```

4. In `/etc/openldap/`, edit `ldap.conf` to comment out the default settings.
5. In `/etc/openldap/`, edit `slapd.conf`, replacing the following with the declarations below:

```
suffix          "dc=kn-openldap,dc=com"
rootdn          "cn=admin,dc=kn-openldap,dc=com"
rootpw         password
```

6. Restart OpenLDAP.

```
# service ldap restart
```

Task 2: Create an ESS Container and ESS Admin User in the OpenLDAP Authorization Server

For the openLDAP server that is to be used for authorization, use the `ldapadd` program to add ESS entries to the OpenLDAP directory. `ldapadd` expects input in LDIF form, so you will first create a file named `kn_init_objects.ldif`, run `ldapadd`, and change the password. The user created in this example will be able to administer the permission patterns in the authorizing database. The DN of this user, which in this example is **cn=LSadmin,ou=liveserver,dc=kn-openldap,dc=com**, is used in the **LiveServer Authoring DN** parameter.

To create an ESS container and admin user in OpenLDAP,

1. Use your favorite editor to create an LDIF file named `kn_init_objects.ldif` that contains the following:

```
dn: dc=kn-openldap,dc=com
objectclass: dcObject
objectclass: organization
o: KnowNow
dc: kn-openldap

dn: ou=liveserver,dc=kn-openldap,dc=com
objectclass: organizationalUnit
ou: liveserver

dn: cn=LSadmin,ou=liveserver,dc=kn-openldap,dc=com
objectclass: person
sn: LSadmin
cn: LSadmin
```

2. Run the following command to add the object and user. OpenLDAP prompts for the rootdn password you specified in `slapd.conf`.

```
>ldapadd -x -D "cn=admin,dc=kn-openldap,dc=com" -W -f kn_init_objects.ldif
```

3. Change the password:

```
>ldappasswd -x -v -S -W -D cn=admin,dc=kn-openldap,dc=com \  
cn=LSadmin,ou=liveserver,dc=kn-openldap,dc=com
```

Task 3: Load LiveServer Schema Extensions in the OpenLDAP Authorization Server

LDAP will need a definition of the objects that ESS will request.

To load the LiveServer schema extensions,

1. Stop the OpenLDAP daemon.

```
>service ldap stop
```

2. Copy the file named *knpermission.schema* to the OpenLDAP schema directory:

```
>cp /opt/knserver/lib/ldap/schema/knpermission.schema /etc/openldap/schema/
```

3. Edit *slapd.conf* to extend the schema and to give the needed permissions as follows.

- Insert the following line for the schema extensions:

```
include /etc/openldap/schema/knpermission.schema
```

- Insert the following lines for permissions:

```
access to *  
by users write  
by * read
```

4. Restart the OpenLDAP daemon:

```
>service ldap restart
```

Task 4: Load Permission Patterns in the OpenLDAP Authorization Server

Now run a Perl script to load the permission patterns and authorizations.

```
> cd /opt/knserver/lib/ldap  
  
perl ../../bin/knloadldif \  
--doit \  
--ldap_url ldap:LDAP_URL:port/ \  
--bind_dn DN_for_authorized_user \  
--password user_password \  

```

```
--kn_container DN_for_ESS_container_on_authorization_server \
--group_container DN_for_ESS_container_on_authentication_server \
--user_class inetOrgPerson data/knobjects_openldap.ldif \
--name Server_container_name
```



Warning: If you choose to copy and paste commands from this documentation, first, *copy and paste only from the help files*, not from the PDF. Second, be careful: when copying from HTML, certain editors may embed trailing blanks, carriage returns, or line breaks, rendering the command ineffective. In addition, apostrophes may be incorrectly transcribed as beginning single quotes.

Task 5: Load the LiveServer Users and Groups in the OpenLDAP Authentication Server

In order to function, ESS requires four specific LiveServer users belonging to two specific LiveServer groups (as discussed under [“Create the LiveServer Users and Groups” on page 167](#)).

In the OpenLDAP that you will be using as the authentication server, create a container to hold those users and groups. This task can should be performed by the OpenLDAP administrator; it can also be done by using the Perl script provided by KnowNow. The following instructions tell how to perform this task using the Perl script provided by KnowNow.

On Linux,

1. In OpenLDAP, add a user whose DN will be used to bind to the OpenLDAP server that you will be using for authentication so that the user can query the authentication database. In this example, we call this user **DSadmin**.
2. Set your Perl environment variables as described under [“Perl Environment Requirements” on page 162](#).
3. Change to the LDAP directory, which on Linux is the following by default:

```
/opt/knserver/lib/ldap/
```

4. Execute the Perl command to add users and groups to OpenLDAP.

```
perl ../../bin/knloadldif \
--doit \
--ldap_url ldap:LDAP_URL:port/ \
--bind_dn DN_for_authorized_user \
--password user_password \
```

```
--kn_container DN_for_ESS_container_on_authentication_server \  
--user_class inetOrgPerson \  
--group_class groupOfNames data/knusers_openldap.ldff
```

Note the following:

- If you're connecting in the clear, specify **ldap** instead of **ldaps** for the protocol.
 - An initial password is pre-loaded for the **kndaemon** user, and matches one in the default configuration files for LiveServer.
 - *knloadldif* has a lot of options. You can provide them on the command line or by setting them in your environment (without the leading double dashes (--)).
5. The LiveServer users and groups should now be visible under the LiveServer container.
 6. Use *ldp* or a similar program to validate the following.
 - Validate that **kndaemon** and **kninternal** are members of **KnowNow_Internal**.
 - Validate that **knadmin** is a member of **KnowNow_Administrators**.
 - Validate **kndaemon**, **kninternal**, and **knadmin** to make sure they have attributes for a **uid** and password.

Task 6: Configure the LiveServer's LDAP Module

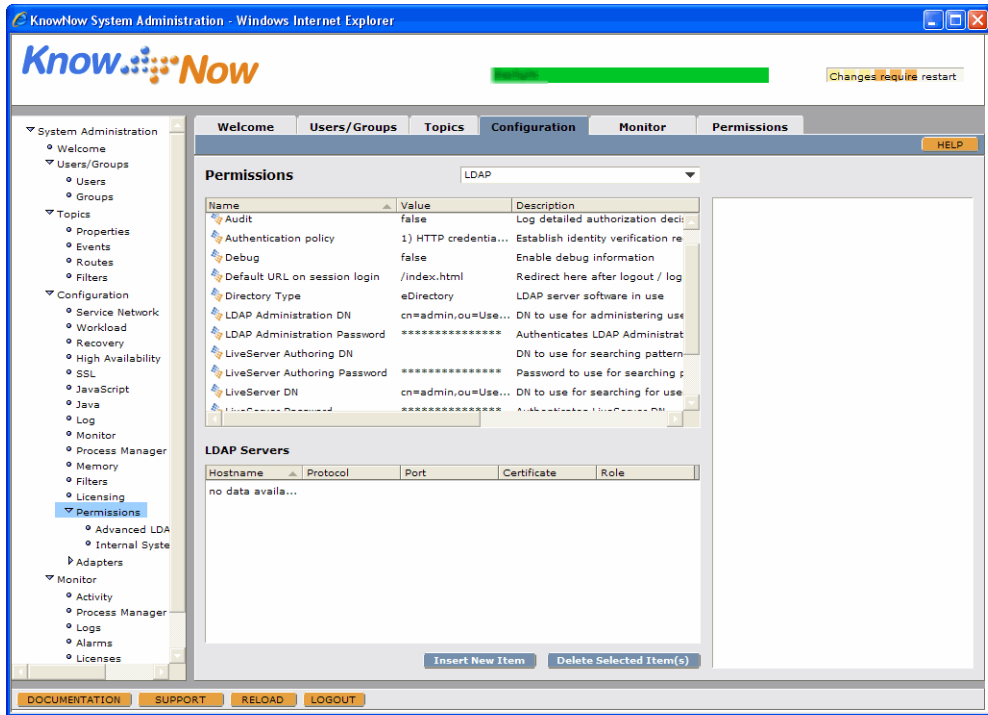
Once you've populated the schema and the objects, you are ready to configure the LiveServer's LDAP module.

To select and configure the LDAP module,

1. Access the KnowNow System Administration console.
2. Under **Configuration**, select **Permissions**. The Permissions configuration panel is displayed.

- From the drop-down list, select **LDAP**. When you do so, the parameters for configuring LDAP appear.

Figure 7-5. The LDAP parameters.



- For each parameter you wish to configure, select that parameter. A field appears to the right for that parameter. Modify the value of the parameter in that field, then choose **Update**. Set the parameters as shown below. The meanings and values of the parameters are provided under “LDAP and LDAP Test Parameters” on page 225.
 - Set **Directory Type** to **OpenLDAP**
 - Set **LDAP Administration DN** to the **DSadmin** DN
 - Set **LDAP Administration Password** to the password for the **DSadmin**
 - Set **LiveServer Authoring DN** to the DN for **LSadmin**
 - Set **LiveServer DN** to the **DSadmin** DN
 - Set **LiveServer Password** to the password for the **LSadmin**
 - Set **Root Distinguished Name** to the base container for searching in AD

5. Add the OpenLDAP authentication server. Choose **Insert New Item** in the lower pane. A set of fields appears to the right. Set the parameters as shown here, then choose **Insert**.

For the OpenLDAP authentication server,

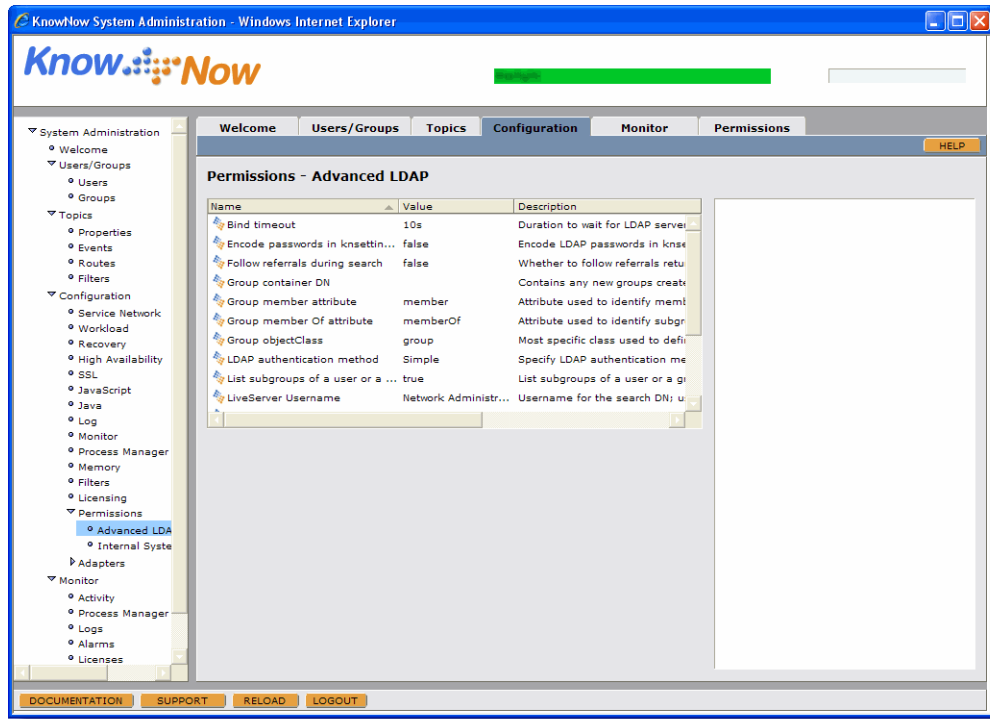
- Set **Hostname** to *OpenLDAP_authentication_server_name* (or IP address)
 - Set **Protocol** to **LDAP**
 - Set **Port** to *OpenLDAP_authentication_server_port*
 - In the **Role** field, enter **authentication** (you will need to manually enter this; make sure you enter it exactly like this, all lowercase, as it is case sensitive)
6. Add the OpenLDAP authorization server. Choose **Insert New Item** in the lower pane. A set of fields appears to the right. Set the parameters as shown here, then choose **Insert**.

For the OpenLDAP authorization server,

- Set **Hostname** to *OpenLDAP_authorization_server_name* (or IP address)
- Set **Protocol** to **LDAP**
- Set **Port** to *OpenLDAP_authorization_server_port*
- In the **Role** field, enter **authorization** (you will need to manually enter this; make sure you enter it exactly like this, all lowercase, as it is case sensitive)

7. Expand the Permissions category and select **Advanced LDAP Configuration**.

Figure 7-6. The Advanced LDAP panel.



8. Set the following parameters:

- Set the **Group member Of attribute** parameter to **isMemberOf** (if you are using this value).
- Set the **Group objectClass** parameter to **groupofNames**.
- Set the **User identifier attribute** parameter to **uid**.

9. To make your values effective, restart LiveServer.

10. Once LiveServer has restarted, close down the browser and re-access the LiveServer opening screen. You will be prompted for network authentication. Enter **LSadmin** and the **LSadmin** user's password, after which you will see the opening screen.

Task 7: Validate Your Configuration

The following steps describe how to validate your configuration.

To validate your configuration,

1. Connect to LiveServer and perform a Check Health test from the LiveServer opening screen.
2. Using the KnowNow System Administration console's Users/Groups tab, you should be able to browse for LDAP users and groups and check their permissions.

Deploying ESS in a Standalone OpenLDAP Environment

This section provides step-by-step instructions for deploying ESS in a standalone AD environment. The tasks to perform for this environment are as follows. For background information on each of the tasks provided in this section, see “Background Information on LDAP Integration Tasks” on page 164.

1. Task 1: Install and Validate the OpenLDAP Server
2. “Task 2: Create an ESS Container and ESS Admin User” on page 198
3. “Task 3: Load LiveServer Schema Extensions in the OpenLDAP Authorization Server” on page 199
4. “Task 4: Load the LiveServer Users and Groups” on page 199
5. “Task 5: Load the Permission Patterns” on page 200
6. “Task 6: Configure LiveServer’s LDAP Module” on page 201
7. “Task 7: Validate Configuration” on page 204

Task 1: Install and Validate the OpenLDAP Server

Install the OpenLDAP server according to the instructions for that product. Then validate your OpenLDAP installation by running the following commands on the OpenLDAP machine. The commands given apply to a linux installation.

To validate the OpenLDAP installation on Linux,

1. Validate the LDAP installation and change the run levels using the following commands:

```
>chkconfig --list ldap  
>chkconfig --levels 235 ldap on
```

2. Start the OpenLDAP server by running the following command:

```
>service ldap start
```

3. Check the OpenLDAP server health by running the following command:

```
ldapsearch -x -b '' -s base '(objectclass=*)' namingContexts
```

4. In `/etc/openldap/`, edit `ldap.conf` to comment out the default settings.

5. In `/etc/openldap/`, edit `slapd.conf`, replacing the following with the declarations below:

```
suffix          "dc=kn-openldap,dc=com"
rootdn          "cn=admin,dc=kn-openldap,dc=com"
rootpw         password
```

6. Restart OpenLDAP.

```
# service ldap restart
```

Task 2: Create an ESS Container and ESS Admin User

In OpenLDAP, first create an organizational unit (OU) named **liveserver**; this is a container for the ESS objects. Note the fully qualified distinguished name (DN) for this OU (*LiveServer-Distinguished-Name*).

To accomplish these tasks you can use the *ldp* tool, which is part of the Windows Support Tools, which can be installed from the Windows Server 2003 installation CD. You can also use Apache Directory Studio's LDAP browser or a similar tool. Or, if you wish, you can use the *ldapadd* tool; for instructions that assume you are using that tool, see ["Task 2: Create an ESS Container and ESS Admin User in the OpenLDAP Authorization Server" on page 189](#).

The user created in the following example will be able to create, delete, and modify objects in the LiveServer application partition.

To create an ESS container and admin user in OpenLDAP,

1. Create an organizational unit (OU) named **liveserver**.
2. Create a user in the **liveserver** OU named **LSadmin**.
3. Give the **LSadmin** user a password.
4. Give the **LSadmin** user administrative privileges.
5. Make sure the account is enabled.
6. Validate that you can connect and bind with the user.

Task 3: Load LiveServer Schema Extensions in the OpenLDAP Authorization Server

LDAP will need a definition of the objects that ESS will request.

To load the LiveServer schema extensions,

1. Stop the OpenLDAP daemon.

```
>service ldap stop
```

2. Copy the file named *knpermission.schema* to the OpenLDAP schema directory:

```
>cp /opt/knserver/lib/ldap/schema/knpermission.schema /etc/openldap/schema/
```

3. Edit *slapd.conf* to extend the schema and to give the needed permissions as follows.

- Insert the following line for the schema extensions:

```
include /etc/openldap/schema/knpermission.schema
```

- Insert the following lines for permissions:

```
access to *  
by users write  
by * read
```

4. Restart the OpenLDAP daemon:

```
>service ldap restart
```

Task 4: Load the LiveServer Users and Groups

In order to function, ESS requires four specific LiveServer users belonging to two specific LiveServer groups (as discussed under [“Create the LiveServer Users and Groups” on page 167](#)).

In OpenLDAP, create a container to hold those users and groups. This task can should be performed by the OpenLDAP administrator; it can also be done by using the Perl script provided by KnowNow. The following instructions tell how to perform this task using the Perl script provided by KnowNow.

On Linux,

1. In OpenLDAP, add a user whose DN will be used to bind to the OpenLDAP server so that the user can query the authentication database. In this example, we call this user **DSadmin**.

2. Set your Perl environment variables as described under “Perl Environment Requirements” on page 162.
3. Change to the LDAP directory, which on Linux is the following by default:

```
/opt/knserver/lib/ldap/
```

4. Execute the Perl command to add users and groups to OpenLDAP.

```
perl ../../bin/knloadldif \  
--doit \  
--ldap_url ldap:LDAP_URL:port/ \  
--bind_dn DN_for_authorized_user \  
--password user_password \  
--kn_container DN_for_ESS_container_on_authentication_server \  
--user_class inetOrgPerson \  
--group_class groupOfNames data/knusers_openldap.ldff
```

Note the following:

- If you’re connecting in the clear, specify **ldap** instead of **ldaps** for the protocol.
 - An initial password is pre-loaded for the **kndaemon** user, and matches one in the default configuration files for LiveServer.
 - *knloadldif* has a lot of options. You can provide them on the command line or by setting them in your environment (without the leading double dashes (--)).
5. The LiveServer users and groups should now be visible under the LiveServer container.
 6. Use *ldp* or a similar program to validate the following.
 - Validate that **kndaemon** and **kninternal** are members of **KnowNow_Internal**.
 - Validate that **knadmin** is a member of **KnowNow_Administrators**.
 - Validate **kndaemon**, **kninternal**, and **knadmin** to make sure they have attributes for a **uid** and password.

Task 5: Load the Permission Patterns

Now run a Perl script to load the permission patterns and authorizations.

```
> cd /opt/knserver/lib/ldap  
  
>perl ../../bin/knloadldif \  
--doit \  
--ldap_url ldap:LDAP_URL:port/ \  
--bind_dn DN_for_authorized_user \  

```

```
--password user_password \  
--kn_container DN_for_ESS_container_on_authorization_server \  
--user_class inetOrgPerson data/knobjects_openldap.ldif \  
-name Server_container_name
```



Warning: If you choose to copy and paste commands from this documentation, first, *copy and paste only from the help files*, not from the PDF. Second, be careful: when copying from HTML, certain editors may embed trailing blanks, carriage returns, or line breaks, rendering the command ineffective. In addition, apostrophes may be incorrectly transcribed as beginning single quotes.

Task 6: Configure LiveServer's LDAP Module

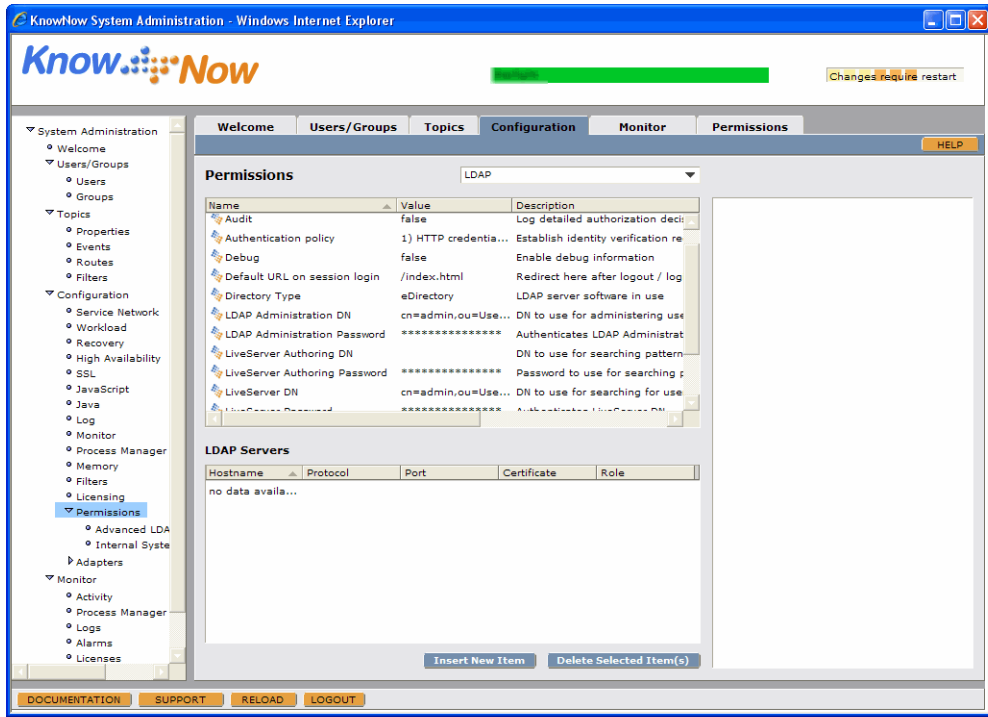
Once you've populated the schema and the objects, you are ready to configure the LiveServer's LDAP module.

To select and configure the LDAP module,

1. Access the KnowNow System Administration console.
2. Under **Configuration**, select **Permissions**. The Permissions configuration panel is displayed.

- From the drop-down list, select **LDAP**. When you do so, the parameters for configuring LDAP appear.

Figure 7-7. The LDAP parameters.

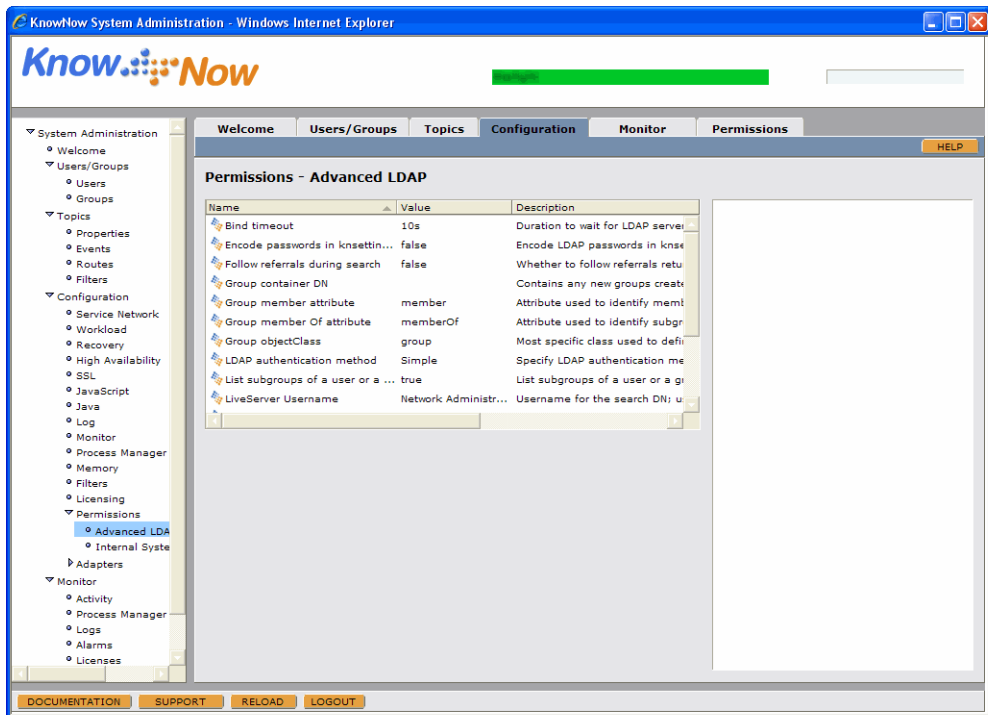


- For each parameter you wish to configure, select that parameter. A field appears to the right for that parameter. Modify the value of the parameter in that field, then choose **Update**. Set the parameters as shown below. The meanings and values of the parameters are provided under “LDAP and LDAP Test Parameters” on page 225.

- Set **Directory Type** to **OpenLDAP**
- Set **LDAP Administration DN** to *LSadmin_distinguished_name*
- Set **LDAP Administration Password** to the *password* for the **LSadmin** user
- Set the **LiveServer Authoring DN** to *LSadmin_distinguished_name*
- Set **LiveServer DN** to *LSadmin_distinguished_name*
- Set **LiveServer Password** to the *password* for the **LSadmin** user
- Set **Root Distinguished Name** to *ESS_container_distinguished_name*

5. Add an OpenLDAP server. Choose **Insert New Item** in the lower pane. A set of fields appears to the right. Set the parameters as shown here, then choose **Insert**.
 - Set **Hostname** to *OpenLDAP_server_name (or IP address)*
 - Set **Protocol** to **LDAP**
 - Set **Port** to *OpenLDAP_server_port*
 - For a single server, do not enter anything in the **Role** field
6. Expand the Permissions category and select **Advanced LDAP Configuration**.

Figure 7-8. The Advanced LDAP panel.



7. Set the following parameters:
 - Set the **Group member Of attribute** parameter to **isMemberOf** (if you are using this value).
 - Set the **Group objectClass** parameter to **groupofNames**.
 - Set the **User identifier attribute** parameter to **uid**.
8. To make your values effective, restart LiveServer.

9. Once LiveServer has restarted, close down the browser and re-access the LiveServer opening screen. You will be prompted for network authentication. Enter **LSadmin** and the **LSadmin** user's password, after which you will see the opening screen.

Task 7: Validate Configuration

The following steps describe how to validate your configuration.

To validate your configuration,

1. Connect to LiveServer and perform a Check Health test from the LiveServer opening screen.

Using the KnowNow System Administration console's Users/Groups tab, you should be able to browse for LDAP users and groups and check their permissions. On the left, click **Users**, then enter the **LSadmin** user ID. You should see the **LSadmin** user (the one you created in [Task 1: Create an ESS Container and ESS Admin User](#)). If you don't see that user, you will need to debug your configuration.

Deploying ESS in a Standalone AD Environment

This section provides step-by-step instructions for deploying ESS in a standalone AD environment. The AD examples in this section are based on the assumption that you have the following installed: *MS-User.ldf*, *Ms-InetOrgPerson.ldf*, and *Ms-UserProxy.ldf*.

The tasks to perform for this environment are as follows. For background information on each of the tasks provided in this section, see “Background Information on LDAP Integration Tasks” on page 164.

1. “Task 1: Create an ESS Container and ESS Admin User”
2. “Task 2: Load LiveServer Schema Extensions” on page 206
3. “Task 3: Load the LiveServer Users and Groups” on page 207
4. “Task 4: Load the Permission Patterns” on page 210
5. “Task 5: Configure LiveServer’s LDAP Module” on page 211
6. “Task 6: Validate Configuration” on page 214
7. “Task 7: Reduce the LSadmin User’s Privileges” on page 214

Task 1: Create an ESS Container and ESS Admin User

In AD, first create an organizational unit (OU) named **liveserver**; this is a container for the ESS objects. Note the fully qualified distinguished name (DN) for this OU (*LiveServer-Distinguished-Name*).

The following instructions refer to the ADSI Edit and *ldp* tools. These tools are both part of the Windows Support Tools, which can be installed from the Windows Server 2003 installation CD.

The user created in the following example will be able to create, delete, and modify objects in the LiveServer application partition.

To create an ESS container and admin user in AD,

1. Using AD’s *Active Directory Users and Computers* program, create an organizational unit (OU) named **liveserver**.
2. Create a user in the **liveserver** OU named **LSadmin**.
3. Give the **LSadmin** user a password by right-clicking the user in *Active Directory Users and Computers* and selecting **Reset Password**.

4. Give the **LSadmin** user administrative privileges by right-clicking on the user, choosing **Properties**, then choosing the **Member Of** tab. Add the *Administrators* group to the user.
5. Make sure the account is enabled by right-clicking on the user. If you see **Enable Account**, the account is disabled and needs to be enabled. To enable the account, click on **Enable Account**.
6. Use the *ldp* tool to validate that you can connect and bind with the user. You must enter the complete distinguished name (available from ADSI Edit) that is entered when binding.

Task 2: Load LiveServer Schema Extensions

In this step, you will need to log in using a DN with administration rights on the configuration partition of the server.

The schema fragment for AD is provided in the LiveServer installation in the file named *ad_schema_extension.ldf*, which is located by default in the following places (the location of the LiveServer installation directory may differ):

- **Windows:** C:\Program Files\KnowNow\Server\lib\ldap\schema\
- **Linux:** /opt/knserver/lib/ldap/schema/

This file (*ad_schema_extension.ldf*) must be copied from the LiveServer installation.

To load the schema, execute the following command from the command prompt.

```
C:> ldifde -i -u -f ad_schema_extension.ldf -s localhost:389 -k -j . -c
"cn=Configuration,dc=X" "#configurationNamingContext"
```

The **-c** option and its two arguments are very important. They will take care of changing occurrences of **dc=X** in the file into the appropriate root DN for your directory.

You will use AD's native *ldifde* tool for the import.

Depending on the AD configuration, an error can occur when running the above command:

```
Add error on line 15: Invalid DN Syntax
The server side error is "The object name has bad syntax."
0 entries modified successfully.
An error has occurred in the program
```

If this error occurs, the command needs to be reissued with the *configurationNamingContext* replaced with the exact DN for the configuration naming context. For example,

```
C:> ldifde -i -u -f ad_schema_extension.ldf -s localhost:389 -k -j . -c
"cn=Configuration,dc=X" "CN=Configuration,DC=example,DC=com"
```

where the correct naming context is given by `DC=example,DC=COM`.

Typical output from `ldifde` for a successful load:

```
C:> ldifde -i -u -f ad_schema_extension.ldf -s localhost:389 -k -j . -c
"cn=Configuration,dc=X" "#configurationNamingContext"
Connecting to "localhost:389"
Logging in as current user using SSPI
Importing directory from file "c:\Program
Files\KnowNow\Server\lib\ldap\schema\adam_schema_extension_example.ldf"
Loading entries.....
22 entries modified successfully.
```

```
The command has completed successfully
C:>
```

Task 3: Load the LiveServer Users and Groups

In order to function, ESS requires four specific users belonging to two specific groups (as discussed under [“Create the LiveServer Users and Groups” on page 167](#)).

In AD, create a container to hold those users and groups. This task can should be performed by the AD administrator; it can also be done by using the KnowNow-provided Perl script on the LiveServer machine, as follows.

On Windows,

1. Set your Perl environment variables as described under [“Perl Environment Requirements” on page 162](#).
2. Change to the LDAP directory, which is the following by default:

```
C:\Program Files\KnowNow\Server\lib\ldap
```

3. Execute the Perl command to add users and groups.

```
perl ..\..\bin\knloadldif ^
--doit ^
--ldap_url ldap://LDAP_server:port ^
--bind_dn LSadmin_distinguished_name ^
--password password ^
--kn_container ESS_container_distinguished_name ^
--name ESS-Server.example.com ^
--user_class user ^
--group_other "groupType: 2147483650" data\knusers_ad.ldff
```

Note the following:

- All the values for following parameters need to be changed to your values: **ldap-url, bind_dn, password, name, and kn_container**.
- If you're connecting in the clear, specify ldap instead of ldaps for the protocol.
- An initial password is pre-loaded for the **kndaemon** user, and matches one in the default configuration files for LiveServer.
- *knloadldif* has a lot of options. You can provide them on the command line, or by setting them in your environment (without the leading double dashes (--)).
- The AD security model does not permit password alteration or group configuration in the same script as for user addition. Therefore, you may see the following errors when running this command:

```
[knloadldif] Error: Could not change
cn=knadmin,OU=liveserver,DC=training,DC=com: 0000001F: SvcErr: DSID-
031A0FC0, problem 5003 (WILL_NOT_PERFORM), data 0
[knloadldif] Error: Could not change
cn=knadmin,OU=liveserver,DC=training,DC=com: 0000052D: SvcErr: DSID-
031A0FC0, problem 5003 (WILL_NOT_PERFORM), data 0
[knloadldif] Error: Could not change
cn=kndaemon,OU=liveserver,DC=training,DC=com: 0000001F: SvcErr: DSID-
031A0FC0, problem 5003 (WILL_NOT_PERFORM), data 0
[knloadldif] Error: Could not change
cn=kndaemon,OU=liveserver,DC=training,DC=com: 0000052D: SvcErr: DSID-
031A0FC0, problem 5003 (WILL_NOT_PERFORM), data 0
done. There were 4 errors.
```

If these errors occur, you *must* enable users/groups and reset passwords by using the [Microsoft Active Directory Computers and Users MMC snap-in](#). Specifically, you must enable the following users: **~KnowNowDummyUser, knadmin, kndaemon, and kninternal**. You may reset the **knadmin** user's password, but you **MUST NOT** modify the **kndaemon** and **kninternal** users' passwords.

4. The LiveServer users and groups should now be visible in ADSI Edit under **OU=liveserver**. In addition, using the ldap tool, you should be able to connect and bind with the **knadmin** user. In order to bind, you must use the full DN for **knadmin**.
5. Additionally, use AD's *Active Directory Users and Computers* program to:
 - validate that **kndaemon** and **kninternal** are members of **KnowNow_Internal**.
 - validate that **knadmin** is a member of **KnowNow_Administrators**.

- validate **kndaemon**, **kninternal**, and **knadmin** to make sure they have attributes for a sAMAccountName and password. You'll need a snap-in for this.

Note: The AD security model does not permit password alternation or group configuration in the same script as for user addition.

On Linux,

1. Set your Perl environment variables as described under “Perl Environment Requirements” on page 162.
2. Change to the LDAP directory, which is the following by default:

```
# cd /opt/knserver/lib/ldap
```

3. Execute the Perl command to add users and groups.

```
perl ../../bin/knloadldif \
--doit \
--ldap_url ldap://LDAP_server:port \
--bind_dn LSadmin_distinguished_name \
--password password \
--kn_container ESS_container_distinguished_name \
--name ESS-Server.example.com ^
--user_class user \
--group_other "groupType: 2147483650" data/knusers_ad.ldff
```

Note the following:

- All the values for following parameters need to be changed to your values: **ldap-url**, **bind_dn**, **password**, **name**, and **kn_container**.
- If you're connecting in the clear, specify `ldap` instead of `ldaps` for the protocol.
- An initial password is pre-loaded for the **kndaemon** user, and matches one in the default configuration files for LiveServer.
- `knloadldif` has a lot of options. You can provide them on the command line, or by setting them in your environment (without the leading double dashes (--)).
- The AD security model does not permit password alteration or group configuration in the same script as for user addition. Therefore, you may see the following errors when running this command:

```
[knloadldif] Error: Could not change
cn=knadmin,OU=liveserver,DC=training,DC=com: 0000001F: SvcErr: DSID-
031A0FC0, problem 5003 (WILL_NOT_PERFORM), data 0
[knloadldif] Error: Could not change
cn=knadmin,OU=liveserver,DC=training,DC=com: 0000052D: SvcErr: DSID-
031A0FC0, problem 5003 (WILL_NOT_PERFORM), data 0
```

```
[knloadldif] Error: Could not change
cn=kndaemon,OU=liveserver,DC=training,DC=com: 0000001F: SvcErr: DSID-
031A0FC0, problem 5003 (WILL_NOT_PERFORM), data 0
[knloadldif] Error: Could not change
cn=kndaemon,OU=liveserver,DC=training,DC=com: 0000052D: SvcErr: DSID-
031A0FC0, problem 5003 (WILL_NOT_PERFORM), data 0
done. There were 4 errors.
```

If these errors occur, you *must* enable users/groups and reset passwords by using the [Microsoft Active Directory Computers and Users MMC snap-in](#). Specifically, you must enable the following users: **~KnowNowDummyUser**, **knadmin**, **kndaemon**, and **kninternal**. You may reset the **knadmin** user's password, but you **MUST NOT** modify the **kndaemon** and **kninternal** users' passwords.

4. The LiveServer users and groups should now be visible in ADSI Edit under **OU=liveserver**. In addition, using the ldp tool, you should be able to connect and bind with the **knadmin** user. In order to bind, you must use the full DN for **knadmin**.
5. Additionally, use AD's *Active Directory Users and Computers* program to:
 - validate that **kndaemon** and **kninternal** are members of **KnowNow_Internal**.
 - validate that **knadmin** is a member of **KnowNow_Administrators**.
 - validate **kndaemon**, **kninternal**, and **knadmin** to make sure they have attributes for a sAMAccountName and password. You'll need a snap-in for this.

Note: The AD security model does not permit password alternation or group configuration in the same script as for user addition.

Task 4: Load the Permission Patterns

Using the same directories used in [Task 3: Load the LiveServer Users and Groups](#), enter the following Perl command (noting that the values for all the following parameters need to be changed to your values: **ldap-url**, **bind_dn**, **password**, **name**, and **kn_container**):

For Windows:

```
perl ..\..\bin\knloadldif ^
--doit ^
--ldap_url ldap://LDAP_server:port ^
--bind_dn LSadmin_distinguished_name \
--password password ^
--name ESS-Server.example.com ^
--kn_container ESS_container_distinguished_name data/knobjects.ldff
```

For Linux:

```
perl ../../bin/knloadldif \
--doit \
--ldap_url ldap://LDAP_server:port \
--bind_dn LSadmin_distinguished_name \
--password password \
--name ESS-Server.example.com \
--kn_container ESS_container_distinguished_name data/knobjects.ldff
```

Task 5: Configure LiveServer's LDAP Module

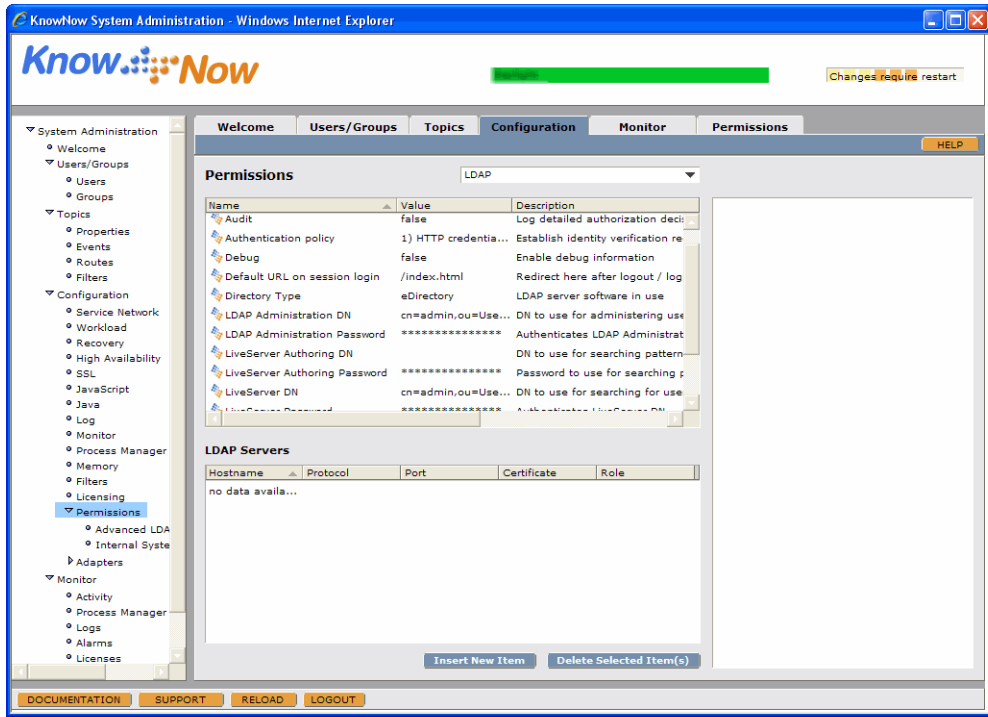
Once you've populated the schema and the objects, you are ready to configure the LiveServer's LDAP module.

To select and configure the LDAP module,

1. Access the KnowNow System Administration console.
2. Under **Configuration**, select **Permissions**. The Permissions configuration panel is displayed.

- From the drop-down list, select **LDAP**. When you do so, the parameters for configuring LDAP appear.

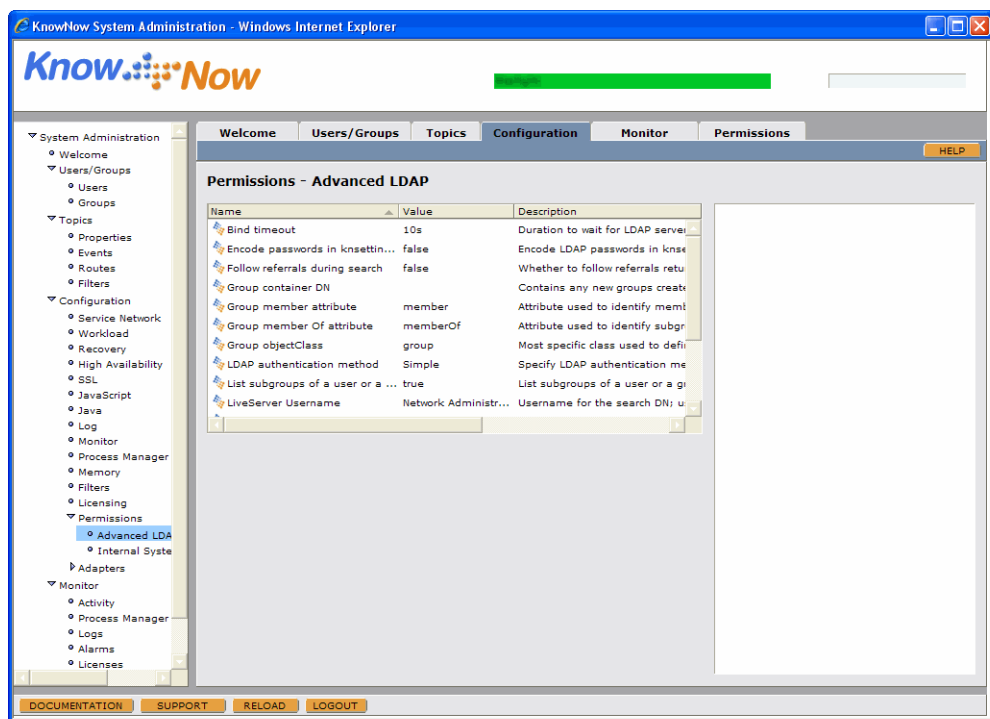
Figure 7-9. The LDAP parameters.



- For each parameter you wish to configure, select that parameter. A field appears to the right for that parameter. Modify the value of the parameter in that field, then choose **Update**. Set the parameters as shown below. The meanings and values of the parameters are provided under “LDAP and LDAP Test Parameters” on page 225.
 - Set **Directory Type** to **Active Directory**
 - Set **LDAP Administration DN** to *LSadmin_distinguished_name*
 - Set **LDAP Administration Password** to the *password* for the **LSadmin** user
 - Set the **LiveServer Authoring DN** to *LSadmin_distinguished_name*
 - Set **LiveServer DN** to *LSadmin_distinguished_name*
 - Set **LiveServer Password** to the *password* for the **LSadmin** user
 - Set **Root Distinguished Name** to *ESS_container_distinguished_name*

5. Add an AD server. Choose **Insert New Item** in the lower pane. A set of fields appears to the right. Set the parameters as shown here, then choose **Insert**.
 - Set **Hostname** to *AD_server_name* (or IP address)
 - Set **Protocol** to **LDAP**
 - Set **Port** to *AD_server_port*
 - For a single server, do not enter anything in the **Role** field
6. Expand the Permissions category and select **Advanced LDAP Configuration**.

Figure 7-10. The Advanced LDAP panel.



7. Set the **User identifier attribute** parameter to **sAMAccountName**.
8. To make your values effective, restart LiveServer.
9. Once LiveServer has restarted, close down the browser and re-access the LiveServer opening screen. You will be prompted for network authentication. Enter **LSadmin** and the **LSadmin** user's password, after which you will see the opening screen.

Task 6: Validate Configuration

The following steps describe how to validate your configuration.

To validate your configuration,

1. Connect to LiveServer and perform a Check Health test from the LiveServer opening screen.
2. Using the KnowNow System Administration console's Users/Groups tab, you should be able to browse for LDAP users and groups and check their permissions. On the left, click **Users**, then enter the **LSadmin** user ID. You should see the **LSadmin** user (the one you created in [Task 1: Create an ESS Container and ESS Admin User](#)). If you don't see that user, you will need to debug your configuration.

Task 7: Reduce the LSadmin User's Privileges

Perform this task if needed. For instructions, see ["Reducing the LSadmin User's Privileges"](#) on page 215.

Reducing the LSadmin User's Privileges

These instructions show how to reduce the privileges of the **LSadmin** user (using ADSI Edit to do so). These steps ensure that the **LSadmin** user has only the necessary privileges needed to work with LiveServer, so that no excessive privileges are granted for LiveServer deployment in your AD environment.

There are three tasks you will need to perform:

1. Remove the **LSadmin** user from the Domain Admins group to restrict that user's rights.
2. Edit the **LSadmin** user's permissions as desired while granting the **LSadmin** user full control over the LiveServer container.
3. Verify that the **LSadmin** user has full control over the LiveServer container.

Remove the LSadmin User From the Domain Admins

Before using ADSI Edit to grant the required permissions to the LiveServer permission patterns container, ensure that the **LSadmin** user is removed from the Domain Admins group.

To remove the LSadmin user from the Domain Admins group,

1. Using AD's *Active Directory Users and Computers MMC* tool, locate the **LSadmin** user.
2. Right-click on the **LSadmin** user.
3. Choose **Properties**, then choose the Member Of tab.
4. Remove the Domain Admins group from the **LSadmin** user if the **LSadmin** user is in the Domain Admins group.

Edit the LSadmin User's Permissions

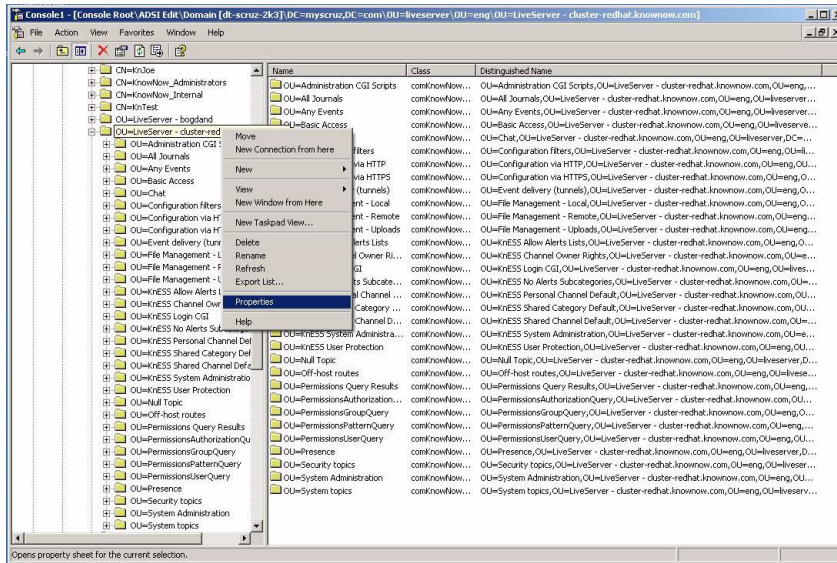
Next, assign the **LSadmin** user full control over the LiveServer container.

To edit the LSadmin user's privileges,

1. Open ADSI Edit.
2. Navigate to the LiveServer permission patterns container.

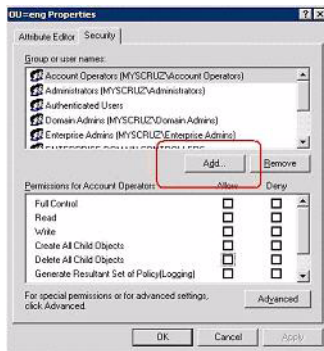
- Right-click on the container and choose **Properties**. A Properties dialog box opens.

Figure 7-11. The Properties menu item.



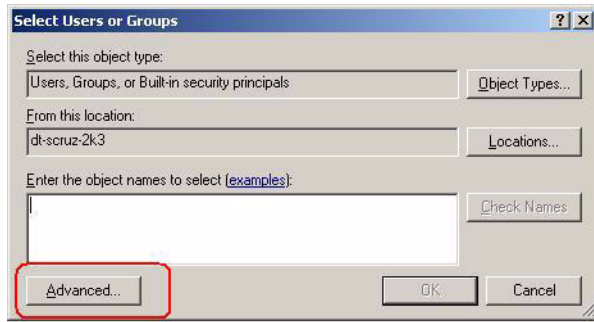
- In the Properties dialog box, access the Security tab. The Security tab is where you will be changing the **Lsadmin** user's permissions; however, first you will need to add the **Lsadmin** user to the list of groups and user names.

Figure 7-12. The Security tab.



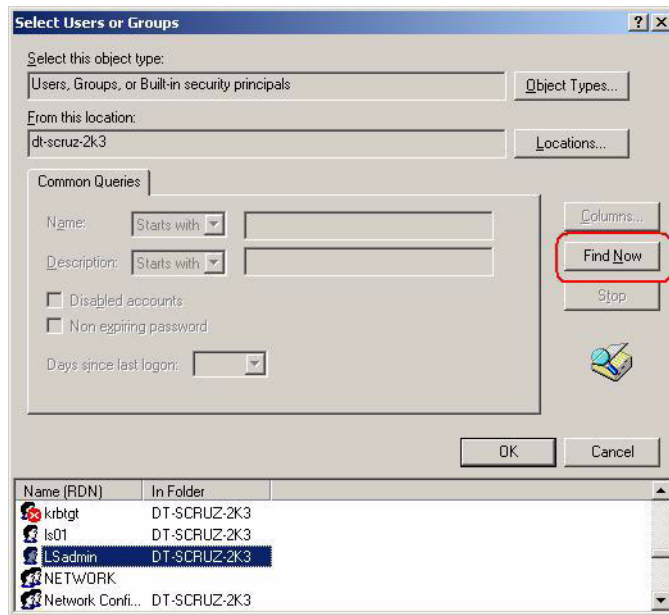
- Choose **Add**. The Select Users or Groups dialog box opens. Enter the **LSadmin** user name. You can click the **examples** link to see examples of object names.

Figure 7-13. Entering the **LSadmin** name.



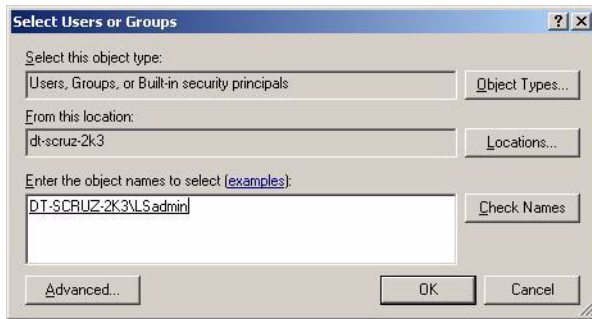
Alternatively, you can have ADSI Edit find the user for you. To search for a user, choose **Advanced...**, then **Find Now** to see a list of AD users and groups. You can select the user from the name (RDN) list. You can also specify a search filter in the Common Queries area.

Figure 7-14. Searching for a user.



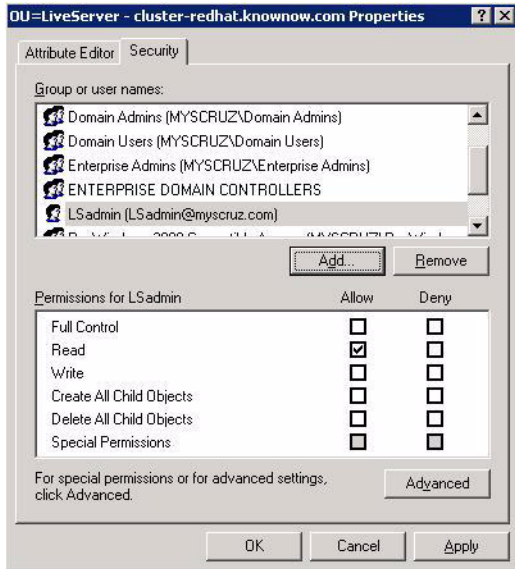
6. Once you've found the **LSadmin** user, choose **OK**. The **LSadmin** user is now in the list of users in the Select Users or Groups dialog box.

Figure 7-15. The LSadmin user has been added to the Select Users or Groups dialog box.



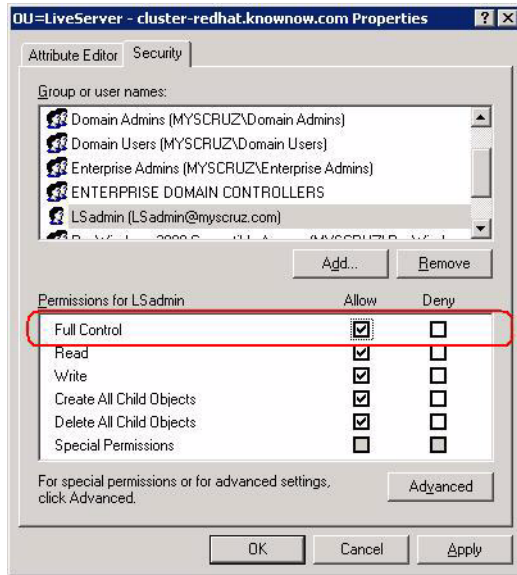
7. Choose **OK**. This closes the Select Users or Groups dialog box and returns you to the Security tab. The **LSadmin** user is now in the Groups or User Names window (representing groups and users who are in the LiveServer permission patterns container ACL).

Figure 7-16. The LSadmin user in the ACL list.



8. Select the **LSadmin** user and grant that user full control to the LiveServer container.

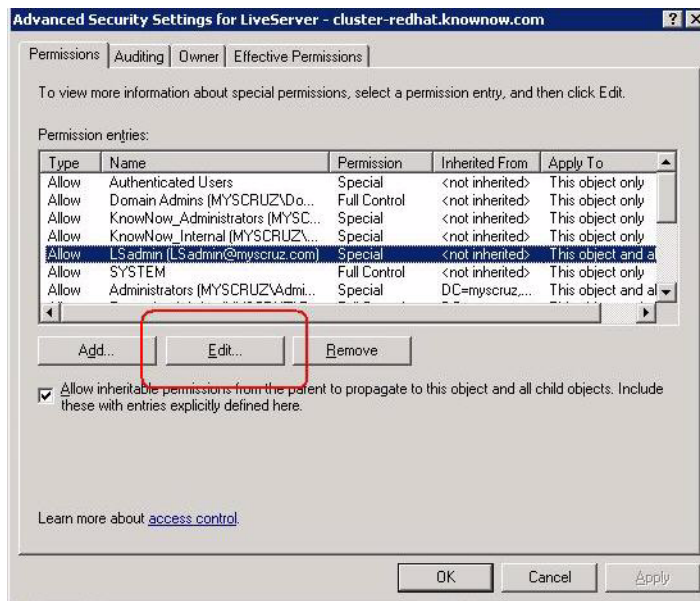
Figure 7-17. Granting the LSadmin user full control to the LiveServer container.



9. Next, make sure that the **LSadmin** can create permission patterns under the LiveServer container. With the user still selected, choose **Advanced**. The Advanced Security Setting for LiveServer - *hostname* dialog box opens.

10. Select the **LSadmin** user, then choose **Edit**.

Figure 7-18. Editing the LSadmin user.

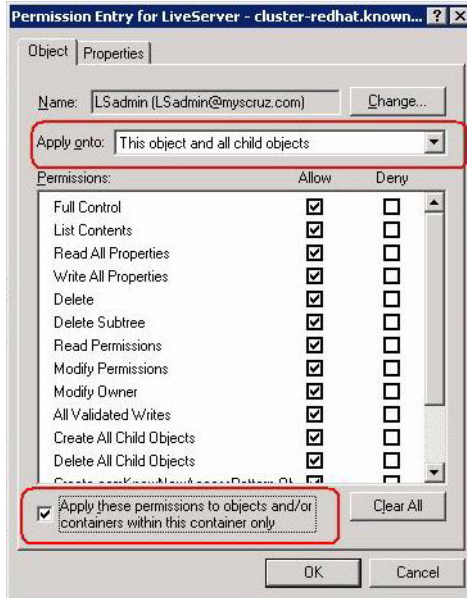


11. The Permission Entry for LiveServer - *hostname* dialog box opens. The following settings should be in place:

- The **Name** is **LSadmin**.
- All permissions are allowed.
- The **Apply onto** combo box points to **This object and all child objects**.

- The **Apply these permissions to objects and/or containers within this container only** option is enabled.

Figure 7-19. Ensuring the proper permissions are assigned.



12. Close each properties dialog box in turn by choosing **OK** for each dialog box.

Verify the LSadmin User's Rights

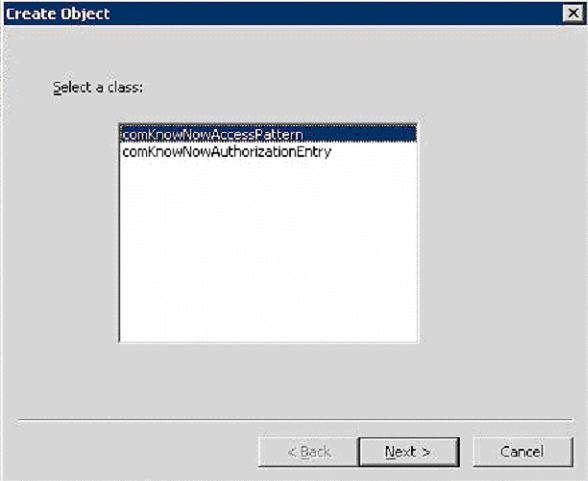
The final task is to verify the updated **LSadmin** user's privileges.

To validate the LSadmin user's privileges,

1. Use ADSI Edit to connect to AD.
2. Log in as the **LSadmin** user.
3. Right-click on the LiveServer permission patterns container and select **New > Object**. This opens the Create Object window. Here you should see that the

LSadmin user has the rights to create the comKnowNowAccessPattern and comKnowNowAuthorizationEntry objects.

Figure 7-20. Verifying the LSadmin user's rights.



Selecting and Configuring an LDAP Module

An extended discussion on integrating LDAP with LiveServer is provided under “[Integrating ESS With Your LDAP Environment](#)” on page 161. One of the tasks is selecting an LDAP module. For the purposes of that discussion, explicit examples with specific parameter settings have been provided.

This section is a more generic description of the tasks involved in selecting and configuring an LDAP module. It can be referred to when you know what your parameter values need to be and you just want a summary of the steps involved in selecting and configuring an LDAP module.

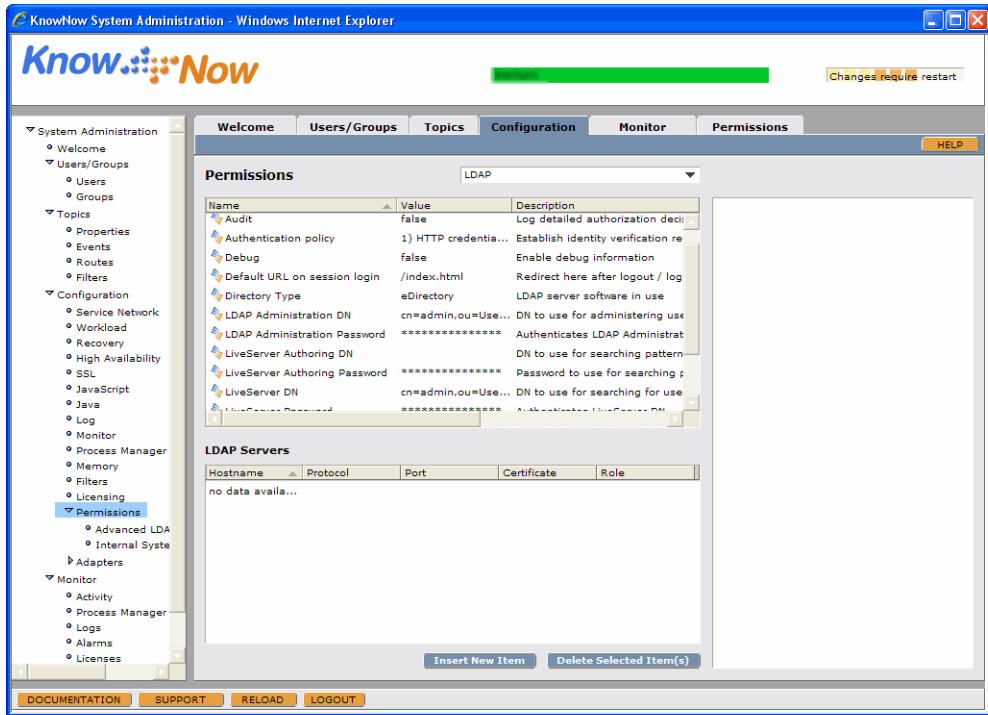
Settings made for LDAP Test parameters are automatically transferred over to the matching parameters of the LDAP module. (Though the LDAP module has a few additional parameters that the LDAP Test module does not have.) Since the settings for LDAP and LDAP Test are identical, the following procedure can be used to select and configure either module.

To select and configure an LDAP module,

1. Access the KnowNow System Administration console.
2. Under **Configuration**, select **Permissions**. The Permissions configuration panel is displayed.

3. From the drop-down list, select **LDAP**. When you do so, the parameters for configuring LDAP appear.

Figure 7-21. The LDAP parameters.



4. For each parameter you wish to configure, select that parameter. A field appears to the right for modifying the value of that parameter. Modify the value of the parameter in the field to the right, then choose **Update**. The meanings and values of the parameters are provided under “LDAP and LDAP Test Parameters” on page 225.
5. For each LDAP server you wish to configure, choose **Insert New Item** in the lower section of the panel. A set of fields appears to the right. Fill in the fields as follows:
 - a. In the **Hostname** field, enter the name of the LDAP server. You can use an IP address or fully qualified name.
 - b. In the **Protocol** field, choose **LDAP** or **LDAP/SSL** (the equivalents of HTTP and HTTPS). **LDAP/SSL** specifies using LDAP over SSL for all operations. If you choose **LDAP** (which is *not* recommended), LiveServer uses clear-text LDAP.
 - c. In the **Port** field, enter the port number for that server.

- d. In the **Certificate** field, enter a certificate for that server. You will need to upload that certificate using this field. If you selected **LDAP/SSL** for your protocol, you must provide a certificate.



Warning: LiveServer requires a copy of the LDAPS:// certificate in either DER or B64 format in order to operate in LDAPS:// mode. (This is not the case for LiveServer operation in straight LDAP:// mode.)

- e. You will need to manually enter one of three things in the **Roles** field: **authentication**, **authorization**, or, in the case of a single server, nothing (an empty field). The field is case sensitive; **authentication** and **authorization** must be in all lowercase letters.
- f. Choose **Insert**. The server appears in the LDAP Servers list in the lower panel. LDAP servers are used in the order in which you add them.

Repeat this step for each LDAP server you wish to add.

6. To delete an LDAP server, select it in the lower panel and choose **Delete Selected Item(s)**. You can select more than one server to delete at a time, and you can select discontinuous servers by pressing and holding the control key while clicking on each choice.
7. Configure additional parameters as needed as described under [“Advanced LDAP Configuration” on page 229](#).
8. Configure permissions patterns as described in [Chapter 8 , “Managing Permissions.”](#)
9. If desired, configure parameters for internal users as described under [“Configuring Internal Users” on page 257](#).
10. If you are done making changes, restart LiveServer so your values can take effect.

LDAP and LDAP Test Parameters

The LDAP parameters configure administrative aspects. Some of these parameters are available only in **LDAP** mode, not in **LDAP Test** mode.

For additional LDAP parameters, see [“Advanced LDAP Parameters” on page 230](#). For an explanation of the format for units in parameters, see [“Entering Data and Editing LiveServer Parameters” on page 53](#).

Audit

When set to **true**, detailed authorization decisions are entered as log entries in the LiveServer's log file, *server.log*. The default value is **false**.

Authentication policy

This field, which applies only to **LDAP** and *not* to **LDAP Test**, offers a number of authentication policies, thus providing a flexible way of configuring authentication policies. It replaces the older **Allow anonymous login** option. For a description of the authentication policies, see [“Authentication policy” on page 154](#).

Debug

Turns on debugging statements in the log. This can be useful when you have **LDAP Test** enabled and are testing your configuration. The default value is **false**.

Default URL on session login

This parameter is for **LDAP** only, *not* for **LDAP Test**.

If you're using certain options in the **Authentication policy** parameter, and a user logs out, then logs in again, the browser is directed to the URL specified in this parameter. This URL is also used in any other case in which a user logs in without having previously presented LiveServer with a non-login URL. (For example, when a user goes directly to <http://example.com/login/login.html> and logs in.) The default value is **/index.html**.

Directory Type

Specifies your LDAP server type. Among the choices are **ActiveDirectory** and **ADAM**.

LDAP Administration DN

The distinguished name (directory) to bind to for administering users, groups, and LiveServer permissions. The default value **cn=admin,ou=Users,ou=demo,dc=knownow,dc=com**. This value should be changed to reflect your particular system.

This user must be able to search the users, groups, and the LiveServer entries, and must also be able to update the LiveServer entries and their children. If you wish to administer users using the KnowNow System Administration console, this user will need to have permission to create, modify, and delete users and groups as well. For information on the administration user, see [“LDAP User Types” on page 254](#).

LDAP Administration Password

The password for the administrator.

LiveServer Authoring DN

The distinguished name to use for creating and searching permission patterns in the authorization database. The default value is **cn=admin,ou=Users,ou=demo,dc=knownow,dc=com**. This value should be changed to reflect your particular system.

LiveServer Authoring Password

Password to use for searching permission patterns.

LiveServer DN

The distinguished name to bind for searching for users, groups, and permissions. LiveServer uses this to find LDAP users and groups. The default value is **cn=admin,ou=Users,ou=demo,dc=knownow,dc=com**. This value should be changed to reflect your particular system.

If you don't want the search user to be the same as the LiveServer admin user, you will need to adjust the LDAP ACIs to allow the search user access to this organizational unit and its children.

LiveServer Password

Provides the set of credentials necessary for LiveServer to use to bind to the authentication LDAP server(s). Authenticates [LiveServer DN](#).

Passwords used throughout the KnowNow system are encoded for security purposes and as such are not human readable.

Open Access URLs

This parameter is for **LDAP** only, *not* for **LDAP Test**.

When any URLs are set in this parameter, and when policy 2, 3, or 5 is in effect, then all readers are allowed open access to those URLs (no cookies are required) unless that access is forbidden by an access pattern. The default value is for this parameter to have no URLs specified. For information on access patterns, see [Chapter 8](#), "Managing Permissions."

Root Distinguished Name

The distinguished name for the container of groups, users, and permissions. The default value is **ou=demo,dc=knownow,dc=com**. This value should be changed to reflect your particular system.

Template for login form

This parameter is for **LDAP** only, *not* for **LDAP Test**.

When selecting certain **Authentication policy** options, users are prompted to log in using a custom login form. LiveServer ships with a default login form; you can customize that form or supply your own custom login form. If you choose to do so, use this parameter to specify the location of that custom form. The default location and file name is either specified by a custom application or is **\${homedir}/runtime/pages/login/login.adp**.

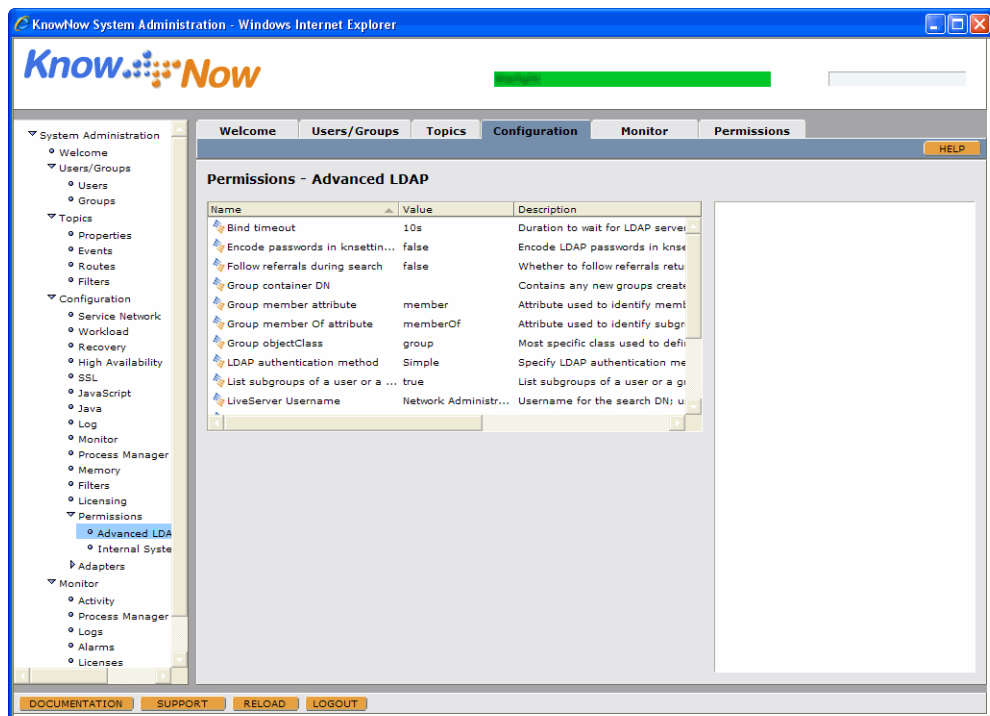
Advanced LDAP Configuration

Once you have selected an LDAP module, you may wish to configure additional parameters.

To configure these additional parameters,

1. Access the KnowNow System Administration console.
2. Under **Configuration**, select **Permissions**. The Permissions configuration panel is displayed. The drop-down list should show your permissions selection, either **LDAP** or **LDAP Test**. The parameters for your selected module should already have been configured as described under “[Selecting and Configuring an LDAP Module](#)” on page 223.
3. Expand the **Permissions** category and select **Advanced LDAP Configuration**.

Figure 7-22. The Advanced LDAP panel.



4. For each parameter you wish to configure, select that parameter. A field appears to the right for modifying the value of that parameter. Modify the value of the

parameter in the field to the right, then choose **Update**. The meanings and values of the parameters are provided under “[Advanced LDAP Parameters](#)” on page 230.

5. If you are done making changes, restart LiveServer.

Advanced LDAP Parameters

The advanced LDAP parameters are defined in this section. The DNs in these parameters are taken by default from the LDAP parameters listed under “[LDAP and LDAP Test Parameters](#)” on page 225. For an explanation of the format for units in parameters, see “[Entering Data and Editing LiveServer Parameters](#)” on page 53.

Bind timeout

The amount of time to wait for the LDAP server to bind a user’s credentials. The default value is **10s** (ten seconds).

Encode passwords in knsettings.conf

Specifies whether to encode LDAP passwords in LiveServer’s *knsettings.conf* configuration file. The default value is **false**.

Follow referrals during search

When enabled, follows referrals during LDAP search. The default value is **false**.

Group container DN

Contains any new groups created by LiveServer. Note that the **LSadmin** user must have the correct privileges in order to create groups.

Group member attribute

Sets the attribute to be used to identify members of a group. The default value is **member**.

Group member Of attribute

Use this variable to specify groups that are members of a group. The default value is **MemberOf**.

Group objectClass

The most specific class used to define groups. The default value is **group**.

LDAP authentication method

Specifies the authentication method to use for LDAP. The default value is **Simple**. RFC 2829 has a very nice introduction to SASL and Simple, along with recommended best practices. See <http://www.ietf.org/rfc/rfc2829.txt>. When selecting the SASL option, you will also need to set the **Realm** parameter.

The options for this parameter are as follows:

- **Simple**. Uses the Simple authentication scheme in LDAP.
- **SASL DIGEST-MD5**. SASL is an acronym for Simple Authentication and Secure Layer. This method prevents clear text passwords from being transferred between LiveServer and the LDAP server when authenticating users. With this enabled, users will need to provide a principle name (user@example.com) and password to authenticate themselves to LDAP servers. User can enter their credentials either with their principle name and password or their user name, password, and domain name.

List subgroups of a user or a group in LDAP queries

When set to true, if a user or group also belongs to subgroups, those will also be displayed in search results. The default value is **true**.

LiveServer Username

This is the user name for the search DN; this is used for SASL bindings.

Permission change poll interval

Persistent search is not implemented in LiveServer. Use this parameter to set how often polling for permissions changes is done. The default value is **5m** (five minutes).

Realm

Specifies the LDAP server's realm. Use this when using the SASL option in the **LDAP authentication method** parameter. The default value is **example.com**.

Trust Login identity

Specifies whether to trust the login identity provided by LiveServer clients during authentication. The default value is **false**.

User container DN

Contains any new users created by LiveServer. Note that the **LSadmin** user must have the correct privileges in order to create users.

User identifier attribute

Attribute used to identify users. The default value is **uid**.

For AD installations, this attribute should be set to the sAMAccountName, which corresponds to the logon user ID. (By default, for AD, the sAMAccountName and the property Microsoft identifies as the user logon name are equivalent. The property can be reset, however.)

User objectClass

The most specific class used to define users. The default value is **person**.

Chapter 8 Managing Permissions

This chapter describes the default access patterns that are shipped with LiveServer and tells how to customize them or create your own permissions patterns. Access patterns are assigned to users and groups to control access to LiveServer areas. They are also a way of controlling access to specific topics.

- [“Creating Access Patterns” on page 234](#)
- [“The Access Pattern Table” on page 236](#)
- [“The LiveServer’s Default Access Patterns” on page 237](#)
- [“Viewing and Editing Access Patterns” on page 241](#)
- [“Creating New Access Patterns” on page 244](#)
- [“Deleting Access Patterns” on page 247](#)

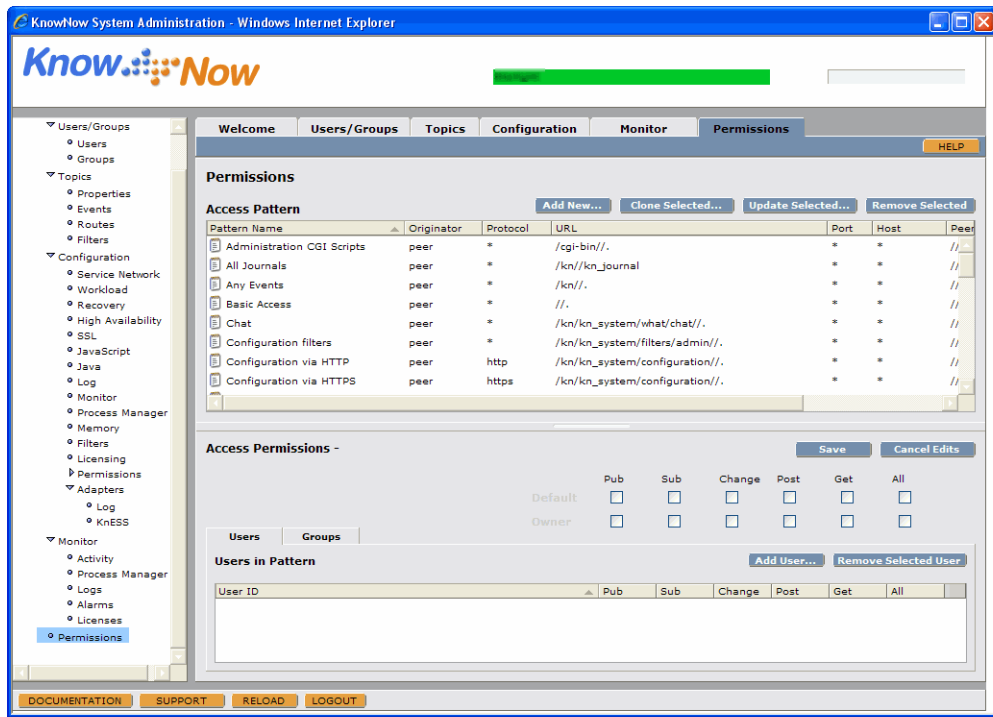
Creating Access Patterns

An access pattern is a way of describing a set of resources that you want to allow access to. Such resources include topics. Another way of defining an access pattern is to say that an access pattern is a way of describing a part of URL space (the way LiveServer sees it; therefore, it supports offhost routes). You can do everything with LiveServer's URL space that you can do with a typical Web space. When access is attempted, LiveServer checks the access patterns to see if there is one that matches the access attempt, and allows or denies the access according to the pattern.

The Permissions tab provides the ability to create and use access patterns. To access the Permissions tab, click on it or select **Permissions** in the navigation tree to the left. The Permissions access patterns panel is displayed.

LiveServer ships with a set of permissions patterns which you can use as is, or which you can use as models for creating your own. In either case, to control access, you add users and groups for any permissions pattern and edit their permissions using the Permissions tab.

Figure 8-1. The Permissions tab.



The checkboxes grant or deny the following permissions for a selected resource to the selected user or group. **Default** refers to the default permissions to use if permissions haven't been specifically assigned to a user or group, and **Owner** refers to the owner of the resource.

- Enabling the **Publish** checkbox grants the ability to publish to the selected resource.
- Enabling the **Subscribe** checkbox grants the ability to subscribe to the selected resource.
- Enabling the **Change** checkbox grants the ability to make changes to events in the selected resource.
- Enabling the **Post** checkbox grants the ability to send Post events to the selected resource.
- Enabling the **Get** checkbox grants send Get events to the selected resource.
- Enabling the **All** checkbox grants all permissions.

The Access Pattern Table

The Access Pattern table displays the current set of patterns. The columns in the Access Patterns table are

- **Pattern Name.** This is the name of the pattern. When creating an access pattern, use a short descriptive phrase that will help you remember what the pattern does. The name of an access pattern is used as a tie-breaker when determining which pattern to execute. For information on how LiveServer determines which access pattern to use, see [“Determining Which Access Pattern Applies” on page 240](#).
- **Originator.** The originator indicates where the access is coming from; that is, who is originating the HTTP/HTTPS access. You can use one of two values in this field: **Peer** or **LiveServer**. Most accesses are initiated by a peer. **LiveServer** is used when LiveServer is delivering offhost.
- **Protocol.** The protocol used to gain access, which can be HTTP or HTTPS.
- **URL.** The URL is the path to which you want to allow access. For example, the following pattern allows access to any URL that starts with /kn and ends with /kn_journal. The period (.) indicates any number of topics.
`/kn/*./kn_journal`
- **Port.** This indicates an IP port on LiveServer.
- **Host.** This indicates the other end of the connection. It could be described using an IP address, an IP subnet, or a domain pattern.
- **Peer URL.** This indicates the URL for the peer. It can be left blank.
- **Peer Port.** This indicates the port for the peer. It can be left blank.

You can change the sort order of each column to ascending or descending by clicking on the column header.

For information on the syntax to use when creating access patterns, see [“Syntax in Access Patterns” on page 238](#).

The LiveServer's Default Access Patterns

LiveServer ships with a set of access patterns that you can use just as they are or that you can modify as desired. Access patterns define a resource that you want to control access to. They take into account where access is coming from, which ports are involved, and what access is being attempted. Whenever a GET, POST, subscribe, publish, or change action takes place on some number of objects, an access pattern is used.

Access patterns are not necessarily configured with users and groups; therefore, you will almost certainly want to add users and groups to the following patterns as described under [“Viewing and Editing Access Patterns” on page 241](#). You can also create a clone of any of the following patterns, then edit the access patterns in that clone as desired. Information on cloning access patterns is provided under [“Creating a New Access Pattern By Cloning” on page 245](#).

The included access patterns are

- Administration CGI Scripts. This pattern can be used to control access to CGI scripts in the CGI bin.
- All Journals. This pattern can be used to control access to all journals.
- Any Events. This important pattern can be used to control access to any event. When configuring users and groups for this pattern, you must allow anyone who is using LiveServer to have access. This pattern is required for LiveServer operations and should not be removed.
- Basic Access. This important pattern can be used to control access to everything. As with the Any Events pattern, when configuring users and groups for this pattern, you must allow anyone who is using LiveServer to have access. This pattern is also required for LiveServer operations and should not be removed.
- Chat. This pattern can be used to control access to the /kn/what/chat topic and to all subtopics beneath it.
- Configuration filters. This pattern can be used to control access to the /kn/kn_system/filters/admin topic and all subtopics beneath that topic. This pattern controls access to the administration files, which you will probably want to carefully control.
- Configuration via HTTP. This pattern can be used to control access to the /kn/kn_system/configuration topic and all topics beneath it when accessing those topics via HTTPS. This pattern controls access to the configuration files, including

where the KnowNow System Administration console accesses and configures parameters, so you will probably want to carefully control this access as well.

- Configuration via HTTPS. This pattern can be used to control access to the `/kn/kn_system/configuration` topic and all topics beneath it when accessing those topics via HTTPS.
- Event delivery (tunnels). This pattern is used to control access to the tunnel routes that belong to journals. Restricting access to these (beyond the restrictions already given) can cause problems with LiveServer applications, including the KnowNow System Administration console; be certain to allow access for administrators before restricting them further.
- File Management - Local, File Management - Remote, and File Management - Uploads. These patterns can be used to control access to the LiveServer's internal process, the filemanager, and to the ability to upload files.
- Off-host routes. This pattern can be used to control access from any URL on this LiveServer to any URL on any other LiveServer. Essentially, this specifies who is allowed to create off-host routes.
- Presence. This pattern can be used to control access to all `/kn_routes` subtopics. (These subtopics can be used to monitor presence; i.e., the ability to find out if someone else is online.)
- Security topics. This pattern can be used to control access to the `/kn/kn_system/permission` topic and all subtopics beneath it. These are the topics used to add and delete users.
- System Administration. This pattern can be used to control access to KnowNow System Administration's URL (`/admin`).
- System topics. This pattern can be used to control access to the `/kn/kn_system` topics. Many of these topics are used by the KnowNow System Administration console to manage LiveServer. When restricting this, remember that all users *must* be able to subscribe to `/kn_system/heartbeat`.

Syntax in Access Patterns

To indicate URLs and topics in access patterns, you can either type in the exact desired path, or you can use certain special characters as wildcards and separators. (In the following discussion, although we use the term "URL," you can also mentally substitute or add the term "topic" as well.) The special characters are the asterisk (*) and a double slash (//). A URL pattern may contain two or more wildcards (and can use both * and //).

The asterisk identifies a set consisting of a collection of child URLs (topics). It replaces any character or set of characters, including the null character, except for a slash (/) and a space. A double asterisk does the same, only it also includes the slash (/). Spaces are only legal in the pattern name itself. If you do not enter anything in one of the access pattern fields (i.e., if you leave it blank), that is the equivalent of “anything.” (You can think of a blank field as the biggest wildcard possible.)

For example, the following pattern identifies all the child URLs of the root URL t1. This URL pattern does not include the root URL t1 itself, and it does not include any grandchildren or further descendents of t1.

```
/kn/t1/*
```

To build up longer paths, you can use wildcard characters in conjunction with fixed child URL names. For example, the following pattern identifies all grandchildren of /kn/t1 that have the name t3.

```
/kn/t1/*/t3
```

As in XPATH, double slashes (//) identify all descendents (subject to the constraints implied by the remainder of the path), not just immediate children. If the URL pattern ends with the //, this indicates that the URL pattern matches a URL subtree. For example, the following pattern identifies a subtree consisting of /kn/t1/t3 and all its descendents.

```
/kn/t1/t3//
```

If the URL pattern ends with a double slash followed by an asterisk (//*), this indicates that the URL pattern matches all the descendents of a topic, but not the topic itself. For example, the following pattern identifies the subtree consisting of the descendents of /kn/t1/t3 but, unlike the previous example, does not include /kn/t1/t3 itself.

```
/kn/t1/t3/**
```

To include all the URLs managed by a LiveServer, use the following pattern:

```
//.
```

Double slashes can also be used in the middle of a URL pattern; for example, the following pattern identifies all descendents of /kn/t1 that have the name t3. This is the key technique to use to control access to journals, for example.

```
/kn/t1//t3
```

Determining Which Access Pattern Applies

When there is a conflict between two or more access patterns, LiveServer finds and uses the most specific pattern. The definition of a more specific pattern is this:

- In general, the fewer wildcards used in the rule, the more specific the rule is.
- If there are the same number of wildcards in two patterns, and a wildcard in one of those patterns expands to cover all paths and a wildcard in the other does not, then the one that does not expand is used.
- If a rule has more items on the right of the rule than another rule does, it is used.
- If no other order of precedence rule works, then the pattern whose name comes first alphabetically is used.

For users and groups, the order of precedence is similar: The most explicit set of permissions is used, with the user-level permissions being the most explicit, followed by the owner, then the group, and finally the default.

Viewing and Editing Access Patterns

You can view and edit the LiveServer's default access patterns as described here, or you can create a new one as described under [“Creating New Access Patterns”](#) on page 244. You could also edit the `access_patterns` file located in the `/conf/` directory; however, we recommend that you use the KnowNow System Administration console instead (it writes to this file directly).

As part of the process of editing access patterns, you will almost certainly want to edit the users and groups for those access patterns. The process of doing so is described here as well.



Caution: Changes made in the Permissions screen apply immediately. Be careful not to remove your own access permissions.

To view and edit an existing pattern,

1. Access the KnowNow System Administration console.
2. Choose **Permissions**. The Permissions access patterns panel is displayed, containing the Access Pattern table.
3. To edit the desired access pattern, select the access pattern and choose **Update Selected**. The Update Existing Pattern dialog box opens. In this dialog box, you can change any aspect of the access pattern except the name of the pattern.

Figure 8-2. Updating an access pattern.

Pattern Name:	Administration CGI Scripts
Originator:	peer
Protocol:	*
URL:	/cgi-bin//.
Port:	*
Host:	*
Peer URL:	//.
Peer Port:	*

For an explanation of each of the fields in this dialog box, see [“The Access Pattern Table”](#) on page 236. For an explanation of the syntax to use in these fields, see [“Syntax in Access Patterns”](#) on page 238.

4. When you have completed the changes, choose **Update**. A success notification window opens; choose **OK** to close it. If an error notification window opens, choose **OK** to close it, then follow the instructions provided to correct the problem.
5. To set access details (i.e., to assign permissions for users and groups for the selected access pattern), double-click on a pattern name in the Access Pattern table. When you double-click a pattern name, the Access Permissions area below the Access Pattern table displays information about that pattern.
6. Set the default and owner access permissions by enabling or disabling the checkboxes. The **Default** checkboxes indicate permissions that everyone added to this pattern would have by default. The **Owner** checkboxes indicate permissions only the owner has. (The owner is whoever created the topic. The owner should have at least as many permissions as the default.) If you enable or disable the **All** check box, the other check boxes in the same row are enabled or disabled as well.

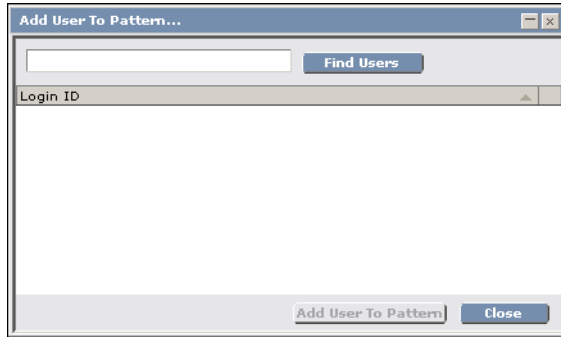
To alter the permissions that a specific user or group in the pattern is to have, add a user or group, and then edit their permissions in the Users or Groups tab as described in [step 7](#) and [step 10](#).

You could also set permissions by editing the `pattern_permissions` file located in the `/conf/` directory; however, we recommend that you use the KnowNow System Administration console to edit permissions instead; the KnowNow System Administration console writes your changes to this file.

7. View and add users and groups by selecting the Users or Groups tab below the Access Pattern table. By adding a user or group to a pattern, you are indicating that that user or group has all the specified permissions. (Users and groups can be defined using the Users/Groups tab, as defined in [Chapter 9](#), "Managing Users and Groups.")
8. To add a user or group, choosing **Add User** (when the User tab is showing) or **Add Group** (when the Groups tab is showing). Enter a user or group name, then choose **Find Users** or **Find Groups**. You must enter the exact name, though if you don't know the exact name, you can use wild cards, such as an asterisk (*). If the user or group is in the system, it will be displayed in the User ID column. When the user or

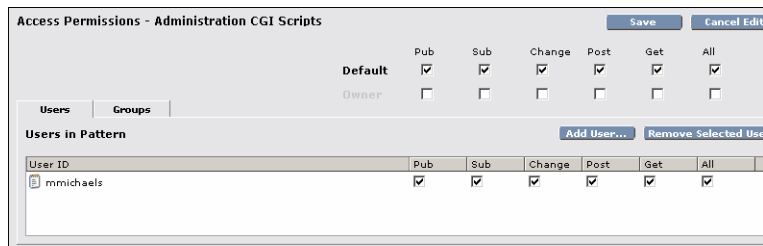
group has been found, if it is the correct one, choose **Add User to Pattern** or **Add Group to Pattern**.

Figure 8-3. Adding a user.



9. You can remove a user or group from the selected pattern by selecting the user or group in the Users or Groups tab and choosing **Remove Selected User** or **Remove Selected Group**.
10. Once you've added a user or group, set the permissions for that specific person or group by enabling or disabling the appropriate checkboxes.

Figure 8-4. A sample user.



11. Once you have made the desired changes, choose **Save**. (Or, if you wish to discard your changes, choose **Cancel Edits**.) If your changes were successful, a success notification is displayed, and then another to indicate that the information was inserted. Choose **OK** to close each notification box.
12. To view your changes, double-click the access pattern name again (in the Access Pattern table). The permitted users and groups show in the Access Permissions area below the Access Patterns table.

Creating New Access Patterns

Although the patterns that ship with LiveServer cover most, if not all, the permissions that you may wish to use, you may wish to create a new set of patterns. You can create a new access pattern entirely from scratch, or you can clone a pattern, then edit it to meet your needs.



Caution: Changes made in this screen apply immediately. Be careful not to remove your own access permissions.

This section describes creating access patterns under the following headings:

- [“Naming Access Patterns”](#)
- [“Creating a New Access Pattern From Scratch”](#)
- [“Creating a New Access Pattern By Cloning” on page 245](#)

Naming Access Patterns

When using the Flat File or LDAP permissions modules, permission pattern names may not contain asterisks (*), newlines, or carriage returns. All other characters (including multibyte) are okay.

Creating a New Access Pattern From Scratch

Creating an access pattern from scratch means to create every aspect of it without using the **Clone Selected** capability. If it would be faster for you to copy an existing pattern and edit it, use the instructions under [“Creating a New Access Pattern By Cloning” on page 245](#).

To create a new access pattern from scratch,

1. Access the KnowNow System Administration console.

2. Under **Permissions**, choose **Add New**. The Create New Pattern dialog box opens.

Figure 8-5. Creating a new access pattern.

3. Fill in the pattern name, originator, protocol, URL, port, host, peer URL, and peer port. Each of these is described under [“The Access Pattern Table” on page 236](#).
4. Choose **Create**. (Or, if you wish to discard your changes, choose **Cancel**.) If your settings were successful, a success notification is displayed. Choose **OK** to close the notification box.
5. Add users and groups and set their permissions as desired as described in [step 7](#) and [step 10 on page 243](#).

Creating a New Access Pattern By Cloning

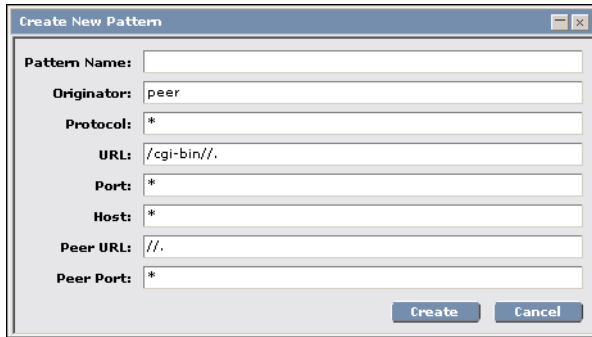
You can use a copy (a clone) of an existing access pattern to create a new pattern. This can save you some time. When you clone an access pattern, the name, access permissions, users, and groups for the pattern you are cloning are **not** cloned; you will need to create those for the new pattern.

To a new access pattern by copying an existing pattern,

1. Access the KnowNow System Administration console.
2. Under **Permissions**, choose **Clone Selected**. The Create New Pattern dialog box opens with the originator, protocol, URL, port, host, peer URL, and peer port already

filled in as they were for the access pattern you are cloning. (These parameters are described under “The Access Pattern Table” on page 236.)

Figure 8-6. Cloning an access pattern.



3. Fill in a name for this new pattern. Make changes as desired to the originator, protocol, URL, port, host, peer URL, and peer port as desired.
4. Choose **Create**. (Or, if you wish to discard your changes, choose **Cancel**.) If your settings were successful, a success notification is displayed. Choose **OK** to close the notification box.
5. Add users and groups and set their permissions as desired as described in [step 7 on page 242](#) and [step 10 on page 243](#).

Deleting Access Patterns

You can delete an access pattern by double-clicking the pattern name in the Access Pattern table and choosing **Remove Selected**. You will **not** be given a warning dialog box; instead, the pattern will be deleted immediately, so make sure you want to delete it before choosing this action.

Chapter 9 Managing Users and Groups

This chapter describes the capabilities of the Users/Groups tab in the KnowNow System Administration console. You can use LiveServer to add, edit, and delete users and groups; however, we assume that in many cases, you may have your own software for managing users and groups. If you are using LDAP, the LiveServer's users and groups capabilities can tie into your system so that you can manage users and groups using either LiveServer or your own software; changes made using your software will show up in LiveServer and vice versa.

Once you've created users and groups, you can control their access by assigning them to access patterns as described in [Chapter 8, "Managing Permissions."](#)

For information on using different permissions modules, see [Chapter 6, "Selecting a Permissions System"](#) and [Chapter 7, "Implementing Security with LDAP."](#)

The topics covered in this chapter are:

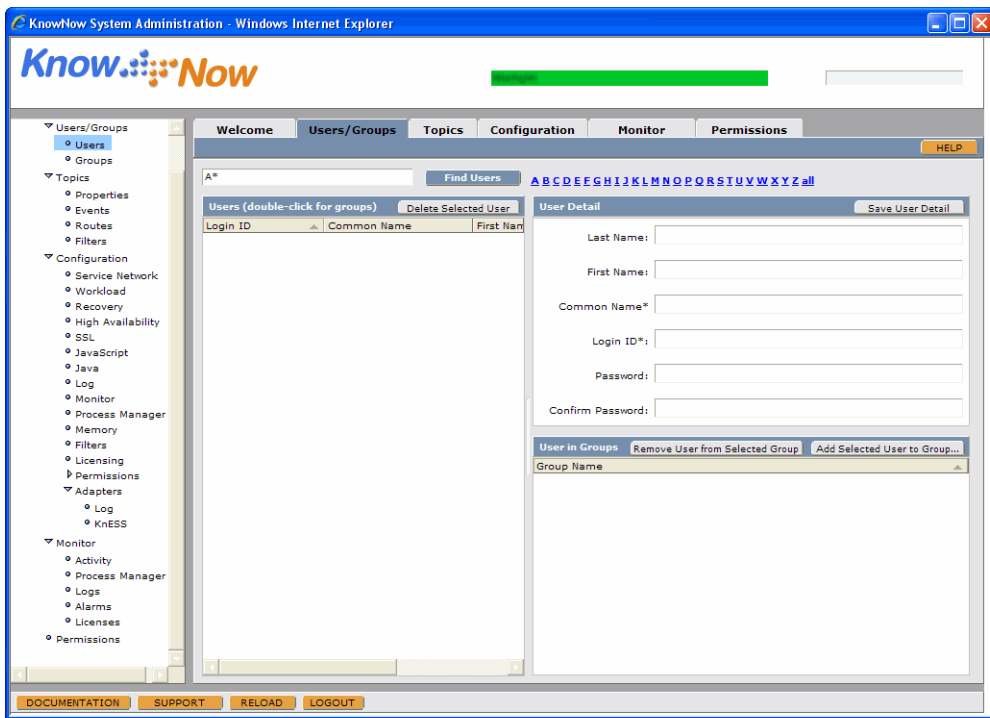
- ["Introducing the Users/Groups Tab"](#) on page 250
- ["Managing Users"](#) on page 252
- ["Managing Groups"](#) on page 260

Introducing the Users/Groups Tab

When you access the KnowNow System Administration console and select **Users/Groups**, the Users/Groups panel is displayed. This panel is where you configure users and groups.

If you don't want to use the KnowNow System Administration console to manage your users and groups, you can use an LDAP editor. You can use either the KnowNow System Administration console or an LDAP editor or both. Any changes you make using an LDAP editor or other such tool will be reflected in the KnowNow System Administration console and in LiveServer. For more on using LDAP, see [Chapter 7, "Implementing Security with LDAP"](#)

Figure 9-1. The Users/Groups tab in users mode.



The Users/Groups tab has a user mode and a group mode, and consists of several distinct areas. When you have **Users** selected in the navigation tree to the left, you are in user mode. When you have **Groups** selected in the navigation tree to the left, you are in groups mode. Depending on which mode you are in,

- The field in the top left is for searching for a user or group by name. You can use wildcards in the search, such as * for zero or more letters.
- The hyperlinked alphabetical list to the right can be used to quickly access all users or groups whose names start with a certain letter.
- The panel to the left contains a list of all users by login ID, first name, and last name or all groups by group name, group ID, and description. You can sort the list by clicking on any one of the three columns.
- The **User Detail** panel (the top panel to the right in users mode) provides editable details on a specific user. The **Group Detail** panel (the top panel to the right in groups mode) provides editable details on a specific group.
- The **User in Groups** panel (the bottom panel to the right in user mode) lists users by group. This panel is replaced by the **Users in Group** panel when in group mode lists the users in the selected group. When in either mode, you can use the **Add User to Group** button to add a user to a group.

In all cases, fields marked with an asterisk (*) are required.

Managing Users

In order to be able to use the Users/Groups tabs, you must have selected one of the following permissions options: **Flat Files**, **LDAP**, or **LDAP Test**. For information on the permissions options and how to select and configure them, see [Chapter 6](#), “[Selecting a Permissions System](#).”

This section provides information on users under the following headings:

- “[The LiveServer’s Default Users and Groups](#)”
- “[LDAP User Types](#)” on page 254
- “[Adding, Finding, and Deleting Users](#)” on page 255
- “[Internal System Users Parameters](#)” on page 258

The LiveServer’s Default Users and Groups

LiveServer is configured with the following default users and groups, each with a different purpose and scope.

- The **~KnDummy** user is used only in the KnowNow System Administration console; it never logs in. This user is used to create LDAP groups (since LDAP groups can’t be empty).
- The **kndaemon** user is used for all of the administration daemons (knlogd, knauthd, and filemanager) and all .cgis within LiveServer. This user must be able to publish and subscribe to the /kn_system/permissions/**, /kn_system/nodes/*/log, /kn_system/nodes/*/alarms, /kn_system/filemanager/**, and /kn_system/alarms topics, and also must be allowed to run any .cgi in the /cgi-bin/*.cgi URL space. This user must have a valid password that can be bound on. The scripts that use **kndaemon** may be configured to use either HTTP or HTTPS. They will automatically trust the LiveServer’s certificate for HTTP/SSL operations. For information on changing this user’s name and password, see “[Configuring Internal Users](#)” on page 257.
- The **kninternal** user is for KnowNow’s C++ internal modules (including cluster communication and recovery), and is the owner of any topics created by internal modules. This user never binds to LDAP. Note that **kninternal** is not allowed for use across a network or interprocess communication channel of any kind, since it is not protected by any credential system.
- The **knadmin** user is for administrators. This user is set up to be in the KnowNow_Administrators group, and is therefore allowed to administer LiveServer. This user has rights to configure LiveServer, subscribe to all topics, add,

change, and delete users, notify to the top level topics, and create new topics at the top level.

- The **kn-demo** user is for guest users. This user has the rights to create and to publish to topics in `/what/demo`. This user is not installed in the LDAP backend.
- Members of the **KnowNow_Administrators** group are allowed to administer the allowed to administer LiveServer.
- Members of the **KnowNow_Internal** group include the **kndaemon** and **kninternal** users. This group must be allowed to work in the `/kn_system//` topic space and to create journals for themselves.

The following information applies to these users:

- All users have rights to create journals in `/who/s`.
- Both **kninternal** and **kndaemon** must have **all** permissions in the `/kn_system//` topic space. You may restrict them to access only from the local host (if yours is a single-machine deployment) or the cluster subnet (using appropriate permissions patterns). Note that both **kninternal** and **kndaemon** directly post events from one machine in a cluster to another machine in the cluster.
- If you configure your permissions in a cluster so that passwords are required, you must provide the **kninternal** user with `notify` permission to `/kn/kn_config/recovery`. No other topic space permissions are required for this user for clustering purposes, though they are required for other purposes.
- The LDAP backend installs **knadmin**, **kndaemon**, **kninternal**, and **~KnDummy**. Of these, only **knadmin** and **kndaemon** can log in (i.e., have passwords). For Active Directory only, the password for **knadmin** is `kn@dm1n94089`.

LDAP User Types

There are three kinds of LDAP user that the various KnowNow components need to become at various times. Each type of user needs specific permissions. For additional information about users, see [“Advanced LDAP Parameters” on page 230](#) and [“The LiveServer’s Default Users and Groups” on page 252](#).

Table 9-1. LDAP user types.

User Type	Associated KnowNow System Administration Parameters	Description
LDAP Schema Administrator	Not configured using the KnowNow System Administration console	This user needs to have the ability to install a new schema and make it available for creating new objects. This user is only used when installing the first LiveServer into your LDAP environment (as described under “Integrating ESS With Your LDAP Environment” on page 161).
LDAP User, Group, and LiveServer object creation and modification	LDAP Administration Password	This is an administrative user who can create objects, change passwords, add users to groups, and add objects into the LiveServer part of the directory information tree (DIT). This user is used by the knauthd backend, which handles all the queries, adds, updates, and deletions provided by the KnowNow System Administration console. This user can also be used by the administrator who installs the user, group, and LiveServer objects described under “Task 3: Load the LiveServer Users and Groups” on page 207 and “Task 4: Load the Permission Patterns” on page 210 .
LDAP User, Group, and LiveServer object lookup	LiveServer Password , LiveServer DN	This user (who might be administrative but probably won’t be if you are running a tight ship) is used by LiveServer itself to look up people and figure out what they can do. This user does not need to be able to modify any information in the LDAP database.

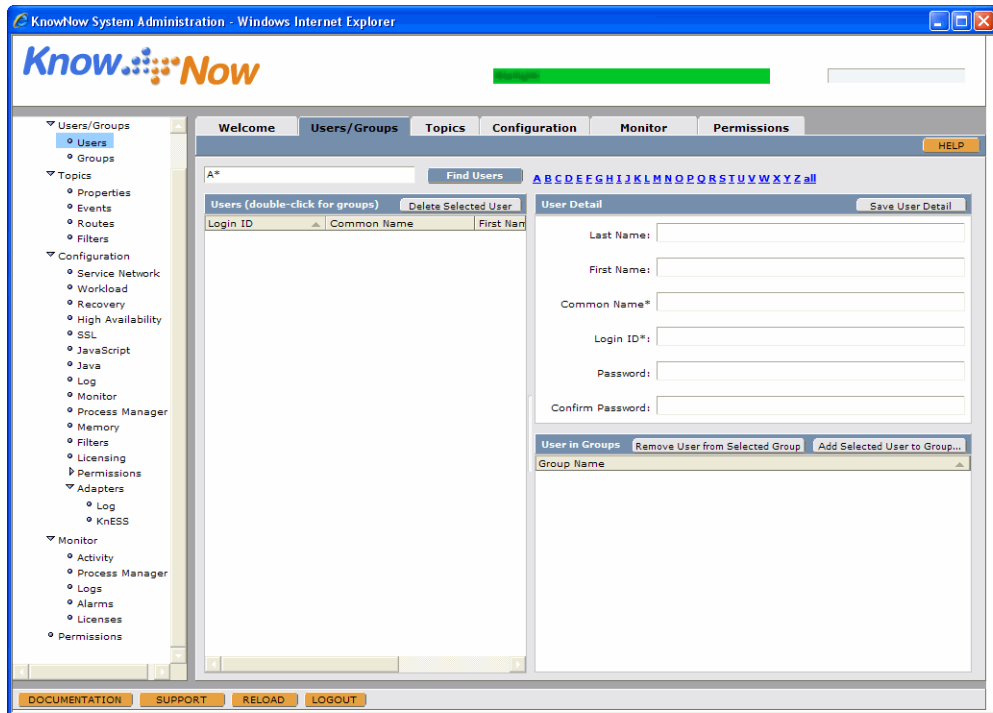
Adding, Finding, and Deleting Users

The following procedure tells how to use the KnowNow System Administration console to add, find, or delete users, and to add or delete users from groups.

To manage users,

1. Access the KnowNow System Administration console.
2. Under **Users/Groups**, choose **Users**. The Users panel is displayed.

Figure 9-2. Managing users.



3. To add a user, enter the user's details in the User Detail area. Fields marked with an asterisk are required. The common name is normally the first and last name. When the information is entered, choose **Save User Detail**. The user is added to the list of users.

A user with no password is a nologin user. Nologin users are useful when making things owned by some central authority. You can change the ownership of topics

using the `kn_owner` header; for more information on `kn_owner`, see the *KnowNow LiveServer Developer's Guide*.

4. To find a user, use one of the following methods.
 - Enter the user's name or a portion of the user's name in the field to the left of the **Find Users** button, then choose **Find Users**.
 - Select one of the letters in the alphabet to the right of the **Find Users** button. The list of all users whose names start with that letter appears in the Users list to the left.
5. To view a user's information, find the user and double-click the user's name in the Users list. The user's information appears in the User Detail area to the right.
6. To delete a user, find the user and double-click the user's name in the Users list. Choose **Delete Selected User**.
7. To see which groups a user is in, find the user and double-click the user's name in the Users list. The group(s) that the user belongs to are shown in the Users in Groups area to the bottom right.
 - To remove the user from a group, choose **Remove User from Selected Group**.
 - To add the selected user to a group, choose **Add Selected User to Group**. The Add User to Group dialog box pops up. To find a group, choose **Find Groups**. LiveServer searches for and displays a list of all groups. Once the groups appear, select a group and choose **Add User to Group**. The user is added to the group and the Add User to Group dialog box closes.

Figure 9-3. Adding a user to a group.



8. To grant or restrict rights for a user, assign that user to an access pattern using the Permissions tab as described in [step 7 on page 242](#).

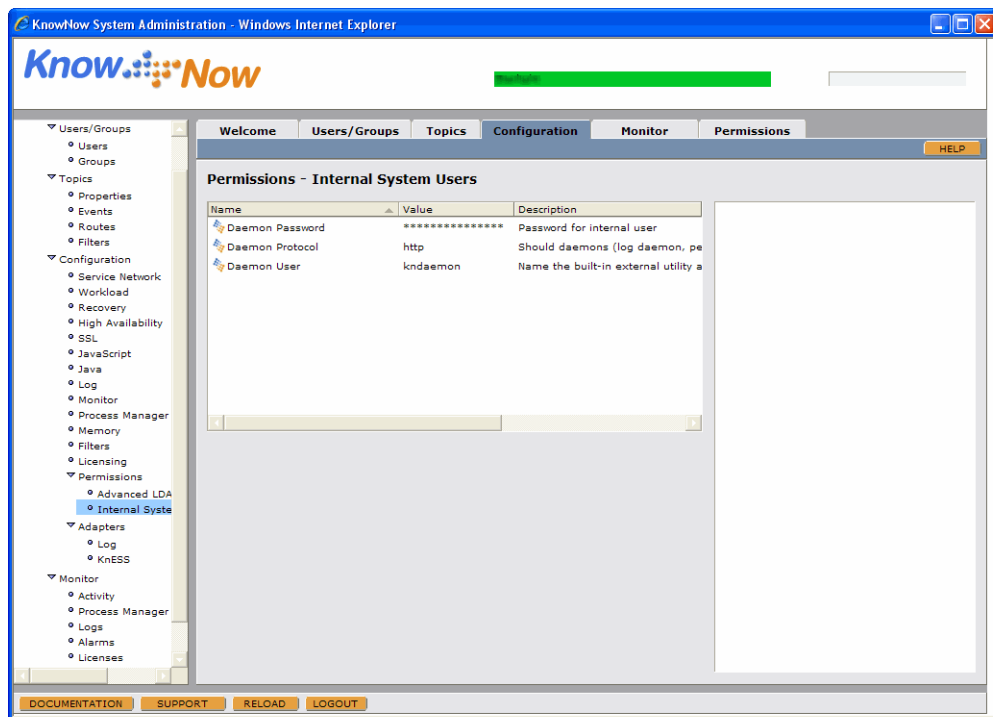
Configuring Internal Users

Once you have selected the Flat File, LDAP, or LDAP test permissions modules, you may wish to configure the password, protocol, and name for the **kndaemon** user. Before changing the **kndaemon**'s user name and password, be aware that changing this user's name and password can have serious consequences. The correct order for changing the user name and password is as follows. (If you are just changing the password, see "Daemon Password" on page 258 for information on doing so.)

To change the kndaemon user's name and password,

1. Access the KnowNow System Administration console.
2. Add a new user (using the Users/Groups tab, as described under "Managing Users" on page 252). This user must also be in the KnowNow_Internal group.
3. Under **Configuration**, select **Permissions**. The Permissions configuration panel is displayed.
4. Select **Internal System Users**.

Figure 9-4. The Internal System Users panel.



5. For each parameter you wish to configure, select that parameter. A field appears to the right for modifying the value of that parameter. Modify the value of the parameter in the field to the right, then choose **Update**. The meanings and values of the parameters are provided under “[Internal System Users Parameters](#)” on [page 258](#); note especially the character restrictions for the **kndaemon** user’s password.
6. Restart LiveServer.
7. Verify that knauthd started up (for example, use the Users/Groups tab to search for a user; if it works, knauthd is running).
8. Once you have verified that knauthd started up, you can delete the old user name using the Users/Groups tab.

Internal System Users Parameters

The internal system parameters identifies the **kndaemon** user, which LiveServer needs in order to operate.



Note: If you are planning to change the name and password for **kndaemon**, make sure that **kndaemon** exists in the regular user section (which is described under “[Managing Users](#)” on [page 252](#)) before you change those parameters here. This user must also be in the KnowNow_Internals group.

For more on this and other default users, see “[The LiveServer’s Default Users and Groups](#)” on [page 252](#).

Daemon Password

This parameter sets the password for the **kndaemon** user. The name of the **kndaemon** user is set by the **Daemon User** parameter. For security purposes, you may wish to change the password. If you are just changing the password and not the user, change the password here first, then change the password in the Users/Groups tab, then restart immediately (within two minutes).

Passwords used throughout the KnowNow system are encoded for security purposes and as such are not human readable.

The kndaemon user’s password cannot contain any of the following characters:

@ # ? % = \$ / \ :

Daemon Protocol

Specifies which protocol (HTTP or HTTPS) the daemons should use.

Daemon User

This parameter sets the user name for the **kndaemon** user. The password for the **kndaemon** user is set by the **Daemon Password** parameter. The **kndaemon** user can only connect to the local host; also, this user is only listened to on restart.

Managing Groups

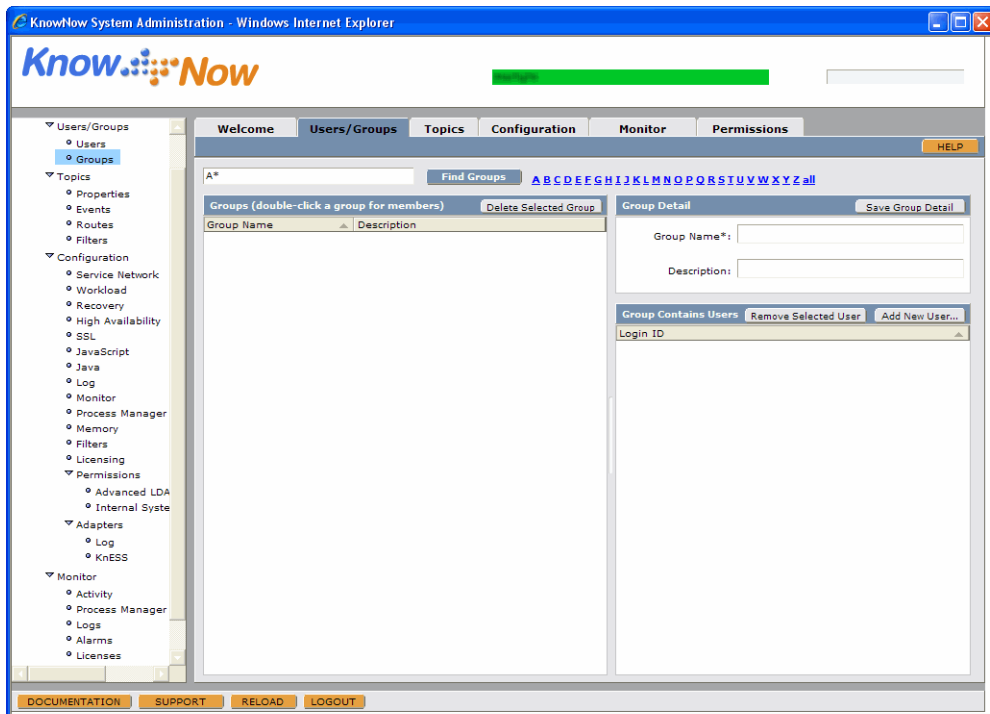
In order to be able to use the Users/Groups tabs, you must have selected one of the following permissions options: **Flat Files**, **LDAP**, or **LDAP Test**.

When in groups mode (accessed by selecting Groups in the navigation tree to the left), the window displays group management capabilities. You can manage all groups here; the only restriction is that LiveServer does not manage groups of groups. In order to be able to use the Users/Groups tabs, you need to have selected one of these permissions options: Flat Files, LDAP, or LDAP Test. For information on the permissions options and how to select and configure them, see [Chapter 6](#), "Selecting a Permissions System."

To manage groups,

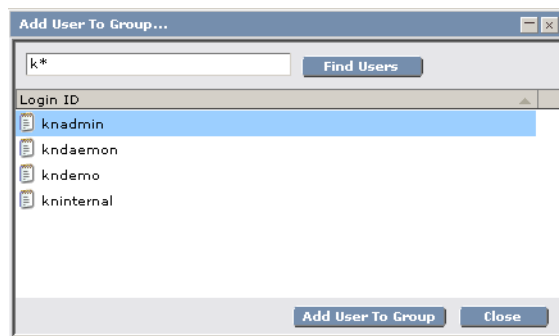
1. Access the KnowNow System Administration console.
2. Under **Users/Groups**, choose **Groups**. The Groups panel is displayed.

Figure 9-5. The Users/Groups tab in groups mode.



3. To add a group, enter the group's details in the Group Detail area. The group name is required. When the information is entered, choose **Save Group Detail**. The group is added to the list of groups.
4. To find a group, use one of the following methods.
 - Enter the group name or a portion of the group name in the field to the left of the **Find Groups** button, then choose **Find Groups**.
 - Select one of the letters in the alphabet to the right of the **Find Groups** button. The list of all groups whose names start with that letter appears in the Groups list to the left.
5. To view a group's information, find the group and double-click the group's name in the Groups list. The group's information appears in the Group Detail area to the right.
6. To delete a group, find the group and double-click the group's name in the Groups list. Choose **Delete Selected Group**.
7. To see which users are in a group, find the group and double-click the group's name in the Groups list. The user(s) that are in the selected group are shown in the Group Contains Users area to the bottom right.
 - To remove a user from a group, select the user and choose **Remove Selected User**.
 - To add a user to a group, select a group, then choose **Add New User**. The Add User to Group dialog box pops up. Use **Find Users** to find a user. Once the user or list of users appears, select a user and choose **Add User to Group**. The user is added to the group and the Add User to Group dialog box closes.

Figure 9-6. Adding a user to a group.



- 8.** To grant or restrict rights for a group, assign that group to an access pattern using the Permissions tab as described in [step 10 on page 243](#).

Chapter 10 Implementing Clusters

Clusters are groups of two or more servers working together, each containing identical data, each sharing the workload as assigned by a load balancer. LiveServer offers a fast, high-end, scalable multicast-based cluster system for LANs and WANs. The module that supports this cluster system is the Multicast module, or Multicast for short.

In this chapter, it is assumed you understand the basics of clusters; therefore, the focus is on how Multicast works with LiveServers, and how to configure Multicast with LiveServers. (For an excellent reference work on clusters, see Greg Pfister's *in Search of Clusters*.)

For information on monitoring the state of the clusters, see [Chapter 11](#), "Performance Monitoring and Process Management."

This chapter describes how to configure LiveServers in a cluster under the following headings:

- "Introducing LiveServer Clusters" on page 264
- "Cluster Creation Overview" on page 268
- "Enabling and Configuring a Multicast Cluster" on page 271
- "Adding Second and Additional Nodes to a Cluster" on page 278
- "Editing Nodes in an Existing Cluster" on page 280
- "Deleting Nodes in an Existing Cluster" on page 281
- "LiveAdapters Clusters" on page 282

Introducing LiveServer Clusters

A cluster is when two or more LiveServers are configured together to provide a machine fail-over mechanism and scalable connections. The LiveServer's Multicast module uses the `knmcluster` module, a fast, scalable, multicast-based clustering module with advanced capabilities, including reliability and ordering options. The capabilities and uses of this module are described starting under [“Enabling and Configuring a Multicast Cluster”](#) on page 271.

The following sections discuss various aspects of LiveServer clusters:

- [“Cluster Module Information”](#)
- [“How Clustering Works”](#) on page 265
- [“Load Balancer Configuration”](#) on page 267
- [“Important Cluster Considerations”](#) on page 267
- [“Cluster Creation Overview”](#) on page 268
- [“Clusters and Security”](#) on page 269

Cluster Module Information

The following facts apply to the LiveServer's clustering capabilities:

- The Multicast module is shipped with LiveServer, but is not enabled by default. You use the **Configuration > High Availability** tab in the KnowNow System Administration console to select and configure it; information on doing so is under [“Cluster Creation Overview”](#) on page 268 and [“Enabling and Configuring a Multicast Cluster”](#) on page 271.
- The Multicast module supports hot node additions and deletions, which means you can get a node up and running, then add it into the cluster while the cluster is still up.
- Changes made to a cluster's configuration using the KnowNow System Administration console are not node-specific; instead, changes made to one node are propagated to every other node in the cluster. If what you want to do with your cluster can't be expressed in a way that is not node-specific, then seek advice from the KnowNow Professional Services Group.
- With external data sources, each node has its own security module and therefore its own security cache. When a node is restarted, the security module is responsible for repopulating its security cache (if any) from the external data source.

- When you create a cluster, you need a software component that makes all the machines in your cluster look like a single machine to your customers. The host name and domain name (configured in the Service Network panel) are the names that customers use to talk with the cluster.
- Journals are persisted in the LiveServer database. This assists recovery of a failed or out-of-service LiveServer or node. The recovery modules on LiveServers that have failed will help bring the auto-restarted LiveServer back into sync with the other LiveServers in the cluster with no loss of data.

How Clustering Works

All LiveServers (nodes) in a cluster mirror each other to look and act like one LiveServer. The Multicast module supports a large number of nodes in a cluster.

When you make configuration changes to one node in a cluster, the changes propagate to all other nodes.

Nodes in a cluster share the state of topics. Every HTTP event that affects topic space that is received at any node in a cluster is automatically replicated to every other member before any processing is done to that event. (In other words, any HTTP event that contains any active `do_method` requests is replicated before processing.) (For more on the `do_method` commands, see the *KnowNow LiveServer Developer's Guide*.)

These events are also not authorized before a cluster begins functioning, though security credentials themselves are replicated. For example, an authorization module looking for the peer IP parameter (in the `kn_authorize` interface) receives the IP address of the client.

In a cluster, every member processes every event from beginning to end. Once the machine gets the event, it processes it as though it came in on a regular port. When writing a filter, you need to install that filter on every member of the cluster and you will need to write the filter so that it will come up with the same answer on each node in the cluster.

When the cluster comes up, the LiveServer cluster modules manage the replication of events.

Additional information on how clusters work is provided in the following sections:

- [“Failover Mechanism” on page 266](#)
- [“Flow Control and System Recovery” on page 266](#)
- [“Scalable Connections” on page 266](#)

Failover Mechanism

Failed machines in a LiveServer cluster can recover on the fly; there is no need to take the cluster down. This is possible in part because of the LiveServer's implementation of data recovery. Configuration of data recovery is discussed under ["Recovery" on page 69](#). The failover mechanism allows for uninterrupted LiveServer requests to continue even if one of the LiveServers in the cluster goes down. If a machine in the cluster fails or is taken out of service, and is then restored, it synchronizes itself with the most current active member of the cluster. Meanwhile, the other nodes in the cluster continue to service LiveServer requests without any interruption to service.

The recovery module contributes to smooth recovery and restart during this process. In fact, a cluster cannot recover without the recovery module; therefore, when you enable clustering, recovery is also enabled.

Flow Control and System Recovery

Network outage recovery will always be bound by memory and queue limitations. During network partition recovery, a minimum of a 15-minute outage and a certain high continued publish rate may cause the Spread process to not recover. In order to help prevent queue overflows and having to restart the Spread process manually, LiveServer performs some flow-control operations on the sending side by limiting messages in transit to about 1,000. Also, the Spread's queue size is set to 10,000. If you experience queue overflows, try restarting the Spread process. Lowering the publish rate during a network partition will also alleviate the problem.

When one node (Node A) stops the Spread process on another node (Node B), although Node B will correctly report the state of the Spread process, Node A will not. This is because, once the process has been stopped, communication has been lost between the two nodes. However, Node A will know that Node B is detached from the cluster.

Scalable Connections

Clusters can be used to increase the total number of connections to LiveServer. With a single machine, the number of concurrent connections to a LiveServer can go up to the operating system's limits. You can exceed the operating system's limits by running several LiveServers in a cluster and distributing the connection load over multiple machines.

LiveServer requests can be directed to any node in the cluster. The cluster module automatically takes each request and mirrors it across all of the other machines in the cluster. Therefore, every node has the same topic hierarchy and receives all of the same events.

Load Balancer Configuration

For clusters, KnowNow recommends using “round robin” algorithms with no slow-start provisions.

The load balancer should be configured so that it is only talking to the ports specified in the **HTTP Port** and **HTTPS Port** parameters. Ideally, the **Cluster port** parameter should point to a separate network interface card (NIC) from the customer service NIC. A good LiveServer cluster configuration uses dual-NIC machines, with one NIC looking at the load balancer and a second LAN/switch for the cluster port.

Important Cluster Considerations

When creating a cluster, there are some things to keep in mind.

- On each client cluster box, remember to set the host name to the externally visible DNS name of the cluster.
- You must keep the cluster parameters in sync with the **Cluster port** parameter and the machine names of the machines in the cluster. Each peer must be able to recognize itself as one of the values in the cluster list.

Cluster Creation Overview

Cluster creation and configuration is managed through the KnowNow System Administration console.



Note: In order for clusters to operate properly, you must configure your clusters behind a load balancer. As part of that configuration, you will need to set the Service Network's **Hostname** parameter as described under [“Configuring the Service Network”](#) on page 58.

The procedure for adding a node to a cluster is different for the first node than for the second and additional nodes. Essentially, to create a cluster, you create and configure the first node, restart LiveServer, then add additional nodes as desired. Under the following headings, this section provides an overview of the procedure for creating the first node in a cluster. Detailed information is provided under [“Enabling and Configuring a Multicast Cluster”](#) on page 271.

- [“Cluster Creation Checklist”](#)
- [“Clusters and Security”](#) on page 269

Cluster Creation Checklist

This checklist is a very high-level overview of the steps needed to create clusters. It is intended primarily for your use when you are already familiar with the steps and just need a checklist for review when creating clusters.



Caution: The checklist in this section provides a valid and supported method for configuring clusters. Please note that the following is *not* a valid and supported method and may cause problems: Having only one node in a cluster configured with a second node in its cluster list, where the second node has no mention of a cluster peer, doesn't have a cluster thread started, and is not replicating events through the cluster. If you accidentally create this invalid configuration, the symptoms are (A) the configured node throws a 404, and (B) an error message, along with the full journal path to the configured node, appears on the second node.

To create a cluster (at a high level),

1. Have at least two machines ready to go for your cluster (i.e., have LiveServer installed on each node and have the machines in a network). For information on installing LiveServer, see [Chapter 2, "Installing KnowNow's LiveServer."](#)
2. Start up the KnowNow System Administration console on one machine and configure it as your first node. For instructions on configuring your first node, see ["Enabling and Configuring a Multicast Cluster" on page 271.](#)
3. Configure security on your cluster. For guidelines, see ["Clusters and Security" on page 269.](#)
4. Restart LiveServer on that node.
5. Add more nodes as described under ["Adding Second and Additional Nodes to a Cluster" on page 278.](#)
6. Test your cluster; some quick ways for testing the cluster are described under ["Adding Second and Additional Nodes to a Cluster" on page 278.](#)

Clusters and Security

LiveServer requires a user name and password for every communication in a cluster. By default, the user name and password are the user name **kndaemon** and the password *kndaemon*.

You can change the **kndaemon** user name and password as described under “[Managing Users](#)” on page 252.

Passwords used throughout the KnowNow system are encoded for security purposes and as such are not human readable.

Enabling and Configuring a Multicast Cluster

The multicast-based clustering module, `knmcluster`, is scalable. In addition, it uses the Spread toolkit, so that each LiveServer instance has a Spread daemon that it manages; cluster communications take place through that daemon. Information can be found on configuring and maintaining a Multicast cluster under the following headings:

- [“Initial Multicast Configuration: Creating the First Node”](#)
- [“Multicast Parameters” on page 274](#)
- [“Adding Second and Additional Nodes to a Cluster” on page 278](#)
- [“Editing Nodes in an Existing Cluster” on page 280](#)
- [“Deleting Nodes in an Existing Cluster” on page 281](#)
- [“LiveAdapters Clusters” on page 282](#)

Initial Multicast Configuration: Creating the First Node

The following instructions describe how to configure the **first** node in a LAN or WAN multicast cluster.



Caution: Do not add a second node during this procedure. For instructions on adding a second and additional nodes to a new or existing Multicast cluster, see [“Adding Second and Additional Nodes to a Cluster” on page 278](#).

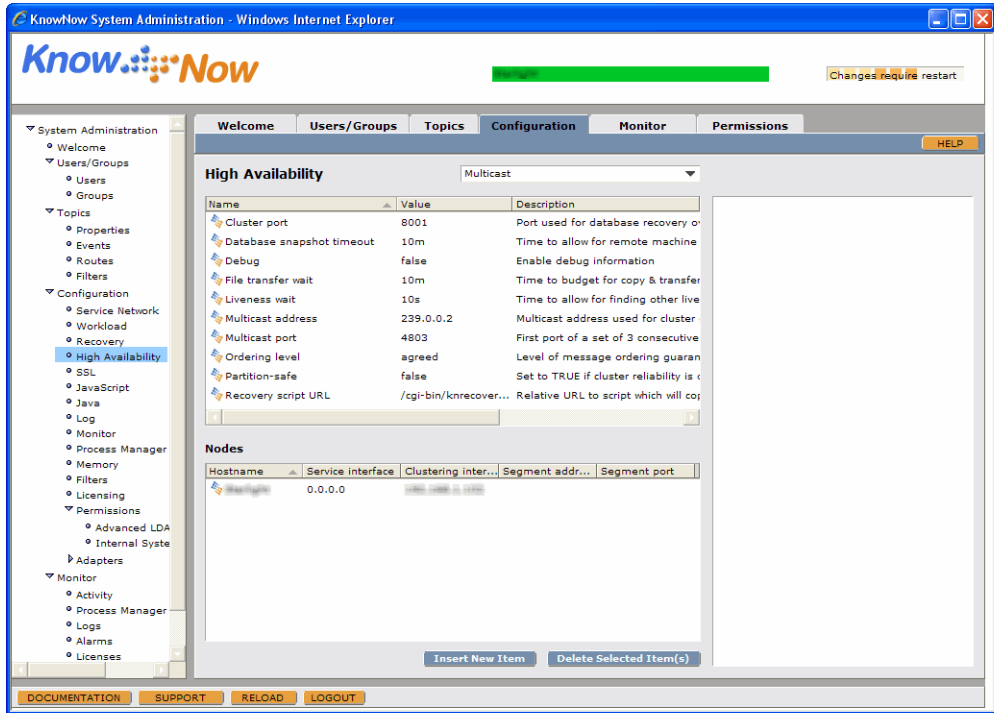
For instructions on editing or deleting existing nodes, see [“Editing Nodes in an Existing Cluster” on page 280](#) and [“Deleting Nodes in an Existing Cluster” on page 281](#).

To enable and configure Multicast for the first node in a new cluster,

1. Access the KnowNow System Administration console.
2. Under **Configuration**, select **High Availability**.
3. In the drop-down box to the right of the words **High Availability**, choose **Multicast**. In selecting this option, you enable Multicast clustering. Also, when you select this option, the parameters for Multicast appear. The upper panel lists cluster-wide parameters; the lower panel lists each node in the cluster and shows

that node's information. Your node's information should be automatically detected and displayed in the lower panel.

Figure 10-1. Configuring Multicast.



4. To edit the node-specific parameters (those in the lower panel), click on the node in the lower panel. When you do so, fields for editing that node appear to the right.

a. The **Hostname** specifies the public name of a node in the cluster. This field must have a value. Node names cannot exceed 100 characters in length.

Important: You also need to set the Service Network **Hostname** parameter in as described under “[Clustered Deployments Require This Parameter](#)” on [page 62](#). Setting this parameter takes place in [step 7 on page 273](#).

b. The **Service Interface** specifies an IP address or interface for the cluster to use. LiveServer resolves the host name if possible. To provide client services (HTTP, HTTPS), this parameter sets the IP address that LiveServer should bind to. This variable is useful if a machine has more than one network connection. The default value is 0.0.0.0, which instructs LiveServer to use every interface card (or bind to all interfaces on this machine). Use this default value if you don't want

LiveServer to use the host name. The service interface value can be different for each node in the cluster.

- c. The **Clustering Interface** parameter specifies the IP/interface address to use for clustering messages. This is useful if the node has multiple network addresses. In this case, the Service Interface parameter can use one of the node's IP addresses and the **Clustering Interface** parameter can use the node's other IP address.
- d. The **Segment Address** parameter specifies the multicast address for cluster group communication. The Multicast cluster is identified by this address. (The nodes are talking to the cluster through this address.) This value must be the same on all cluster nodes in the same network. The value in this field defaults to the value in the **Multicast address** parameter. If you enter a value here, it is used rather than the value in the **Multicast address** parameter. For more information on this parameter, see ["Multicast address" on page 275](#).

Also, if you are configuring a WAN Multicast cluster, the address must be the same for all nodes in a given network, but must be unique from the address(es) used for nodes in another network. For example, if you have two networks, A and B, all nodes in A must share one address, and all nodes in B must share one address that is different from the address used for the nodes in A.

- e. The **Segment Port** parameter specifies the port to use for multicast cluster group communication. The value in this parameter must be the same on all cluster nodes in the same network. The value in this field defaults to the value in the **Multicast port** parameter. If you enter a value here, it is used rather than the value in the **Multicast port** parameter. For more information, see ["Multicast port" on page 276](#).
5. When you have entered the information in these fields, choose **Update**.
 6. To set or change any of the cluster-wide parameters (which are listed in the upper part of the High Availability panel), select the desired parameter. A field appears to the right listing the current value for that parameter. These parameters all ship with default values, so you may not need to make any changes here. If you do wish to change anything, change the value in that field and choose **Update**. The parameters are defined under ["Multicast Parameters" on page 274](#).
 7. Access the Service Network panel and set the **Hostname** parameter. For clusters, this parameter specifies the name of the cluster or of the cluster's load balancer. For more information on setting Service Network parameters, see ["Configuring the Service Network" on page 58](#). Also read ["Clustered Deployments Require This Parameter" on page 62](#).

8. Optional: If not already done, configure recovery as described under [“Enabling and Configuring Recovery” on page 69](#). (Recovery is automatically enabled when you choose one of the High Availability options.) You can perform this task at another time if you wish.
9. Configure security for the cluster. See [“Clusters and Security” on page 269](#) for some guidelines.
10. After you have made the necessary changes (including to the security files), you must restart the node you have just made changes to. To do so, choose **Restart Cluster** on the Welcome tab.
11. After restarting, for each of the other cluster nodes, follow the instructions under [“Adding Second and Additional Nodes to a Cluster” on page 278](#). You can also use those instructions for hot node addition (that is, to add a completely new LiveServer node into an already running cluster). You don’t need to bring down any previously running LiveServers in the cluster.
12. To configure SSL for your cluster, see [“Configuring the Service Network” on page 58](#) and [“Configuring SSL” on page 75](#).

Multicast Parameters

This section lists and describes the parameters for configuring Multicast. Note that several of the Multicast parameters are for Spread configuration and should not be changed.

For a discussion of the units of measure used in some parameters, see [“Specifying Units in Parameters” on page 53](#).

Cluster port

Specifies the port for intra-cluster machine communications and database recovery; this is the value that all members in a group should be talking to. It must be identical across all nodes. The default value is **8001**.

Database snapshot timeout

Specifies the amount of time to allow for the remote node to take a snapshot during recovery. The default value is **10m** (ten minutes). For information on enabling and configuring recovery, see [“Recovery” on page 69](#).

Debug

Turns on debugging statements in the log. Enable this parameter only when asked to do so by KnowNow Support. The default value is **false**.

File transfer wait

The value in this parameter is used during recovery. It specifies the length of time for a node to wait for the transfer of a database file from another live node. For information on enabling and configuring recovery, see [“Recovery” on page 69](#).

The default value is **10m** (ten minutes).

Liveness wait

The value in this parameter is used during recovery. The default value is **10s** (ten seconds).

This parameter tells LiveServer how long to wait for a response from another machine in the cluster before deciding that the failed/recovering machine (the one that is waiting) is the first one to recover. If another machine is already up and responds within the amount of time specified in this parameter, the failed/recovering machine uses the data from that machine. If the recovering machine cannot find another live machine within the amount of time specified by this parameter, it assumes that it is the first machine to recover and uses its own database. As soon as a failed machine comes back up, it sends an “I’m alive” message to the other machines in the cluster. For information on enabling and configuring recovery, see [“Recovery” on page 69](#).

If you are creating a WAN cluster, you may need to increase the value in this parameter, depending on the speed of the routers in between different networks and on how long it takes for the nodes on the different networks to see each other. If the router speed is too slow and this parameter is too low, nodes might not correctly recover from their cluster peers.

Multicast address

This Spread configuration parameter specifies the multicast address for cluster group communication. The Multicast cluster is identified by this address. (The nodes are talking to the cluster through this address.) This must be the same on all nodes in a cluster.

Also, this address must be unique on a network for a particular cluster. If you plan to run a different LiveServer cluster in the same network, you will need to make sure that the multicast address for each cluster is different from the multicast address for any other LiveServer cluster on that subnet. We recommend that you use the multicast range 239.0.0/8 (239.0.0.0 to 239.255.255.255).

If you are configuring a WAN Multicast cluster, the address must be the same for all nodes in a given network, but must be unique from the address used for nodes in another network.

The default value is **239.0.0.2**.

Multicast port

This Spread configuration parameter specifies the port to use for multicast cluster group communication. The value in this parameter must be the same on all nodes in the same cluster in the same network. The default value is **4803**.

Ordering level

Sets the level of message ordering guarantee. Multicast has four levels of reliability, which you select using this parameter. Each higher level provides what all lower levels provide, and adds features as well. The choices are **reliable** (the lowest level), **causal**, **agreed**, and **safe**. The default value is **agreed**.

- At the **reliable** (unordered) level, there are no ordering guarantees, but all messages are delivered reliably.
- At the **causal** level, all messages are reliable and “are delivered in an order consistent with Lamport’s definition of ‘Causal’ order.” (Quote taken from http://www.spread.org/docs/guide/users_guide.pdf, page 5. For more on Lamport, see both <http://www.podc.org/influential/2000.html> and <http://portal.acm.org/citation.cfm?doid=359545.359563>.)
- At the **agreed** (total) level, messages are reliable and causal, and all subscribers receive all events in the same order.
- At the **safe** level, messages are reliable and safe ordered, and are also queued in case of node failure across a partition. This mode is also guaranteed to be partition safe. If there is a partition between the cluster nodes, then events that are posted during the partition time are queued. Once the partition is recovered from, those events are posted to the other cluster nodes.

Partition-safe

When this parameter is enabled (set to **true**), events posted during the partition time are queued. This means that if a cluster recovers from a partition, its queue will continue to grow, and it could run out of disk space. It is recommended that the administrator fix the partition as soon as possible or bring down the node that has been detached.

Set this value to true if cluster reliability is desired across network partitions. When this parameter is set to **true**, the value of [Ordering level](#) is automatically set to **safe**.

The default value is **false**.

Recovery script URL

A relative URL used during recovery. The URL identifies a CGI script on a live LiveServer to be used during system recovery; that script copies the database from the live machine to the failed/recovery machine as described under [“Failover Mechanism” on page 266](#). For information on enabling and configuring recovery, see [“Recovery” on page 69](#).

The default value is **/cgi-bin/knrecover.cgi?node=%s**.

Adding Second and Additional Nodes to a Cluster

Once you have configured a cluster with a first node as described under “[Enabling and Configuring a Multicast Cluster](#)” on page 271, you can complete the cluster by adding nodes to your new cluster as described in this section.



Caution: Be sure to first create your first node and restart LiveServer before performing this procedure.

Also, for existing clusters (i.e., once a cluster is up and running with all its nodes), you can use these instructions to add a new node to that existing cluster without having to bring the other nodes down.



Note: When you configure a new node with a host name, and that host is up and running, then the LiveServer where you are doing the configuration will push the configuration for that node to all other nodes, and will then restart the newly configured node.

To add nodes to a cluster, for each new cluster node,

1. Start LiveServer on the new node in the default configuration.
2. On the new node, in the KnowNow System Administration console, select the clustering module you wish to use. You don't need to change any other parameters, but do ensure that the settings for the **Multicast address** and **Multicast port** parameters are consistent with the settings of those same parameters for the first node you created (using the instructions under “[Enabling and Configuring a Multicast Cluster](#)” on page 271; we will call that first node Node A).
3. Restart LiveServer on the new node using the **Restart Cluster** button on the Welcome page.
4. Using the first node (Node A), in the lower panel, choose **Insert New Item**. Depending on which type of cluster you are adding nodes to, a varying number of fields, including the **Hostname**, **Service Interface**, and **Clustering Interface** fields, appear to the right. For information on these fields, see [step 4 on page 272](#).
5. Enter the proper values for the new node in these fields.
6. Choose **Insert** to add the node to the cluster. If the node is up, the node is added to the list of nodes in the lower part of this panel. Also, as soon as you choose **Insert**,

the first node of the cluster pushes all its configuration information to the new node, instructing it to restart so that the changes can take effect.



Note: After adding, for example, Node B to Node A's list, we recommend that you wait until Node B is completely assimilated into the cluster (as indicated by a green Node B status on Node A's Welcome screen) before following the above instructions to add Node C, Node D, and so on.

7. To test your cluster, use the KnowNow System Administration console to create a topic under `/kn_system/` on one node. Check the other node to see if that new topic appears. Next, send a test event to that new topic and check to see if the event is received on the other node as well. If a node does not join the cluster, or if the cluster is inconsistent, insure that the `knsettings.conf` files are correct on both nodes.

Editing Nodes in an Existing Cluster

You can edit nodes in a cluster as described in this section.

To edit existing nodes,

1. Access the KnowNow System Administration console.
2. Under **Configuration**, select **High Availability**. Your chosen cluster module and its parameters are displayed.
3. To edit the cluster-wide parameters (those in the upper panel), click on the desired parameter and update it as needed. Information for the cluster-wide parameters is provided under [“Multicast Parameters” on page 274](#). When you make these changes on any node, the new settings are immediately pushed to the other nodes.
4. To edit the node-specific parameters (those in the lower panel), click on the node in the lower panel. When you do so, a number of fields for that node, including the **Hostname**, **Service Interface**, and **Clustering Interface**, appear to the right. For information on these fields, see [step 4 on page 272](#).
5. When you have entered the information in these fields, choose **Update**.
6. Restart LiveServer using the **Restart Cluster** button on the Welcome page. The changes take effect when you restart LiveServer.

Deleting Nodes in an Existing Cluster

You can delete nodes in a cluster using the instructions in this section.

To delete nodes,

1. Access the KnowNow System Administration console.
2. Under **Configuration**, select **High Availability**.
3. For each node you wish to delete, select the node in the Nodes list (in the lower panel) and choose **Delete Selected Item(s)**.
4. Restart LiveServer using the **Restart Cluster** button on the Welcome page. The changes take effect when you restart LiveServer.

LiveAdapters Clusters

LiveAdapters allow for deployment in a clustered environment such that two or more LiveAdapters can work in unison to provide all LiveAdapter services. This setup is called a LiveAdapters cluster. A LiveAdapters cluster is built over a LiveServer cluster. Such a cluster allows business methods to be available in case of the failover of any node in the cluster.

This type of cluster provides basic failover functionality. We recommend that you use it with a load-balancing mechanism such as DNS round robin to achieve better performance. While reading the following instructions, bear in mind that, when using shared file systems, you will need to address any issues of latency, performance, and security.

Instructions for creating LiveAdapters clusters are provided under the following headings:

- [“Creating LiveAdapters Clusters on Windows 2000 or Older and Linux”](#)
- [“Creating LiveAdapters Clusters on Windows 2003” on page 284](#)

Creating LiveAdapters Clusters on Windows 2000 or Older and Linux

The following instructions are for Windows 2000 and older, and for Linux. If you are creating a LiveAdapters cluster on Windows 2003, Windows XP, or Windows Vista, see [“Creating LiveAdapters Clusters on Windows 2003” on page 284](#).

To set up a LiveAdapters cluster,

1. Install LiveServer and LiveAdapters on each node in the cluster. For information on installing LiveServer, see [Chapter 2, “Installing KnowNow’s LiveServer.”](#) For information on installing LiveAdapters, see the *KnowNow LiveAdapters User’s Guide*.
2. Set up a DNS round robin or similar load-balancing scheme.
3. Create a shared directory for the shared database (/knadapters_db/). Creating this shared directory will differ for each platform. For details on how to set up a shared directory, see [“Windows 2000 and Older Shared Directory Notes” on page 283](#) or [“Linux Shared Directory Notes” on page 284](#).
4. Map or mount the shared directory on each of the nodes so that all nodes use a common path. LiveAdapters will be accessing the shared directory using this path. Again, please refer to [“Windows 2000 and Older Shared Directory Notes” on](#)

page 283 or “Linux Shared Directory Notes” on page 284 for information on how to perform this step.

Note: The mapped/mounted path of the shared directory should be the same on each node. (For example, **F:** or **/mnt/liveadapters/shreddir**.)

5. Set up a LiveServer cluster as described under “Enabling and Configuring a Multicast Cluster” on page 271.

Note: In a LiveAdapters cluster, the LiveServer alias should point to the LiveServer cluster name rather than any individual node name. For example, if you have a cluster of two nodes, where the node names are

```
node1.example.com  
node2.example.com
```

and the cluster name is cluster.example.com, then while creating the LiveServer alias, the host portion of the LiveServer URL should be cluster.example.com rather than either of the node names. This will ensure correct data publication even if one of the nodes is down.

6. Restart the cluster.
7. Now the shared directory must be made known to LiveAdapters. To do so, set the **Shared Filesystem Path** parameter as described under “Configuring LiveAdapters” on page 104. This is the final task you need to perform. LiveAdapters will restart themselves automatically.

Windows 2000 and Older Shared Directory Notes

All the nodes should be in the same workgroup/domain.

To create a mapped, shared directory,

1. Create a directory on one node.
2. Share it such that all nodes in the cluster have read-write permissions on the directory.
3. Map the directory on each node. This can be done by right-clicking on “My Computer” and selecting “Map Network Drive.”
4. Select the shared directory you created in [step 3 on page 282](#) using a drive letter that is common on all nodes of the cluster.

Linux Shared Directory Notes

These notes contain information on creating a mounted NFS directory. NFS is just one way to set up a shared directory. You are free to use any other network file system, such as SMB or SFS, to set up the shared directory for your LiveAdapters cluster.

To create a mounted NFS directory, you need to set up an NFS server on the system that will host the shared directory. The cluster nodes would be NFS clients of the shared directory.

- For information on setting up an NFS server, see <http://nfs.sourceforge.net/nfs-howto/ar01s03.html>.
- For information on mounting the shared directory on the nodes, see <http://nfs.sourceforge.net/nfs-howto/ar01s04.html>.

Ensure that the **knserver** user on each node has read/write access to the shared directory.

Creating LiveAdapters Clusters on Windows 2003

The following instructions are for creating a LiveAdapters cluster on Windows 2003. If you are creating a LiveAdapters cluster on Windows 2000 and older, or on Linux, see “Creating LiveAdapters Clusters on Windows 2000 or Older and Linux” on page 282.

To set up a LiveAdapters cluster for Windows 2003,

1. Install LiveServer and LiveAdapters on each node that will be in the cluster. For information on installing LiveServer, see [Chapter 2, “Installing KnowNow’s LiveServer.”](#) For information on installing LiveAdapters, see the *KnowNow LiveAdapters User’s Guide*.
2. Set up a DNS round robin or similar load-balancing scheme.
3. Create a shared directory for the shared database (/knadapters_db/):
 - a. Create a directory on one node.
 - b. Share the directory by setting both file permission and sharing permissions so that the following entities have read-write access to the directory:
 - i. SYSTEM
 - ii. all nodes of the cluster
 - iii. Administrator Group

4. Set up a LiveServer cluster as described under “Enabling and Configuring a Multicast Cluster” on page 271

Note: In a LiveAdapters cluster, the LiveServer alias should point to the LiveServer cluster name rather than any individual node name. For example, if you have a cluster of two nodes, where the node names are

```
node1.example.com  
node2.example.com
```

and LiveAdapters clusters and the cluster name is `cluster.example.com`, then when creating the LiveServer alias, the host portion of the LiveServer URL should be `cluster.example.com` rather than either of the node names. This will ensure correct data publication even if one of the nodes is down.

5. Restart the cluster.
6. Now the shared directory must be made known to LiveAdapters. To do so, set the the **Shared Filesystem Path** parameter (as described under [step 3 on page 105](#)). This parameter specifies the windows UNC path for a shared directory for a LiveAdapters cluster. Provide the path without any trailing slashes. For example, set this parameter as `//machine_name/share_folder`. **Important: Even though this is Windows, use forward slashes in the path.**

Chapter 11 Performance Monitoring and Process Management

Using the Configuration and Monitor tabs in the KnowNow System Administration console, you can configure and monitor the performance of various aspects of your LiveServer by viewing statistics, log files, and monitoring alarms. You can also manage processes using the LiveServer's Process Manager.

The tasks for configuring and monitoring activity, the Process Manager, logs, alarms, and memory are described under the following topics:

- ["Performance Measurement and Tuning" on page 288](#)
- ["Monitoring Activity" on page 290](#)
- ["Configuring and Using the Process Manager" on page 293](#)
- ["Configuring and Viewing Logs" on page 301](#)
- ["Configuring and Monitoring Alarms" on page 308](#)
- ["Configuring Memory" on page 313](#)

Performance Measurement and Tuning

LiveServer always measures certain performance timings. Those timings are recorded in *statistics.xml*. The KnowNow System Administration console provides a number of parameters that support performance measurement and tuning. Some are enabled by default; others are not. You can enable more measurements, particularly of the reaper. Especially, see the **Configuration > Memory > Keep detailed XML reaper statistics** parameter. This identifies specific operations that should be measured.

To obtain various kinds of data, the actions and parameters described in this section can be of use.

You can deliberately cause very long timings to fault the server and get a core dump using the **Configuration > Log** parameters shown in [Table 11-1](#).

Table 11-1. Performance measurement by crash landing,

Parameter Name	Description	Default Value	Requires restart?
Crash if a performance measurement is out of range	Crash (with dump) when any performance measurement is out of range (as set by the "Performance range..." parameters).	false	No
Performance range for fast operations	Event operations should complete in this time (or assert).	60s	Yes
Performance range for background operations	Database and reaper operations should complete in this time (or assert).	20m	No

You can also have the (internal, no network) response time measured for each operation that starts with a topic space command (for example, `do_method`), with a log entry being generated for each command that exceeds a specified length of time. The parameter for this is **Maximum response time allowed for any command processing**. Almost everything in ESS that isn't an `.adp` is one of these; in particular, `new_user` (login) is one of these. Note that many of these actions ("services") require many event operations. For example, a typical `new_user` may require creation of hundreds of events, representing subscriptions, personal categories, preferences, etc., and routing of dozens more.

While that parameter focuses on response time, you can also try to measure the space you're using, at least in some ways. LiveServer has monitors for both heap and stack, as follows:

- Most of LiveServer's important data structures (events, topics, strings) are reference-counted. All of LiveServer's memory allocations are counted. You can enable a simple verification that all reference-counted objects were freed at shutdown by setting the **Enable Leak Check** parameter.
- When the **Enable Stack Check** parameter is enabled, LiveServer checks whether the stack size is close to being exceeded every time it starts a TCL function call (whether in a filter or .adp). It also checks that stack size when it's been given a new topic space operation from TCL. It will log if the current stack frame is within 5K of the top of the stack. This check is very cheap in terms of resources, so we recommend that you leave it always on. If you're trying to squeeze out the last microsecond, you can disable it.
- The reaper can log how much process memory the overall LiveServer process is using. If you enable the **Log memory statistics?** parameter, it will record this, along with a set of statistics about what got reaped, every time it runs.

Table 11-2. Memory-monitoring parameters,

Parameter Name	Description	Default Value	Requires restart?	Notes
Enable Leak Check	Memory leak-check level.	0	Yes	
Enable Stack Check	Check the stack depth before calling any TCL functions and when calling new topic space commands from filters. Log if it's getting close to the configured stack size (specified on the Workload tab).	true	No	Inexpensive, resource-wise. Don't disable.
Log memory statistics?	Logs available memory on every reaper run.	false	No	Remember to view the log.

Monitoring Activity

The LiveServer activity information comes out of /kn_statistics. Discarding the LiveServer recovery files and restarting LiveServer resets these values to zero.

Under normal conditions, statistics are generated every ten seconds as events with 30-second expirations. Statistic generation may be delayed by the reaper, the process that runs at intervals to delete expired events. Configuration of the various things that feed into the monitor is covered earlier in this chapter.

To view system activity,

1. Access the KnowNow System Administration console.
2. Under **Monitor**, select **Activity**. The Activity panel displays.
3. Under **Activity - Specific Node**, choose a node from the drop-down box. The node's activity displays. [Table 11-3](#) lists and describes the activity statistics.

Figure 11-1. The Activity panel showing a node's activity.

The screenshot shows the KnowNow System Administration console in a Windows Internet Explorer browser. The 'Monitor' tab is selected, displaying two tables of activity statistics. The top table, 'Activity - All Nodes', shows statistics for all nodes. The bottom table, 'Activity - Specific Node', shows statistics for the selected node 'Starlight'.

Statistic Name	Current	Max	Category
Active Connections	5	8	connection
Events Delivered	0	0	event
Events Published	3734	3734	event
Events Routed	25139	25139	event
Journal Topics	5	12	topic
KNLanguage Subscribers	3	3	connection
LiveBrowser Subscribers	1	1	connection
LiveServer Version	dev_build	dev_build	info
Name Topics	915	932	topic
Other Subscribers	1	1	connection

Statistic Name	Current	Max	Category
Active Connections	5	9	connection
Events Delivered	0	0	event
Events Published	3734	3734	event
Events Routed	25139	25139	event
Journal Topics	5	12	topic
KNLanguage Subscribers	3	3	connection
LiveBrowser Subscribers	1	2	connection
LiveServer Version	dev_build	dev_build	info
Name Topics	915	932	topic
Other Subscribers	1	1	connection

The columns contain the following information:

- **Statistic Name** contains the name of the statistic. For information on what each statistic indicates, see [Table 11-3](#).
- **Current** indicates the current value for that particular statistic.
- **Max** indicates the largest value ever seen for that particular statistic.
- **Category** indicates the category in which the statistic falls. The categories are Connection, Event, Info, and Topic. Each category contains one or more statistics (as shown in [Table 11-3](#)).

[Table 11-3](#) organizes the statistics by statistic name and describes what each statistic means.

Table 11-3. LiveServer statistics in the Activity panel.

Category	Statistic Name	Meaning
Connection	Active Connections	Current number of TCP/IP connections by subscribers or publishers. Includes everyone who posts (i.e., non-tunnel connections)
Event	Events Delivered	Events sent down a tunnel. Each event gets counted each time.
Event	Events Published	Any event originated by this LiveServer. Each event gets counted once no matter what.
Event	Events Routed	Every time an event is routed, it gets counted.
Topic	Journal Topics	Current number of sessions with subscribers.
Connection	KNLanguage Subscribers	Current number of KNLanguage subscribers.
Connection	LiveBrowser Subscribers	Current number of subscribers using LiveBrowser.
Info	LiveServer Version	The version of the LiveServer that you are running.
Topic	Name Topics	Current number of normal (named) topics.
Connection	Other Subscribers	Anyone who connects using a Connector that didn't tell LiveServer what it is (for example, non-LiveBrowser browsers).

Table 11-3. LiveServer statistics in the Activity panel. *(continued)*

Category	Statistic Name	Meaning
Topic	Route Topics	Any topics that contain the /kn_routes topic; this measures the number of notify or journal topics that have any subscribers.
Topic	Subtopic Topics	Current number of topics that have any subtopics.
Connection	Total Subscribers	Current total number of active tunnels. Includes people who are only connected to post.
Topic	Total Topics	Current number of topics of any kind in LiveServer.

Configuring and Using the Process Manager

The LiveServer's Process Manager manages a number of vital LiveServer processes. The standard LiveServer processes it manages are:

- the license manager, which controls how you manage your licenses (you can choose to enable or disable this based on whether you want LiveServer to manage your license manager and daemon)¹
- the log daemon, knlogd
- the file manager, which is how the KnowNow System Administration manages changes to configuration and displays data onscreen

Depending on your configuration, the Process Manager may also manage other LiveServer processes. Processes are managed on a per-node basis.

In addition to managing LiveServer processes, the Process Manager can also manage your processes—essentially, any executable built for your operating system. You can add your processes to the Process Manager, then manage those same processes, using the information in the following sections:

- [“Configuring Processes”](#)
- [“Monitoring Processes” on page 298](#)

1. Still in product?

Configuring Processes

The first step in managing your own processes is to add those processes to the Process Manager. This task is performed as a configuration task.



Note: When LiveServer shuts down, it will clean up all the processes you enter using the LiveServer's Process Manager configuration panel. Therefore, if you wish a process to keep running even if LiveServer is shut down, you may not wish to have the LiveServer's Process Manager manage that process.

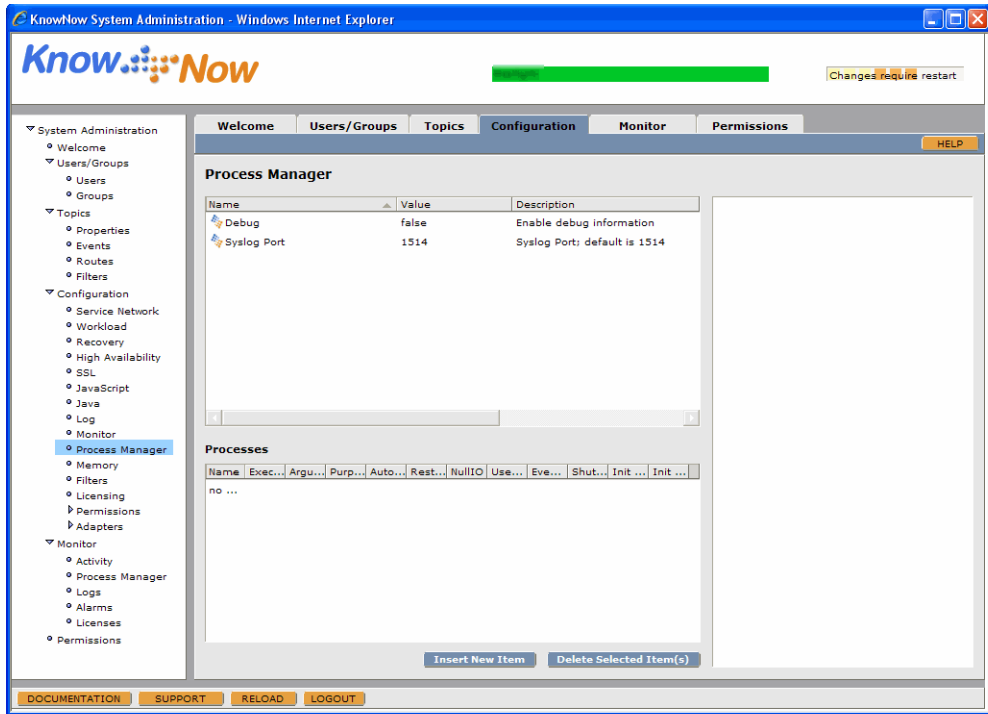
Also note that if you are ending LiveServer from the Windows Task Manager, be sure to choose **End Process Tree** (rather than **End Process**).

To configure the Process Manager,

1. Access the KnowNow System Administration console.

2. Under **Configuration**, select **Process Manager**. The Process Manager panel is displayed. This panel is where you set your options for the Process Manager and where you manage processes.

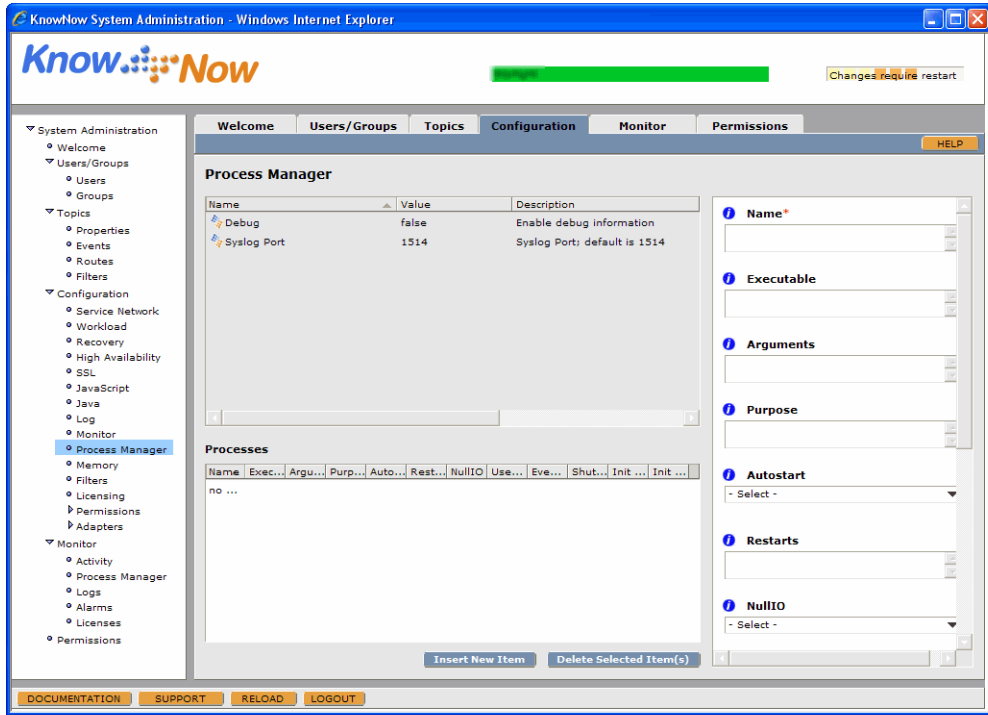
Figure 11-2. The Process Manager panel.



3. In the upper panel, there are two parameters.
 - **Debug.** This parameter enables debug logging in the Process Manager module. Enable this parameter only when asked to do so by KnowNow Support. The default value is **false**.
 - **Syslog Port.** This parameter sets the port for the Syslog process. The default value is **1514**.

- In the lower panel is a list of your processes if you have entered any. To add a process, choose **Insert New Item**. A number of fields are displayed to the right.

Figure 11-3. Adding a new process.



The fields in the right-hand panel are as follows:

- Name.** A name is required for this process. The name can be any string you want, though note that LiveServer will replace spaces with an underbar (so, for example, Your Process Name will become Your_Process_Name). The name you enter will be used by the Process Manager; for example, when viewing processes as described under [“Monitoring Processes” on page 298](#).
- Executable.** This is a full path to the executable, and is also required. Spaces are okay in the path. If you want to pass system environments with the executable, you can. Contact KnowNow Support for information on how to do so.
- Arguments.** These are arguments you want to have passed to the process. Enter a comma-separated list of parameters that the process will understand and that it needs to start up. Do not use spaces; the commas will be replaced with spaces.

- **Purpose.** If you wish, enter a description of the process. It will show up in the monitoring screen as described under [“Monitoring Processes” on page 298](#).
- **Autostart.** Set to **true** (the default) to automatically start the process when LiveServer starts.
- **Restarts.** The value in this field is used if the process stops or exits (though not if you have manually stopped the process). The default value is 10.
 - Set to a positive number for a finite number of restart attempts in one minute. If LiveServer attempts to restart the process, it will attempt to restart for that many times in one minute. For example, if you set this field to **5**, LiveServer will attempt to restart the process five times.
 - Set to any negative number for an infinite number of restart attempts.
 - Set to 0 to disable restart attempts.
- **NullIO.** Set to **true** if stdout redirection to NUL is desired. Otherwise, set to **false**.
- **Uses Server log.** Set to **true** if you wish the process to use the LiveServer’s *server.log* file. Otherwise, set to **false**.
- **Event-driven Shutdown.** You may not want the managed process to be terminated without warning by the Process Manager at server shutdown or when the process is stopped by pushing the **Stop Process** or **Restart Process** buttons in the Monitor > Process Manager screen. In that case, set this option to **true**. When this option is set to **true**, the process is informed when it needs to stop. That allows the process to perform any cleanups that it needs to do before exiting. In order to be informed of the stop, the process needs to subscribe to the management topic, which it can get from its environment variable `KN_PROC_MGMT`. On getting an event whose `kn_id = “process name,”` where the header `kn_action` is set to “stop” or “restart,” the process needs to clean up and exit. The default value is false
- **Shutdown Timeout.** When **Event-driven shutdown** is set to **true**, this parameter specifies the maximum number of seconds that the Process Manager will give the process to exit before the Process Manager terminates the process. The default value is 60 seconds.
- **Init process.** If you wish to have an executable run before the process is started (for example, an initialization process), enter a full path to that executable here. This field uses the same rules as the **Executable** field.
- **Init arguments.** In this field, enter arguments for the process you entered in the **Init process** field. This field uses the same rules as the **Arguments** field.

5. After entering the information for a process, choose **Insert**.
6. If you wish to add more processes, repeat [step 4](#) and [step 5](#).
7. If you are done making changes, restart LiveServer.

Monitoring Processes

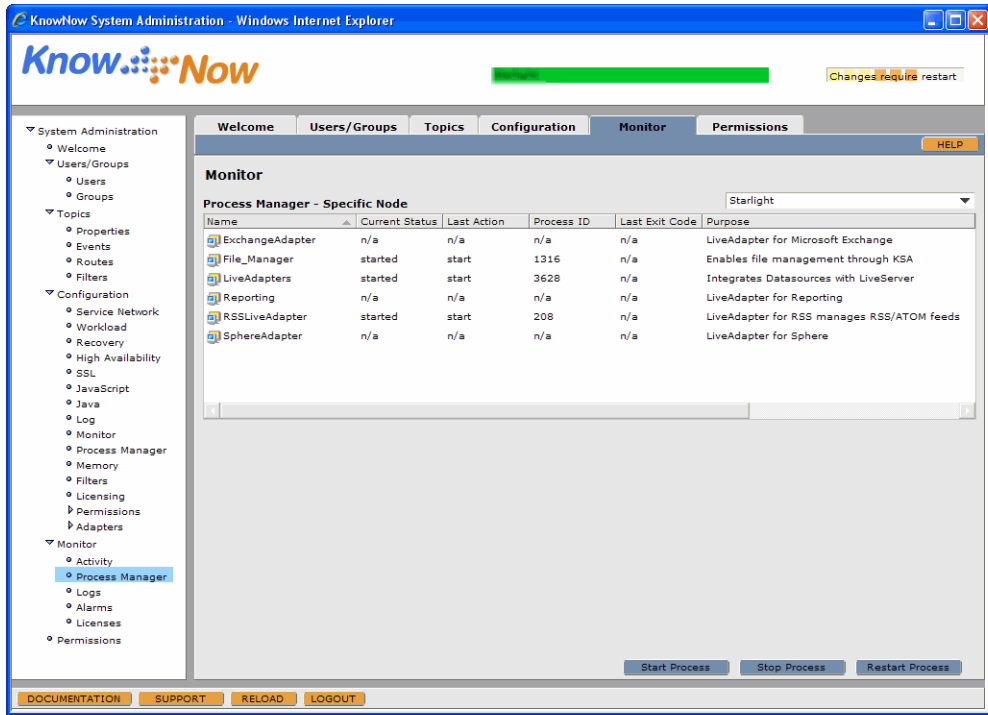
Once you have restarted LiveServer after configuring one or more processes, you can monitor those processes through the Process Manager's Monitor panel. If you have admin rights, you can start and stop processes here.

To monitor processes,

1. Access the KnowNow System Administration console.
2. Under **Monitor**, select **Process Manager**. The Process Manager panel is displayed. This panel is where you set your options for the Process Manager and where you manage processes. At a minimum, you should see the three core LiveServer

processes (described under “Configuring and Using the Process Manager” on page 293).

Figure 11-4. The Monitor > Process Manager panel.



3. In this panel, the columns are as follows:

- **Name.** This is the name given to the process. In the case of your processes, this is the name you entered in the **Name** field when adding the process as described under “Configuring Processes” on page 294.
- **Current Status.** The current status can be one of the following:
 - **started.** This means the process has been started.
 - **exited.** This means the process died or stopped by itself.
 - **stopped.** This means the process was manually stopped using the **Stop Process** button and is not running.
 - **restarted.** This means the process was restarted using the **Restart Process** button.

- **Last Action.** This indicates the last action that was made upon that process. The values can be **start**, **stop**, or **restart**, which mean that the process was started, stopped, or restarted, respectively.
 - **Process ID.** This displays the process ID (PID) as obtained by LiveServer from the operating system. If the process has exited or is not running, **(null)** will be displayed.
 - **Last Exit Code.** This applies only to a process that is not running or that exited previously while LiveServer was still up. Processes send their exit codes to the operating system. LiveServer retrieves those exit codes and displays them here. **n/a** means that an exit code doesn't apply.
 - **Purpose.** This is a description of the process's purpose. In the case of your processes, this is the text you entered in the **Purpose** field when adding the process as described under ["Configuring Processes" on page 294](#).
4. If you have administrative rights, you can start, stop, or restart a process by selecting it and choosing **Start Process**, **Stop Process**, or **Restart Process**.

Configuring and Viewing Logs

LiveServer maintains log files for each node. You can configure various aspects of log files and view the contents of the log files using the KnowNow System Administration console. These tasks are described under the following headings:

- [“Configuring Log Options”](#)
- [“Viewing Log Files” on page 306](#)

Configuring Log Options

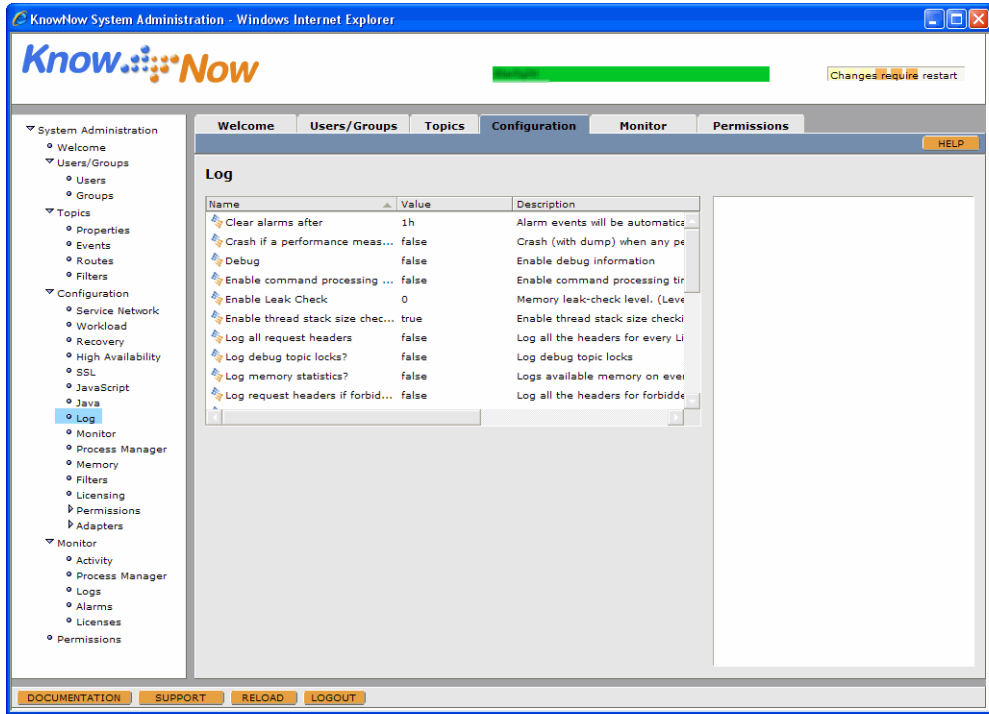
LiveServer offers a variety of log parameters to configure whether to maintain logs, how many log files to keep, and so on.

To configure logs options,

1. Access the KnowNow System Administration console.

2. Under **Configuration**, select **Log**. The Log panel is displayed. This panel is where you set your options for log files and related activities.

Figure 11-5. The Log panel.



3. For each parameter you wish to configure, select that parameter. A field appears to the right for modifying the value of that parameter. Modify the value of the parameter in the field to the right, then choose **Update**. The meanings and values of the parameters are provided under “[Log Configuration Parameters](#)” on page 302.
4. If you are done making changes, restart LiveServer.

Log Configuration Parameters

The log parameters tell LiveServer whether to maintain logs, how many log files to keep, and so on. For an explanation of the format for units in parameters, see “[Entering Data and Editing LiveServer Parameters](#)” on page 53.

Clear alarms after

Sets an interval of time after which alarms will be cleared. The default value is **1h** (one hour).

Crash if a performance measurement is out of range

Causes LiveServer to crash (with a dump) if any of the performance measurements are out of range. See also [“Performance range for background operations” on page 304](#) and [“Performance Measurement and Tuning” on page 288](#). The default value is **false**.

Debug

Enables debugging statements for the LiveServer core. Enable this parameter only when asked to do so by KnowNow Support. The default value is **false**.

Enable command processing time measurement

Enables measuring of the command processing time. The default value is **false**.

Enable Leak Check

Gathers statistics for internal data structures and their memory allocations. Change this value from the default value only when collaborating with KnowNow technical support unless you are debugging a filter module (for more information on that topic, see [“Filter Module Debugging Options” on page 98](#)).

The default is **0-None**, meaning that it is disabled. The other possible value is **1-Simple**.

Enable Stack Check

Enables checking of the thread stack size. The default value is **true**. See also [“Performance Measurement and Tuning” on page 288](#).

Enable thread stack size checking

Enables checking of the thread stack size. The default value is **true**.

Log all request headers

Specifies whether to log all the headers for every routing request. The default value is **false**.

Log debug topic locks?

Specifies whether to log debug topic locks. The default value is **false**.

Log memory statistics?

Specifies whether LiveServer should log the statistics of how many things were looked at and reaped (deleted) on each run. The default value is **false**.

Log request headers if forbidden

Specifies whether to log the headers for LiveServer requests that resulted in a “forbidden” response. The default value is **false**.

Maintain Access Logs

Specifies whether LiveServer is to keep an *access.log* file, which has entries for every GET and PUT command. Access logs are kept in classic NCSA format. The default is **false**.

Max Backup

Specifies the number of old rolled logs to keep, after which the oldest ones are deleted. The log files are named *logfile.001*, *logfile.002*, and so on, where .001 is the file for yesterday, .002 is for two days ago, and so on. The files are renamed every day.

The default value is **10**.

Maximum response time allowed for any command processing

Sets the maximum amount of time allowed for any command to process. If a command takes longer than the specified time to process, a log entry is generated. The default value is **10s**. A value of **0** means unlimited response time. See also [“Performance Measurement and Tuning” on page 288](#).

Performance range for background operations

Specifies a time within which database and reaper operations should complete (or assert). See also [“Crash if a performance measurement is out of range” on page 303](#) and [“Performance Measurement and Tuning” on page 288](#).

The default value is **20m** (20 minutes).

Performance range for fast operations

Specifies a time within which event operations should complete (or assert). See also [“Crash if a performance measurement is out of range” on page 303](#) and [“Performance Measurement and Tuning” on page 288](#).

The default value is **60s** (60 seconds).

Post logs and alarms as events

Specifies whether to run `knlogd` to create events for alarms and log information. The default value is **false**. If you specify **true**, then alarms will be posted to the **Monitor > Logs** screen (discussed under “[Viewing Log Files](#)” on page 306) and the **Monitor > Alarms** screen (discussed under “[Viewing Alarms](#)” on page 310).

Roll Log Time

Specifies the hour of the day (optionally, to the minute) that LiveServer is to roll logs.

Use [Roll Logs](#) to specify whether LiveServer is to create rolling logs.

The format is in 24-hour time specified using this syntax: (0-23[:00-59]). For example, a value of 15 would indicate that the logs are to roll at 3:00 PM. As another example, a value of 8:15 would indicate that the logs are to roll at 8:15 AM.

The default value is **00:30**, meaning 30 minutes after midnight.

Roll Logs

Rolling logs are log files that are created at specified time intervals. Each time a log is rolled, the contents of that log are saved to a separate file, and a new log file is begun. This parameter specifies whether LiveServer is to create rolling logs. The default value is **true**, which means that LiveServer will create rolling logs.

Use [Roll Log Time](#) to set the time of day for LiveServer to roll logs.

Server log file (base) name

Specifies the name of the LiveServer’s log file. The default is **server.log**.

Trace configuration subsystem

Used for advanced support diagnostics. The default value is **false**.

Trace event expiration and periodic event subsystem (reaper)

Used for advanced support diagnostics. The default value is **0 (No tracing)**. Other options provide for tracing various reaper activities.

Trace LiveServer TCL Interface

Used for advanced support diagnostics. The default value is **false**.

Trace network (post & HTTP request) subsystem

Used for advanced support diagnostics. The default value is **false**.

Trace statistics-handling

Used for advanced support diagnostics. The default value is **false**.

Trace TCL scripts (configuration and permissions)

Used for advanced support diagnostics. The default value is **false**.

Trace tunnel (delivery) subsystem

Used for advanced support diagnostics. The default value is **false**.

Viewing Log Files

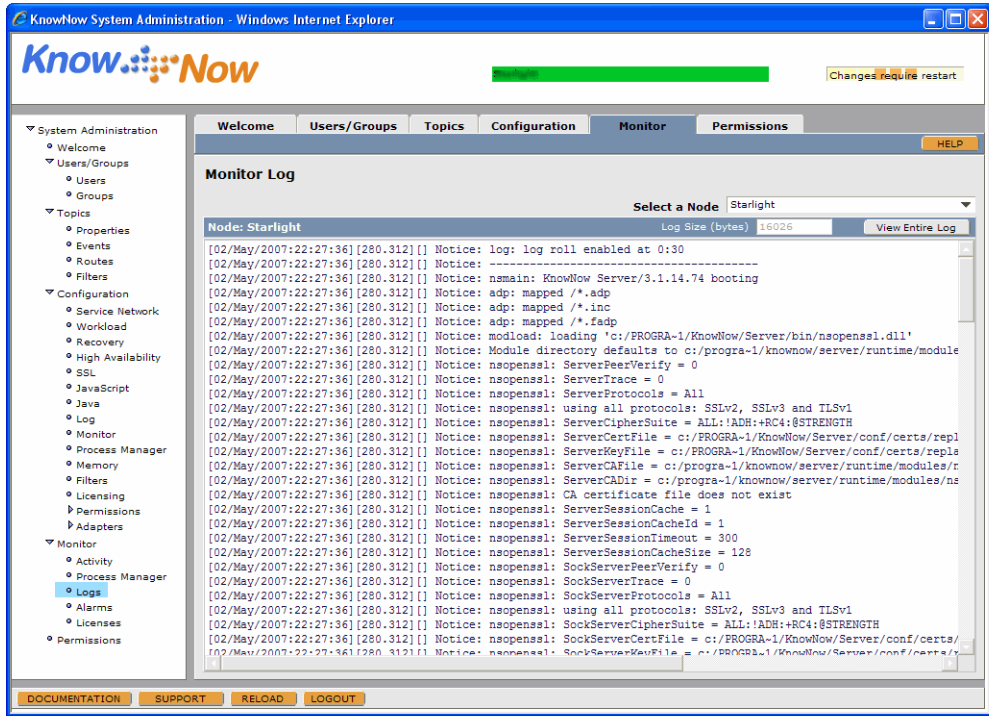
If you have enabled the **Configuration > Logs > Post logs and alarms as events** parameter, LiveServer will maintain log files for each node. You can view the contents of these files using the KnowNow System Administration console.

To view the log file for a node,

1. Access the KnowNow System Administration console.
2. Under **Monitor**, select **Logs**. The Log panel displays.

3. In the **Select a Node** field, select a node. Events for that node from the most recent 12 hours are displayed (where each event can be up to 64K in size). The **Log Size (bytes)** field shows the size of the log file.

Figure 11-6. The Monitor Log panel showing the contents of a log file.



4. To view the contents of the log, choose **View Entire Log**. A separate browser window opens displaying the entire contents of the log file.

Configuring and Monitoring Alarms

Alarms are messages warning of conditions that affect ability of the system to function. Using the KnowNow System Administration console, you can configure which alarms you wish to have sent and you can monitor the status of your system so you can respond as needed. These tasks are described under the following headings:

- [“Configuring Alarms”](#)
- [“Viewing Alarms” on page 310](#)

Configuring Alarms

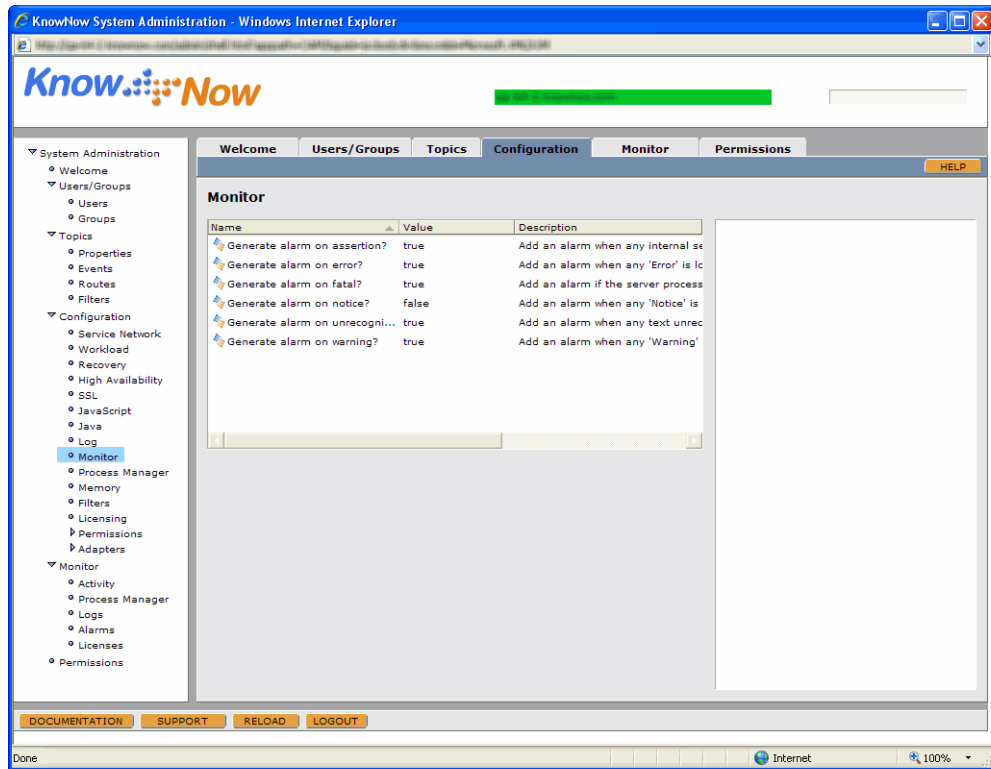
Alarms are divided into levels of severity, from notice to assertion. (The meanings of each level are described in the following procedure.) By default, alarms are disabled. When alarms are enabled, LiveServer has default settings for which alarms are sent to the logs. You can change those settings using the **Configuration** and **Monitor** panels.

To configure alarm monitoring,

1. Access the KnowNow System Administration console.
2. Under **Configuration**, select **Log**, then set the value of the **Post logs and alarms as events** parameter to **true**. This enables logs and alarms.

- Under **Configuration**, select **Monitor**. The Monitor panel is displayed. This panel is where you set your options for alarms.

Figure 11-7. The Monitor panel.



- Set each alarm severity level to **true** or **false** according to your needs. To set each severity level, click on it and use the field that appears to the right. When you set an alarm severity level to **true**, the alarms for that level will be sent to the log. Alarms are posted to the topic for the node that they apply to.

There are six alarm severity levels, presented here in increasing order of severity:

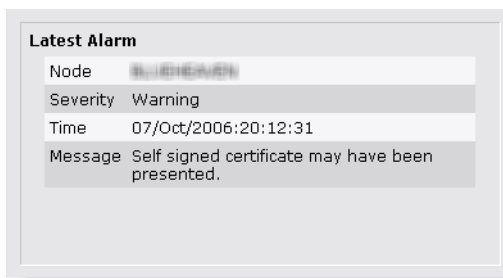
- The **Generate alarm on notice** level indicates that something is out of the ordinary but does not affect system performance. The default value is **false**.
- The **Generate alarm on unrecognized log entries** level indicates that text that is not recognized as a log entry has been logged. The default value is **true**.
- The **Generate alarm on warning** level indicates that something is out of the ordinary and may need attention. The default value is **true**.

- The **Generate alarm on error** level indicates that an error has occurred that needs your attention. The default value is **true**.
 - The **Generate alarm on fatal** level indicates that something is seriously wrong with that system and that the system requires a restart (after repairing the problem as needed). The default value is **true**.
 - The **Generate alarm on assertion** level indicates that something has happened with LiveServer internally. Please report any such problems to KnowNow. The default value is **true**.
5. You may wish to set the **Configuration > Log > Clear alarms after** parameter. This parameter sets a time after which LiveServer will clear the alarm events. For more information, see [“Clear alarms after” on page 303](#).
 6. If you are done making changes, restart LiveServer.

Viewing Alarms

If you have enabled the **Configuration > Logs > Post logs and alarms as events** parameter, LiveServer will maintain alarms for each node and the most recent alarms will be indicated on the Welcome tab of the KnowNow System Administration console, as shown in [Figure 11-8](#). You can also view all alarms through the Alarms panel in the KnowNow System Administration console as described in this section.

Figure 11-8. Alarm display.

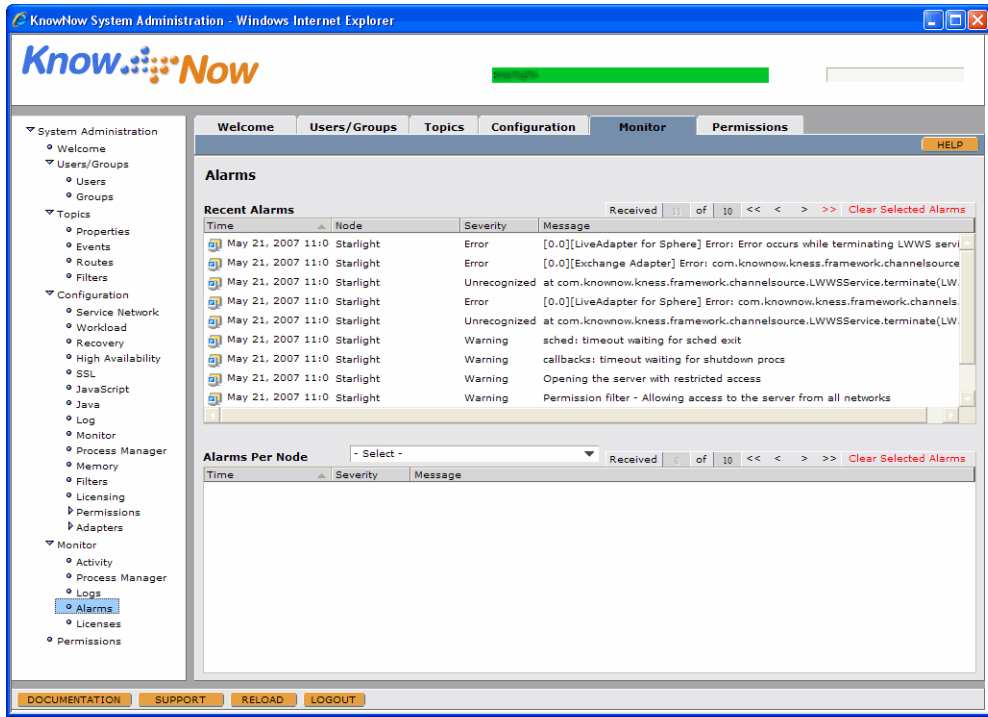


To view alarms,

1. Access the KnowNow System Administration console.

- Under **Monitor**, select **Alarms**. The Alarms panel displays, listing recent alarms.

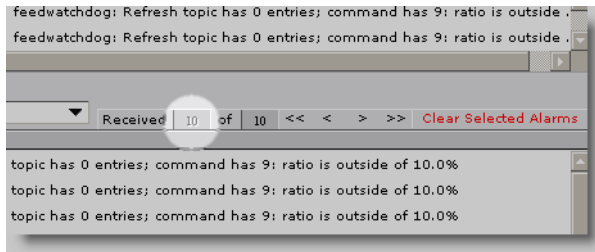
Figure 11-9. The Alarms panel showing the alarms for a node.



- To view the alarms on a specific node, in the **Alarms Per Node** field, select a node. The alarms for that node are displayed.
- You can navigate through the alarms by pressing the <<, <, >, and >> buttons, which get the first n records, the previous n records, the next n records, or the last n records, where n is defined by changing the value on the field just preceding these

buttons. For example, if you enter 5 in this box, the >> button will retrieve the last 5 alarms.

Figure 11-10. The alarm navigation increment field.



5. You can manually clear individual alarms in either the Recent Alarms table or the Alarms Per Node table by selecting an alarm (or by pressing and holding the control key and clicking on multiple alarms) and choosing **Clear Selected Alarms**. (In addition, you can set the may wish to set the **Configuration > Log > Clear alarms after** parameter to have LiveServer automatically clear alarms after a specified amount of time.)

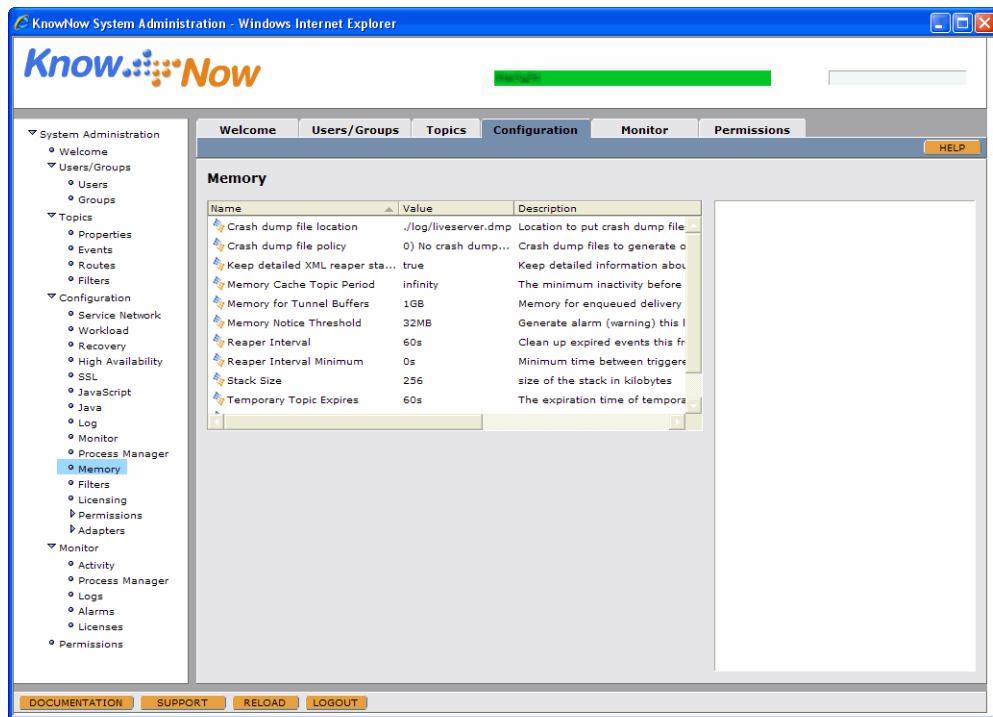
Configuring Memory

The memory parameters tell LiveServer what to do with temporary topics, tunnel errors, and so on.

To configure memory,

1. Access the KnowNow System Administration console.
2. Under **Configuration**, select **Memory**. The Memory panel is displayed. This panel is where you set your options for memory. One of these options determines whether an alarm will be reported and under what circumstances.

Figure 11-11. The Memory panel.



3. For each parameter you wish to configure, select that parameter. A field appears to the right for modifying the value of that parameter. Modify the value of the parameter in the field to the right, then choose **Update**. The meanings and values of the parameters are provided under “[Memory Configuration Parameters](#)” on page 314.

4. If you are done making changes, restart LiveServer.

Memory Configuration Parameters

The memory parameters tell LiveServer what to do with temporary topics, tunnel errors, and so on. For an explanation of the format for units in parameters, see [“Entering Data and Editing LiveServer Parameters”](#) on page 53.

Crash dump file location

Specifies a location for crash dump files. When specifying a location, use forward slashes (/) to separate directory names. Also, if the location names a directory, a unique name is generated for each separate crash file.

The default value is `./log/liveserver.dmp`, which places the files in the `/log/` directory under the LiveServer’s installation directory.

Crash dump file policy

Specifies a policy for handling crash dumps. The default value is **0) No crash dump file**.

Keep detailed XML reaper statistics

Specifies whether to keep detailed information about reaper performance in the file `statistics.xml`. The default value is **true**.

Memory Cache Topic Period

As an optimization, events from idle topics are swapped out of memory after a specified period of time. This makes it possible for the topic database to grow to sizes larger than the main RAM of the node.

This parameter sets the minimum inactivity period that must elapse before events are removed from memory. The default value is **7d** (7 days).

For a related setting, see the [Stack Size](#) parameter.

Memory for Tunnel Buffers

Sets the size of memory for enqueued delivery event data. The memory is shared across all subscribers. The default value is **1GB** (1 gigabyte).

Memory Notice Threshold

If the number of bytes of virtual memory falls below the threshold specified in this parameter, a warning is logged. In Windows, when the amount of memory becomes available again, a clearing message to that effect is logged as a warning. The default value is **32MB**.

Reaper Interval

The reaper is a process that runs occasionally to delete expired events. The reaper interval is how often, in seconds, LiveServer runs the reaper to look for and delete expired events. Set this value to be lower than the default value for a large number of expiring events. The default value is **60s** (60 seconds).

Reaper Interval Minimum

Sets the minimum amount of time to elapse between triggered reaper runs. The default value is **0s** (zero seconds).

Stack Size

This parameter specifies how much memory to use (in kilobytes) before swapping out idle events. The default value is **192** (192 kilobytes).

For a related setting and more information on idle events, see the [Memory Cache Topic Period](#) parameter.

Temporary Topic Expires

The expiration time for all temporary topics. The default value is **60s** (60 seconds). The time value is always taken to be in the future; no preceding characters (e.g., +) are needed.

Use this parameter to assign an expiration time for all temporary topics. The clock starts ticking when a topic has no routes and no subtopics. When the time value set in this parameter is reached, the topic is deleted.

The status of a topic as being temporary is set using the [Temporary Topic Recognize](#) parameter.

To set the expiration time for a *specific* temporary topic, see **kn_temporary_expires** in the *KnowNow LiveServer Developer's Guide*.

Temporary Topic Recognize

Configures LiveServer to automatically recognize temporary topics. The default value is **true**.

If this parameter is set to **true**, any subtopic (other than journals) in any of the following topics will be recognized as temporary, with an expiration as set by **Temporary Topic Expires**:

- /who/wms
- /who/java
- /who/*/s/

If this parameter is **false**, these topics will not be recognized as temporary.

For an alternate method of specifying a temporary topic, see **kn_temporary** in the *KnowNow LiveServer Developer's Guide*.

Tunnel Error Tolerance

Specifies the number of errors to tolerate before closing the tunnel that the errors are on. The default value is **10**.

Chapter 12 Backup, Restore, and Other Utilities

LiveServer ships with several useful utilities; some you may wish to use frequently, and others you may only use occasionally. In addition, there are other activities you may wish to pursue to ensure data safety. This chapter discusses the LiveServer utilities, some backup tasks, and troubleshooting under the following topics:

- “The LiveServer Utilities” on page 318
- “Accessing the Event Publisher” on page 320
- “Backing Up Your Database” on page 321
- “Restoring Your Database” on page 322
- “Moving Recovery Information” on page 324
- “Troubleshooting” on page 325

The LiveServer Utilities

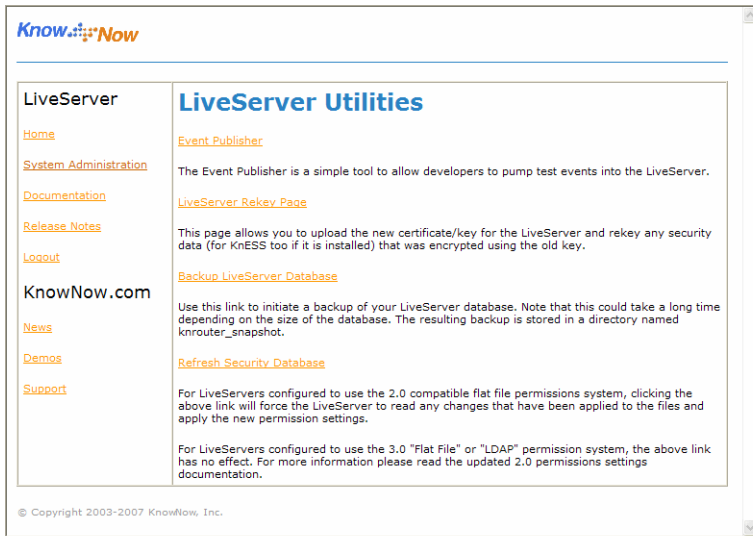
LiveServer ships with a number of useful utilities. In addition, you can use the KnowNow System Administration console to manage users, groups, permissions, topics, configuration, and performance. Use of this tool is described in earlier chapters in this book.

The utilities that are installed with the KnowNow LiveServer are

- The Sleepycat Berkeley DB utilities. This set of tools gives administrators the opportunity to verify and recover the state of LiveServer databases externally to LiveServer.
- The Event Publisher, which sends event to topics using various settings.
- The LiveServer Rekey page. Access this page to upload a new certificate/key for LiveServer and rekey any security data (including for ESS also, if it is installed) that was encrypted using the old key. For more information on using this utility, see [“Uploading Certificates and Keys” on page 80](#).
- A link for backing up your LiveServer database.
- A link for refreshing your security database. This is for LiveServers configured to use the 2.0 compatible flat file permissions system. Clicking this link forces LiveServer to read any changes that have been applied to the files and to apply the new permission settings. For LiveServers configured to use the 3.0 Flat File or LDAP permission systems, this link has no effect.
- OpenSSL, for creating security certificates for LiveServer. Use of this utility is described under [“Using OpenSSL to Generate Certificates and Keys” on page 78](#).

Some of these utilities are available from the LiveServer Utilities page, which is accessible by choosing **Utilities** from the LiveServer opening screen.

Figure 12-1. The LiveServer Utilities page.



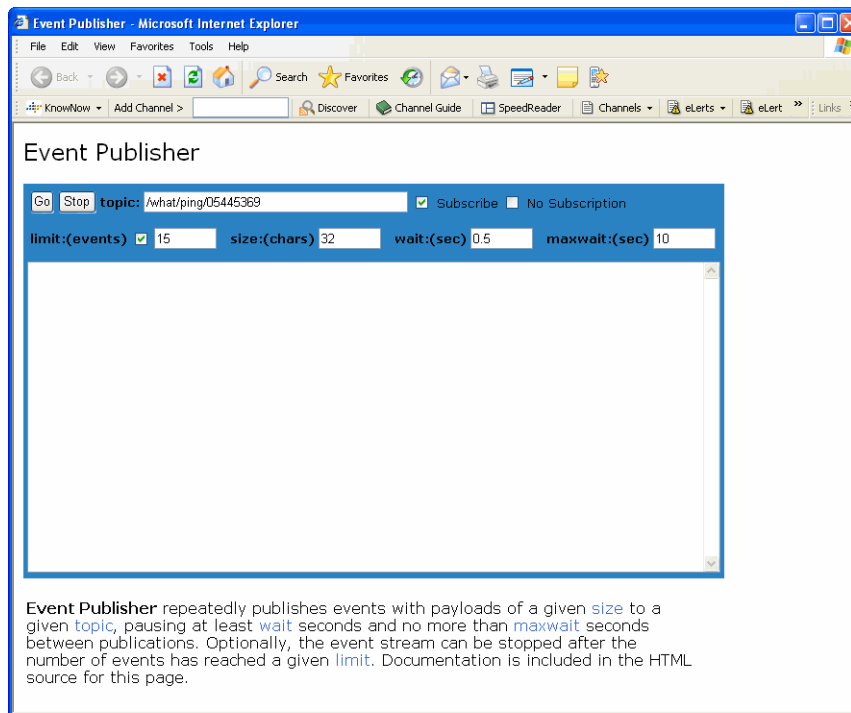
Accessing the Event Publisher

The Event Publisher repeatedly publishes events with payloads of a given size to a given topic, pausing for a specified number of seconds. The use of this tool is pretty self-explanatory. Additional documentation for the Event Publisher is provided in the Event Publisher HTML source code.

To access the Event Publisher,

1. Access the LiveServer opening page.
2. Choose **Utilities**.
3. The page that is then displayed provides links to the **Refresh Security Database** and **Backup LiveServer Database** commands and to the Event Publisher.
4. Choose **Event Publisher**. The Event Publisher appears in a new window.

Figure 12-2. The Event Publisher.



5. Use the information on that page to publish events.

Backing Up Your Database

Data integrity is critical to a production environment. As always, an adequate backup strategy is suggested per your platform's production recommendations. Please refer to your system's operating guides for further information.

LiveServer maintains a database. To ensure data integrity, we recommend that you make backing up this database a part of your regular backup routines. To help make this task a bit easier, we've provided a link to initiate a backup of your LiveServer database.

To back up your LiveServer database,

1. Access the LiveServer opening page.
2. Choose **Utilities**.
3. The page that is then displayed provides links to several LiveServer utilities.
4. Choose **Backup LiveServer Database**. When the process is complete, the window displays a 200 Notified message.

Restoring Your Database

LiveServer usually recovers automatically from a bad or corrupted database, but on rare occasions you may need to perform manual operations on your database files.

LiveServer ships with [Sleepycat's](#) highly useful database utilities, which you can use for backing up and restoring the LiveServer's database files. For information on the Sleepycat utilities, see <http://www.sleepycat.com>.

LiveServer has a hook that can be exploited to back up its database without shutting down LiveServer. If you send an event like the following,

```
http://host:port/kn/kn_system/recovery?kn_id=snapshot&do_method=notify
```

then

1. Your database (default `.../knrouter_db`) will be locked.
2. All of the database files will be copied into your snapshot directory (default `.../knrouter_snapshot`).
3. The database is then unlocked.

You can then copy or work with the snapshot database as you wish.

To start LiveServer from this data,

1. Stop LiveServer (one presumes LiveServer is already down and having trouble starting or you wouldn't need to do this).
2. Delete your `/knrouter_db/` directory (or rename it).
3. Copy the snapshot directory files into the `/knrouter_db/` directory (again, renaming works).
4. Start LiveServer.

Notes, caveats, and assumptions:

- If you're clustered, and another node fails and recovers, that node may request a snapshot after you asked for yours and before you've copied your version to your backup media. To minimize this risk, rename the snapshot directory as soon as you've gotten a 200 OK response from the snapshot request.
- No events will actually be delivered while the snapshot is in progress. As soon as it completes, waiting events will finish being posted.

- Do not just copy the files from the snapshot directory into your `/knrouter_db/` directory without deleting all the files in the destination directory. In particular, you should delete the files whose names end in numbers (for example, `__db.001`).
- If you subscribe to `kn/kn_config/recovery`, you can see some diagnostic information from the snapshot event's operation; that will be in its payload after completion of the operation.
- A snapshot can take a long time if your database has gotten big. It copies the files, not the data—with the LiveServer's DBMS, that can mean many more bytes get copied than are actually used.
- All publications block (that is, they don't receive an HTTP 200 OK) until after the snapshot completes. They are not queued; they are locked out (threads waiting on a condition), so the publication order might not be the same as if there had been no blockage (and might not match `kn_time_t` order, even if the time is assigned by LiveServer). LiveServer is still sending (and accepting) keepalives during that time. From the LiveServer's (and HTTP's) standpoint, the event isn't "sent" until the HTTP request is completed (in other words, when the 200 OK has come back).
- To make a snapshot, you need to be able to notify to the `/kn_config/recovery` topic. In a secured system, only the administrator and the **kninternal** user should be able to do so.

Moving Recovery Information

To move recovery information, you can use normal copy operations as long as you copy the entire recovery directory, which is specified by the **Configuration > Recovery > Database Directory** parameter. You can also use Sleepycat's database utilities for this purpose.

If you want to move your data to a different machine, and you wish to minimize the amount of data that is lost, perform the move operation in the following steps.

To move recovery information,

1. Change the **Configuration > Recovery > Snapshot Directory** parameter so that it points to where you are moving LiveServer.
2. Take a snapshot of your database using the instructions under “[Backing Up Your Database](#)” on page 321.
3. Move the snapshot to the new destination.
4. Use the KnowNow System Administration console to restart LiveServer.



Warning: You will lose any events that are posted between [step 2](#) and [step 4](#).

Troubleshooting

If your LiveServer does not start up, either on first installation or after you've made some configuration changes, check the log file, named *server.log*, which is written to the `/log/` subdirectory. This directory is located directly under the LiveServer's main installation directory. The log file is created when LiveServer is run as a daemon (or as a service on Windows). The *server.log* file contains diagnostics from LiveServer regarding itself.

You may also need to check the `knsettings` files to see if there is an entry preventing LiveServer from starting up; for example, if you selected the **High Availability > Multi-cast** option and your system is not clustered. If that is the case, edit the `knsettings` files to remove references to the module. A separate `access.log` file contains an entry for every HTTP connection to LiveServer.

You must also verify that any firewalls between clients and LiveServer are set up to allow connection on the chosen port.

Verifying the LiveServer Service Under Windows

Normally, you can use the instructions provided on [page 40](#) to verify that LiveServer is running under Windows. However, if for some reason those instructions don't work, you can use these alternate instructions.

To ensure the service is running under Windows,

1. From the **Start** menu, choose **Run**. In the **Open** text field, enter `C:\Program Files\KnowNow\Server\log\server.log`. (If you installed in other than the default location, replace the path with the path to where you installed LiveServer.)
2. Choose **OK**.
3. Scroll through the log and locate the line(s) which contain any of the following text (the Notice lines will be preceded by something like

```
[14/Mar/2003:16:02:04] [664.680] [thread]):  
Notice: nskn: http://yourhostname:8000/kn is a local_host  
Notice: nskn: http://123.245.7.89:8000/kn is a local_host  
Notice: nsock: listening on 123.245.7.89:8000
```

Write down any of the IP addresses or host names listed as a local host or listed as being listened on. Note also the port number.

4. From the Windows **Start** menu, choose **Run** and enter `http://IP_address:HTTP_port`. Use the IP address and port number you noted in [step 3](#). For SSL, use the HTTPS command (`https:// IP_address:HTTPS_port`).
5. Choose **OK**. The KnowNow LiveServer startup page displays as shown in [Figure 3-1 on page 46](#).

Glossary

action	In the security context, something a principal can do to some subject.
authentication	<p>The process of identifying an individual, usually based on a user name and password. Authentication merely ensures that the user is who he or she claims to be, but says nothing about the access rights of the individual. Access rights are defined through authorization. The encryption is done through secure sockets.</p> <p><i>See also “secure sockets” on page 331.</i></p>
authorization	The process of granting or denying access to a network resource. Most computer security systems are based on a two-step process. The first stage is authentication, which ensures that a user is who he or she claims to be. The second stage is authorization, which allows the user access to various resources based on the user’s identity.
batching	<p>Allows an HTTP client to connect to the KnowNow LiveServer and send a series of requests for different operations in one HTTP request. Batches can span multiple topics. The resulting codes for each batch request are sent as part of the HTTP body.</p> <p>Always use the <code>kn_status_</code> parameter with the JavaScript Connector when batching. Outside the JavaScript Connector, batching does not provide any benefits over multiple posts over a persistent connection.</p>
certificate	A certificate contains a distinguished name, public key, private key, and information about the issuer of the certificate.
certification authority	An entity that issues certificates, usually associated with an authentication server. You can check the validity of an issued certificate by checking it against the appropriate authentication server.

certificate file	The storage location for one or more certificates. A certificate file can store certificates, trusted roots, and uncertified key-pairs. An uncertified key pair is one that has not yet been certified by the Certification Authority. <i>See also “trusted root” on page 332.</i>
CLASSPATH	The environment variable that tells the Java compiler where to look for the classes it needs. -classpath is an option to the Java interpreter and the Java compiler that tells them (while compiling or running) where to look for a class. -classpath overrides CLASSPATH.
clients	A server, service, browser, device, or application.
Connector	Specific API libraries and components released by KnowNow to allow applications to interface with LiveServer. A Connector makes it possible for application programmers to easily use the LiveServer’s publish and subscribe features. To download a Connector, see http://support.knownow.com .
control container	An ActiveX-enabled application or development environment that supports programmatic access to ActiveX controls.
do_method	A command sent to LiveServer.
duplicate event squashing	The KnowNow LiveServer allows once-only delivery of events; duplicate events are deleted. This is called duplicate event squashing.
encryption	The scrambling of data so that it can only be unscrambled by an authorized recipient. Encryption strength is usually stated in bits.

event

An event is a message containing some kind of data that is sent to LiveServer by an application; the data is then available on LiveServer for other applications to retrieve. Events have an identification string, a timestamp, an expiration timestamp, metadata headers, and a content payload. The content payload in KnowNow events can be data in any format.

Events are shared with other applications by being sent to various topics, which are locations on LiveServer that are used to collect, store, and share related events. An application sending the data to the topics is called a [publisher](#). An application which is interested in viewing the data subscribes to the topic of interest; hence, this kind of application is called a [subscriber](#). A single application can both publish events (data) to topics and subscribe to other topics for the events being published to those topics. Applications can use a uniform resource locator (URL) to access topics on a LiveServer.

As events are updated, all subscribers are notified of the changes. Duplicate events that are published to a topic are automatically deleted by LiveServer. Events that have expired are also deleted by LiveServer.

More technically, an event is a paired header and value, sent via HTTP/HTTPS, that contains the user's data and other headers that define properties of the event. LiveServer reads the HTTP message's body, extracts the header/value pairs, and builds the event, storing the information in a topic.

Although LiveServer has KnowNow-specific headers (which are named starting with the letters kn_), LiveServer also handles events with custom headers; therefore, you can create your own custom headers as needed. Also, although you cannot name your own custom headers with kn_, you can freely use the majority of the KnowNow headers.

See also [“publish” on page 331](#), [“subscribe” on page 332](#), and [“topic” on page 332](#).

event aggregation

Subscribing a topic to another topic. Using these inter-topic subscriptions, events can be aggregated from multiple topics to a single topic.

event forwarding

Distributed published events to all clients subscribed to a topic. The KnowNow LiveServer maintains a table of subscribers for each topic so that it knows how to distribute published events.

Subscriptions are automatically deleted when they expire or when the KnowNow LiveServer cannot deliver an event to the given subscriber. Clients can also delete subscriptions. KnowNow LiveServers send events as HTTP POST in KN event format (which is x-www.url-form-encoded with special KnowNow fields), or using a JavaScript tunnel or a simple tunnel for HTTP responses.

See also [“tunnel route” on page 333](#).

event replay	A request submitted to the KnowNow LiveServer to send all events that occurred since a specific point-in-time. You can specify a timestamp, request the last <i>n</i> events, or replay all stored events. This event replay request is executed immediately to only the specific subscriber that made the request.
event updating	KnowNow LiveServer property that allows the contents of an event to be updated while maintaining the same event identification number.
filter	See “ modules ” on page 330.
firewall	A firewall helps prevent unauthorized access to a server. LiveServer requires you to open port 8000 for non-SSL connections. For SSL connections, port 8443 is used.
JavaScript Connector vendor	The KnowNow LiveServer can vend the JavaScript Connector, so that the Connector downloads into a browser automatically with a KnowNow-enabled Web page. As a result, end users can use KnowNow applications that do not require additional installs to their browsers.
journal	JavaScript tunnels and simple tunnels use “journaling” topics that know how to multiplex events from many topics through them, such that the Connectors can demultiplex them on the other end. Journaling topics are normal topics without duplicate squashing or event updating.
LDAP	Lightweight directory access protocol. A family of standards and recommendations for organizing and distributing information about users, groups, and applications.
LiveServer	The Web messaging middleware that provides asynchronous message routing over HTTP/HTTPS. The core communication model is publish and subscribe.
modules	Modules are code components—either dynamic linked libraries (DLLs) or shared objects (SOs)—that you write using the KnowNow LiveServer API, and then load into LiveServer using the KnowNow System Administration console. Essentially, a module is a plug-in that performs specific actions with events, topics, routes, and security.
persistent storage	Persistent storage uses operating system asynchronous I/O to write events to disk. This is used for events that should be delivered, but no significant problems will occur if they are not.
polling client	A program that periodically attempts to write or read data from a socket, typically using a timer. The program continues to do other tasks while waiting for a reply. Responses to requests arrive in order.
principal	A person or organization, or a software entity representing same, which can be identified and may be permitted some sort of access to a LiveServer.

provider	An application that accepts data from another application(s) and processes this data for a specific function.
publish	<p>The act of sending an event from a KnowNow LiveServer. Any event that is sent to a topic is “pushed” to all subscribers listening on that topic. The routing of events happens instantly after the events are received. Events persist as long as the event expiration timestamp does not expire.</p> <p><i>See also “event” on page 329, “subscribe” on page 332, and “topic” on page 332.</i></p>
publisher	An application that sends an event to a topic on LiveServer.
requestor	An application that uses the system to send data to another application that processes the data and returns back a response.
return receipt	An event to acknowledge the receipt of a request by a provider.
route	The path an event travels during the publish/subscribe process. In the context of permissions, a route is also used to create or modify a subscription.
secure sockets	<p>A socket is a way of sending information over a network between two programs. Secure sockets make this communication safe. Data is encrypted so only people who the data is meant for can read it.</p> <p><i>See also “encryption” on page 328.</i></p>
SOAP	Simple object access protocol. SOAP is a clean, lightweight, XML-based protocol for exchange of information in a decentralized, distributed environment. The SOAP protocol consists of three parts: an envelope that defines a framework for describing what is in a message and how to process it, a set of encoding rules, and a convention for representing remote procedure calls and responses.
SSL	<p>Secure socket layer standard. A protocol that delivers server authentication, data encryption, and message integrity. SSL is layered beneath application protocols such as HTTP, SMTP, Telnet, FTP, Gopher, and NNTP, and layered above the connection protocol TCP/IP. This strategy allows SSL to operate independently of the Internet application protocols. With SSL implemented on both a client and a server, your Internet communications are transmitted in encrypted form. Information you send can be trusted to arrive privately and unaltered to the server you specify.</p> <p><i>See also “secure sockets” on page 331.</i></p>

subscribe	<p>The act of receiving an event from the KnowNow LiveServer. When a client connects to a LiveServer and subscribes to a topic, the client begins to receive events that are published to that topic. The client continues to receive events, until the subscription has expired.</p> <p>See also “event” on page 329, “publish” on page 331, and “topic” on page 332.</p>
subscriber	<p>An application that connects to LiveServer to receive events stored in a topic.</p>
subject	<p>A software construct that can be protected. In the case of LiveServer, these are topics, events, or routes.</p>
subtopic	<p>See “topic” on page 332.</p>
timestamp	<p>Timestamp recording the original event receipt time. Time is expressed as seconds since Jan 1 1970, 00:00:00 (UNIX time() format; for example, “979273892” for an event created Fri Jan 12 04:31:32 2001).</p>
topic	<p>A location on LiveServer to which applications can subscribe or publish events. Clients can use a uniform resource locator (URL) to refer to and access topics on LiveServer. Topics are arranged in a hierarchy that uses forward slashes (/) to separate the levels. For example: /what/nasdaq/SUNW/bid. A subtopic is simply a topic that is below another topic in the hierarchy.</p> <p>All events are published to some topic. All subscribers subscribe to a topic to retrieve events. As updates occur in a topic, those updates are sent to all subscribers to that topic.</p> <p>Duplicate events that are published to a topic are automatically deleted by LiveServer. You can attach metadata to a topic by using the kn_metadata header. LiveServer has a configurable payload size threshold to discard large events without overloading.</p> <p>See also “event” on page 329, “publish” on page 331, and “subscribe” on page 332.</p>
topic management	<p>The process of creating, editing, or deleting topics.</p>
topic space	<p>The hierarchical structure of topics within a KnowNow LiveServer.</p>
trusted root	<p>A special certificate issued by a well-known and trusted certification authority.</p>

tunnel route	The physical route that connects a KnowNow LiveServer to a client, like a Connector or a standalone HTTP application. Tunnels use persistent connections to send events to desktops and applications while maintaining the policies of Internet administrators for their firewalls, network address translators (NATs), and Web proxies. To maintain a persistent connection, tunnels occasionally send keepalives to the KnowNow LiveServer. Keepalives are single byte events. A client can request that the KnowNow LiveServer close a tunnel at a specific time by setting the <code>kn_expires</code> header parameter.
URI	Uniform resource identifier (also known as URL; uniform resource locator; URL is an alternate name for URI). A unique address that allows applications, such as browsers, to locate documents, images, files, services, electronic mailboxes, and other resources on the Internet. Different access methods, such as HTTP, FTP, and Internet mail, can use URIs. For more information, see http://www.w3.org/Addressing/ .
URL root	<p>The URL root is that part of the HTTP command that, when present, is a flag to LiveServer telling it to handle that information as a LiveServer request. If the information flowing into LiveServer does not contain the URL root, the information is handled by LiveServer as a normal HTTP request, just as any Web server might.</p> <p>The URL root for LiveServers is <code>/kn?</code>. Everything after the URL root comprises LiveServer commands.</p>
XML	The extensible markup language (XML) is a universal format for structured documents and data on the Web. For more information on XML, refer to http://www.w3.org/XML .
XPath	XPath is a language for addressing parts of an XML document, designed to be used by both XSLT and XPointer. For more information on XPath, refer to http://www.w3.org/TR/xpath .
XSLT	XSLT is a language for transforming XML documents into other XML documents. For more information, refer to http://www.w3.org/TR/xslt .
WSDL	Web services description language; defines a SOAP interface.

Index

Symbols

/ (slashes), LiveServer URL prefix and 89
\$ (dollar sign) in knsettings.conf 54
// (double slashes), LiveServer URL prefix
and 89
(parameter) 227
* (asterisk) 53
[] (brackets) in knsettings.conf 54
\ (backslash) in knsettings.conf 54
{ } (braces) in knsettings.conf 54
~KnDummy (user name) 252

Numerics

2.0 Compatible Files, refreshing security
databases 318

A

access control lists (ACL)
AD, editing 215
access control. *See* [permissions](#).
access.log 304, 325
access_patterns 241
access patterns. *See* [permissions](#).
ACL (access control lists)
AD, editing 215
action, definition of term 327
Active Connections 291
Active Directory (AD)
ACLs, editing 215
authorization schema 164

Active Directory (AD) (*continued*)
integration 205–214
administrative user 205
authorization data 210
authorization schema 206
groups 207
LSadmin user privileges 215
users 207
validation 214
knadmin password 253
LDAP module, configuring 211
LSadmin user and 169
Perl preparation 163
User identifier attribute parameter and 232
with ADAM. *See* [Active Directory with ADAM](#).
with OpenLDAP. *See* [Active Directory with OpenLDAP](#).
Active Directory Users and Computers
program
creating LSadmin user 205
removing users from Domain Admins
group 215
validating LiveServer users with 174, 183,
208, 210
Active Directory with ADAM 179–187
administrative user 180
authorization data 183
authorization schema 181
ESS container 180
LDAP module, configuring 184

- Active Directory with ADAM (*continued*)
 - users and groups 181
 - validation 186
- Active Directory with OpenLDAP 170–178
 - administrative user 171
 - authorization data 172
 - authorization schema 172
 - LDAP module, configuring 174
 - OpenLDAP, validating 170
 - users 173
 - validation 177
- Activity 290
- Activity - Specific Node 290
- AD (Active Directory). *See* [Active Directory \(AD\)](#).
- ADAM
 - administrative user 180
 - authorization data 183
 - authorization schema 181
 - installation notes 179
 - Perl preparation 163
 - with AD. *See* [Active Directory with ADAM](#).
- Add Group 242
- Add Group to Pattern 242
- Add New Property (button), event properties 139
- Add Property to Route (command) 145, 146
- addresses, IP. *See* [IP addresses](#).
- Add User 242
- Add User to Pattern 242
- /admin, configuring access to 237
- ADSI Edit
 - AD
 - LSadmin user and 205, 215
 - requirement for 161
 - ADAM
 - creating LSadmin user 180
 - requirement for 161
- Advanced LDAP Configuration 229
- Alarm events expire (parameter) 107
- alarms
 - about 308
- alarms (*continued*)
 - alerts, configuring expiration time 107
 - clearing 312
 - automatically 303, 310
 - cluster states 50
 - configuring 309
 - enabling 305
 - severity levels 309
 - thresholds, ESS 112
 - viewing 310
- Alarms (command) 311
- Alarms Per Node 311
- alerts, configuring expiration time 107
- aliases, entering 60
- All (access permission) 235
- All Events (button) 137
- Allow anonymous login (option)
 - LDAP, replacement for 226
- Always use HTTPS login (parameter) 60
- Anonymous, credentials on demand (policy) 155
- Anonymous users expire after 107
- APIs (application programming interfaces), documentation for xviii
- applications, events and 2
- Arguments (field) 296
- assertions (alarms) 310
- asterisk beside data entry fields 53
- Atomic 126
- atomic mode 71
- Attach Filter (button) 146
- Attach Filter (command) 130
- Attach Route Filter (command) 145, 147
- Attach Selected Filter (command) 130
- audience for book xvii
- Audit (parameter)
 - Flat Files 154
 - LDAP and LDAP Test 226
- Audit Forbidden (parameter) 154
- authentication/authorization
 - See also* [permissions](#).
 - definitions 150
 - ESS and 155
 - logging authorization decisions 154, 226
- Authentication policy, LDAP parameter

- (parameter) 226
- Authentication policy (parameter) 154
- authorization data
 - ADAM 183
 - AD with OpenLDAP 172
 - dual OpenLDAP 190
 - standalone AD 210
 - standalone OpenLDAP 200
- authorization schema
 - ADAM 181
 - AD with OpenLDAP 172
 - dual OpenLDAP 190
 - standalone AD 206
 - standalone OpenLDAP 199
- Autostart (field) 297
- Available Filters list 130

B

- backslash (\), in knsettings.conf 54
- Backup LiveServer Database utility 321
- backups, upgrades and 18
- basic concepts 2
- Berkeley DB (Sleepycat) 69
 - utilities 318, 322
 - recovery and 324
- Bind timeout (parameter) 230
- boldface type, meaning of in book *xix*
- book
 - contents *xvii*
 - conventions *xix*
- braces ({ }) in knsettings.conf 54
- brackets ([]) in knsettings.conf 54
- browsers, deleting events with 137

C

- CA (certificate authority). *See* [certificate authority \(CA\) files](#).
- Certificate (field), LDAP servers 224
- certificate authority (CA) files
 - generating 78
 - keys, passwords and 78
 - LDAP, LDAPS:// mode and 225
 - SSL and, format 77, 78

- certificate authority (CA) files (*continued*)
 - trusted, adding certificates 82
 - uploading 80
- certificate signing request (CSR),
 - generating 79
- certificates. *See* [certificate authority \(CA\) files](#).
- Change (access permission) 235
- characters
 - special, in knsettings.conf 54
- Checkpoint Every Write 71
- Checkpoint Interval 57, 71
- classes
 - imported into LiveServer 92
 - Java, adding 92
 - LiveServer 93
- Classpath for Java VM 92
- Clear alarms after, alarm configuration
 - and 310
- Clear alarms after (parameter) 303
- Clear All Routes (button) 143
- Clustering Interface (parameter), Multicast
 - and 273
- Cluster port (parameter) 274
- clusters 263–277
 - adding nodes to 278
 - capabilities 264
 - checklist 269
 - configuring, recommended practices 268
 - database directory 72
 - default setting 57
 - deleting nodes 281
 - domain names and 265
 - Domain parameter and 62
 - editing nodes 280
 - failover mechanism 266
 - filter modules and 265
 - flow control 266
 - host names 62, 265
 - allowed length 272
 - journals and 265
 - kn_authorize and 265
 - knmcluster. *See* [Multicast](#).
 - LAN Multicast. *See* [Multicast](#).
 - LDAP integration and 161

- clusters (*continued*)
 - LiveAdapter 282
 - Linux 282
 - Windows 2000 and older 282
 - Windows 2003 284
 - load balancers and 62, 268, 273
 - load balancing and 267
 - nodes
 - adding 278
 - assimilating new nodes 72
 - configuration of 264
 - deleting 281
 - hot additions and deletions 264, 268
 - security modules and 264
 - number of machines and 265
 - operations of 265
 - overview 264
 - ports 62, 63
 - recovery 69, 266
 - scalable connections 266
 - security and 269
 - Service Network parameters for 60
 - Spread process and 266
 - states 50
 - topics and 265
 - user names and passwords 269
 - WAN, recovery 275
 - WAN Multicast. *See* Multicast.
- code
 - ellipsis in *xix*
 - typeface, meaning of *xix*
- comKnowNowAccessPattern 165
- comKnowNowAuthorizationEntry 165
- command-line utilities. *See* utilities.
- /conf/ directory, backups and 18
- /configuration, configuring access to 237
- configuration 55–112
 - cross-domain, LiveBrowser 88
 - files
 - knrouter.conf. *See* liveserver.tcl.
 - liveserver.tcl. *See* liveserver.tcl.
 - Linux 17
- connections
 - HTTP, log file for 325
- connections (*continued*)
 - scaling 266
 - simultaneous, setting number of 39, 68
- Connectors
 - functions of 10
 - initialization and 8
 - JavaScript. *See* JavaScript Connector.
 - journals and 8
 - overview 4
 - topics and, naming conventions 8
 - tunnels and 8
- contacting KnowNow *xx*
- containers for ESS
 - ADAM 180
 - AD with OpenLDAP 171
 - dual OpenLDAP 189
 - standalone AD 205
 - standalone OpenLDAP 198
- contents of book *xvii*
- copyright statements *ii*
- Crash dump file location 314
- Crash dump file policy 314
- Crash if a performance measurement is out of
 - range (parameter) 288, 303
- Create Route (button) 142
- Create Route (dialog box) 142
- Create Subtopic (button) 125
- Credentials, anonymous on demand
 - (policy) 155
- Credentials required - no anonymous
 - (policy) 154
- credentials. *See* permissions.
- cross-domain installations
 - applications 86
 - Domain parameter and 86
 - example 88
 - LiveBrowser and 88
 - LiveServer URL prefix and 89
 - parameters for 86
 - security and 85
- cross-domain Web applications, LiveServer
 - configuration changes 86
- CSR (certificate signing request),
 - generating 79
- curly braces. *See* braces.

Current Status (field), Process Manager 299
 Customer added classes 92
 customer ID, Sphere 111
 custom events. *See* [events](#).
 customization. *See* [configuration](#).
 custom login forms 158

D

Daemon 259
 Daemon Password 258
 Daemon User 259
 database
 backing up 318
 operations, completion time 304
 Database Directory 71
 Database password (parameter) 107
 Database read timeout 71
 Database read timeout (parameter) 69
 databases
 backing up 321, 322
 drivers, specifying location 92
 ESS Reporting 107
 recovery and 71
 Sleepycat's Berkeley DB 69
 utilities 318, 322, 324
 Database snapshot timeout (parameter) 69, 274
 Database user name (parameter) 108
 Debug (parameter)
 Flat Files 157
 LDAP and LDAP Test 226
 LiveAdapters 106
 Log 303
 Multicast 275
 Process Manager 295
 Debug database operations 71
 Debug file upload manager 72
 Default database post frequency
 (parameter) 108
 Default Expiration Time 65
 Default page size for paging support
 (parameter) 108
 Default Session Reconnect Time 65
 kn_journal topics and 65

Default URL on session login (parameter)
 Flat Files 157
 LDAP 226
 Delete (button)
 event management 137
 routes 144
 Delete All Events (command) 138
 Delete All Routes (command) 143
 Delete Property (button) 125
 event properties 139
 Delete Selected Item(s) (command), SSL 77
 Delete Selected Route (command) 144, 145
 Delete Topic (button) 127
 desktop application (LiveServer)
 changing to a service 32
 converting LiveServer service to 33
 Detach Filter (command) 132
 Detach Route Filter (command) 145, 148
 developer Web pages [xx](#)
 dictionaries, for event strings 72
 directories
 for KnowNow LDAP files 169
 /knrouter_db/ 71
 recovery data 71
 shared
 Linux 284
 Windows 2000 and older 283
 Windows 2003 284
 directory information trees (DIT) 164
 Directory Type (parameter) 226
 distinguished name
 for searching permission patterns 227
 LDAP 226
 LiveServer 227
 DIT (directory information trees) 164
 documentation
 accessing 52
 adding links to your manuals 47
 conventions [xix](#)
 KnowNow [xviii](#)
 APIs [xviii](#)
 LiveBrowser [xix](#)
 document.domain, cross-domain Web
 applications and 86

dollar sign (\$) in knsettings.conf [54](#)
 Domain (parameter)
 cross-domain Web applications and [86](#)
 JavaScript [86](#)
 configuring [88](#)
 service network [60](#)
 clusters and [62](#)
 SSL and [61](#), [75](#)
 domain names
 common suffixes [88](#)
 Microsoft Exchange server [109](#)
 do_method, using to manage topics [113](#)
 double slashes (/), LiveServer URL prefix
 and [89](#)
 durable, events and [126](#)

E

eDirectory [166](#)
 Edit (button), event properties [138](#)
 email
 SMTP, sending in response to events [63](#)
 Enable command processing time
 measurement (parameter) [303](#)
 Enable Debug [88](#)
 Enable Exchange Adapter (parameter) [108](#)
 Enable file uploads (certificates, licenses,
 etc) [72](#)
 Enable HTTP [61](#)
 Enable HTTPS [61](#)
 SSL and [75](#)
 Enable Java-based filters? [92](#)
 Enable Leak Check, filter module debugging
 and [98](#)
 Enable Leak Check (parameter) [289](#), [289](#), [303](#)
 Enable Moreover (parameter) [108](#)
 Enable Reporting Subsystem (parameter) [108](#)
 Enable Sphere Adapter (parameter) [108](#)
 Enable Stack Check (parameter) [289](#), [289](#), [303](#)
 Enable string dictionary [72](#)
 Enable thread stack size checking
 (parameter) [303](#)
 Encode passwords in knsettings.conf
 (parameter) [230](#)

Enterprise Syndication Solution (ESS)
 authentication policies and [155](#)
 configuring [104](#)
 parameters [107](#)
 containers
 AD [205](#)
 ADAM [180](#)
 AD with OpenLDAP [171](#)
 background information [164](#)
 dual OpenLDAP [189](#)
 standalone OpenLDAP [198](#)
 LDAP and [161](#)
 login expiration [111](#)
 login style [111](#)
 users, background information [164](#)
 errors
 tunnel, threshold [316](#)
 errors (alarms) [309](#)
 ESS Reporting
 database
 JDBC connection URL [110](#)
 JDBC driver name [110](#)
 post frequency [108](#)
 user name [108](#)
 user password [107](#)
 enabling or disabling [108](#)
 log level [110](#)
 partition period [111](#)
 RDBMS name [111](#)
 threads per processor [111](#)
 Event-driven Shutdown [297](#)
 Event ID (column) [133](#)
 Event Publisher [318](#), [320](#)
 events
 See also topics.
 acceptable size, setting [66](#)
 atomic mode [71](#)
 Atomic option and [126](#)
 clearing from topics [137](#)
 configuring access to [237](#)
 definition of term [5](#), [6](#), [329](#)
 deleting [137](#)
 using a browser [137](#)
 durable [126](#)

events (*continued*)

- event table 133, 134
- expiration times 65, 136
 - default 10
- expired, deletion interval 315
- how LiveServer publishes them 9
- identifiers 6
- idle, memory and 314, 315
- information, saving to disk 71
- journals and 65
- lifetime of 10
- LiveServer and 2
- managing 133–139
- number of for delivery 10
- properties
 - adding 139
 - deleting 138
 - editing 138
 - viewing 133
- publishing, using the Event Publisher 320
- queued, memory for 314
- queue size, setting maximum 126
- statistics 291
- strings, dictionary and 72
- testing 135
- time functions 6
- topics and 117
- viewing in topics 133

Events Delivered 291

Events mode. *See* [events](#).

Events Published 291

Events Routed 291

Events tab 117

Exchange adapter threads per processor (parameter) 108

Exchange Adapter UDP Port for receiving updates (parameter) 109

Exchange Server Domain Name (parameter) 109, 109

Exchange Server Password (parameter) 109

Exchange Server Port (parameter) 109

Exchange Server Protocol (parameter) 109

Exchange Server Public folder Base URI (parameter) 109

Exchange Server Sign-in Account (parameter) 109

Executable (field) 96

- Process Manager 296

expiration times

- default, setting 65

External shutdown timeout (parameter) 66

F

failover mechanism 266

fatal (alarms) 310

feedback, providing [xx](#)

feeds, certificate authority files and 82

Feed threads per processor (parameter) 110

Feedwatcher 112

- HTTP and 61

file descriptors, setting number of 39

files

- access.log 325
- access_patterns 241
- configuration. *See* [liveserver.tcl](#).
- configuring access 238
- crash dump
 - policies 314
 - specifying location for 314
- group. *See* [groups](#).
- license. *See* [licenses](#).
- LiveServer executable, stopping 32
- log 301
 - configuring 302
 - LiveServer startup and 325
 - viewing 306
- passwd. *See* [users](#).
- pattern_permissions 242
- perms. *See* [permissions](#).
- RFC2252 compliant 166
- statistics.xml 314

File transfer wait (parameter) 69, 275

filter management 128–132

See also [filter modules](#).

- assigning global filter properties 132
- attaching filters
 - to routes 146

- filter management (*continued*)
 - attaching filters
 - to topics 130
 - available filters 130
 - detaching filters
 - from routes 147
 - from topics 132
 - modules 95
 - parameters, editing 131
 - topics and 117
 - viewing filters 128
 - filter modules
 - See also* filter management.
 - clusters and 265
 - configuring 95
 - debugging, configuration options 98, 303
 - deleting 97
 - routes and 97
 - Java
 - enabling 92
 - support for 90
 - /kn_config and 97
 - /kn_statistics and 97
 - /kn_subtopics and 97
 - loading 95
 - restrictions 97
 - rules for 97
 - Tcl email filter 63
 - unloading 97
 - routes and 97
 - unregistering 97
 - Filter Name (field) 96, 131
 - Filter Parameter (column) 128
 - /filters, using 116
 - Filters mode. *See* filter management.
 - Filters tab 117
 - firewalls 325
 - Windows installation and 29
 - Flat Files
 - configuring 153
 - parameters 154
 - permission pattern names and 244
 - Users/Groups tab access and 260
 - FLEXlm 101
 - default port number 62, 101
 - flow control 266
 - folders
 - Microsoft Exchange server, public 109
 - Follow referrals during search (parameter) 230
 - formats
 - tunnel, simple and JavaScript 9
 - forms
 - login
 - Flat Files 158
 - LDAP 228
 - Forwarding Certificate (parameter) 77
 - Forwarding Key (parameter) 77
 - forward slashes (/). *See* slashes (/).
 - Forward using Service Network Certificate (parameter) 77
 - fully qualified names 60
 - entering 60
- G**
- gather_statistics 290
 - Generate alarm on assertion? (parameter) 310
 - Generate alarm on error? (parameter) 309
 - Generate alarm on fatal? (parameter) 310
 - Generate alarm on notice? (parameter) 309
 - Generate alarm on unrecognized log entries? (parameter) 309
 - Generate alarm on warning? (parameter) 309
 - genrsa utility 78
 - Get (access permission) 235
 - Group container DN (parameter) 230
 - group file. *See* groups.
 - Group member attribute (parameter) 230
 - Group member Of attribute (parameter) 230
 - Group objectClass (parameter) 231
 - groups 260
 - AD with ADAM 181
 - AD with OpenLDAP 173
 - default 252
 - required permissions 253
 - deleting 261
 - dual OpenLDAP 191
 - finding 261

groups (*continued*)

- KnowNow_Administrators 253
- KnowNow_Internal 253
- LDAP and 167
 - container DN 230
- LiveAdapters Admin 107
- LiveAdapters Business Analyst 107
- standalone AD 207
- standalone OpenLDAP 199
- users and 256, 261

H

headers

See also individual header names.

- definition of term 6
- event expiration time 10
- event identifiers 6
- journals 8
- kn_max_queue_size 126
- logging 303, 304
- non-ASCII data and 6
- routes 7, 8
- subtopics 7, 8
- time functions 6
- UTF-8 format and 6

/heartbeat, using 116

histogram, Sphere 112

Hostname (field), LDAP servers 224

Hostname (parameter)

- Multicast 272
 - allowed length 272
- Service Network 62
 - clusters and 62, 272, 273
 - SSL and 75

host names, Microsoft Exchange server 109

HTTP and HTTPS

- connections, log file for 325
- Connector initialization and 8
- enabling HTTP 61
- enabling HTTPS 61
- Exchange Server 109
- Feedwatcher and 61
- HTTP proxy server settings 63
- POST requests, setting maximum size 66

HTTP and HTTPS (*continued*)

- requiring HTTPS for logging in 60
- routes and 9
- HTTP credentials required (policy) 154
- HTTP Port 62
- HTTP Proxy Server Settings (parameter) 63
- HTTPS Port 63

I

IDs

- for events. *See events.*
- for users. *See users.*
- Sphere customer 111
- Sphere partners 112

i icon 52

index.html (LiveServer)

- editing 47
- upgrading or reinstalling LiveServer and 18

infinity (keyword) 53

Init arguments (field) 297

initial route population (IRP) 10

Init process (field) 297

Insert New Item (command), SSL 76

installation

- all KnowNow products 19
- in a cross-domain environment
 - example 88
 - LiveBrowser and 88
 - LiveServer URL prefix and 89
 - security and 85
- Linux 35–39
 - C/C++ libraries 35, 36
- order of installing LiveServer and other KnowNow products 19
- overview 18
- sequence 19
- uninstalling products 19
- Windows 22–29
 - as a service 25
 - changing LiveServer desktop applications to a service 32

installation (*continued*)
 Windows (*continued*)
 changing service to a desktop
 application 33
 firewalls and 29
 Internal shutdown timeout (parameter) 66
 internal system users 257
 kndaemon 258
 parameters 258
 Internal System Users (option) 257
 interval, recovery 71
 IP addresses
 LiveServer, obtaining 325
 IRP (initial route population) 10
 Italic text, meaning of in book xix

J

Java
 classes, adding 92
 classpath 92
 configuring 90
 filter modules, enabling 92
 JRE and JDK and 90
 module for 90
 parameters 92
 root path 93
 VM, options 93
 Java Development Kit (JDK), Java support
 and 90
 Java Runtime Environment (JRE), Java
 support and 90
 JavaScript
 domain name
 default value 60
 setting 86
 tunnels 9
 JavaScript Connector
 configuring 85
 Enable Debug and 88
 parameters 86
 tunnel formats and 9
 JDBC connection URL (parameter) 110
 JDBC Driver name (parameter) 110
 JDBC drivers 92

journalReconnectTimeout 65
 journals
 clusters and 265
 configuring access to 237
 Connectors and 8
 creating 8
 definition of term 8
 events and 67
 expiration times 65
 /kn_subtopics and /kn_routes and 8
 lifetime 65
 naming conventions 8
 parameters for 67
 statistics 291
 tunnels and 8
 Journal Topics 291

K

Keep detailed XML reaper statistics 314
 keys (for certificates). *See* [certificate authority \(CA\) files](#).
 key/value pairs. *See* [headers](#).
 keywords, units of measure 53
 kn_action, process shutdown and 297
 knadmin (user name) 252
 Active Directory password 253
 kn_atomic 126
 atomic mode and 71
 knAuthenticateUser 150
 kn_authorize, clusters and 265
 /kn_config, modules and 97
 kn_config.js 87, 88
 upgrading or reinstalling LiveServer
 and 18
 kn_config_server, cross-domain
 configuration 87, 88
 kndaemon (use name) 252
 kndaemon (user name)
 changing the name of 259
 clusters and 269
 password 258
 required permissions 253
 kn_default_kn_expires 65
 journals and 67

- kn_default_kn_expires (*continued*)
 - setting value of 65
- kndemo (user name) 253
- ~KnDummy (user name) 252
- kn_expires 10, 136
 - deleting events and 137
 - expiration times 65
- kn_history_ 10
- kn_id 6
 - deleting events and 137
 - process shutdown and 297
 - topic properties and 124
- KN imported classes 92
- kninternal (user name) 252
 - required permissions 253
- /kn_journal 8, 8
 - default expiration times 65
- KNLanguage Subscribers 291
- knlogd, enabling 305
- kn_max_queue_size 126
- knmcluster module. *See* Multicast.
- kn_owner, nologin users and 255
- KnowNow
 - contacting xx
 - documentation set xviii
 - Web pages xx
- KnowNow_Administrators (group) 253
- KnowNow Enterprise Syndication Solution.
 - See* Enterprise Syndication Solution (ESS).
- KnowNow_Internal (group) 253
- KnowNow LiveBrowser Developer's Guide xix
- KnowNow LiveServer Developer's Guide* xviii
- KnowNow System Administration
 - accessing 42, 46
 - overview 45–54
- kn_payload, deleting events and 137
- KN_PROC_MGMT 297
- kn_route_location, topic properties and 124
- /knrouter_db/ directory
 - configuring location 71
 - Linux and 17
 - moving 324
 - server shutdown times and 66
- knrouter_db_snapshot 72
- /kn_routes 7
 - configuring access to 238
 - default expiration times 65
 - journals and 8
- KN server classes 93
- knsettings.conf 57
 - special characters and 54
- /kn_statistics 290
 - modules and 97
- /kn_subtopics 7
 - default expiration times 65
 - journals and 8
 - modules and 97
- /kn_system
 - subtopics, using 116
- kn_temporary 316
- kn_temporary_expires 315
- kn_time_t 6
- KN Topic Modeler utility. *See* Topics tab.

L

- LAN Multicast. *See* Multicast.
- Last Action (field), Process Manager 300
- Last Exit Code (field), Process Manager 300
- Last Modified (column) 134
- LDAP 159–232
 - See also* permissions.
 - administration password 227
 - advanced configuration 229
 - attribute types 165
 - authentication methods 231
 - clusters and 161
 - credentials, bind time 230
 - debugging statements 226
 - directories 169
 - group members of a groups 230
 - groups, container DN 230
 - integrating ESS with 161–222
 - AD with ADAM. *See* Active Directory with ADAM.
 - AD with OpenLDAP. *See* Active Directory with OpenLDAP. xvii
 - authorization data 168

- LDAP (*continued*)
 - integrating ESS with 161–222
 - authorization schema 164
 - dual and standalone OpenLDAP. *See* [OpenLDAP](#).
 - Perl preparation 162
 - prerequisites 161
 - RFC2252-compliant files and 166
 - standalone AD. *See* [Active Directory \(AD\)](#).
 - task list 164
 - users and groups 167
 - LDAPS:// mode 225
 - LiveServer users and groups and 167
 - login identities, trusting 232
 - login URL 226
 - module, configuring 223–232
 - AD with ADAM 184
 - AD with OpenLDAP 174
 - dual OpenLDAP 192
 - overview 168
 - standalone AD 211
 - standalone OpenLDAP 201
 - object classes 165
 - overview 160
 - parameters 225
 - advanced 230
 - passwords, encoding 230
 - permission pattern names and 244
 - permission patterns, DN for 227
 - queries 231
 - realm 231
 - searches, following referrals 230
 - servers
 - adding 224
 - deleting 225
 - order of use 224
 - specifying type 226
 - users 254
 - container DN 232
 - logins 253
 - users and 250
 - Users/Groups tab access and 260
 - ldapadd (program) 171
 - OpenLDAP and 189
 - LDAP Administration DN (parameter) 226
 - LDAP Administration Password (parameter) 227
 - LDAP authentication method (parameter) 231
 - LDAPS:// certificate 225
 - LDAP Test. *See* [LDAP](#).
 - licenses 99–102
 - configuring 99
 - FLEXlm and 101
 - monitoring 102
 - parameters 100
 - Linux
 - certificate authority files and 82
 - installing LiveServer on. *See* [installation](#).
 - LDAP and 170, 188
 - LiveAdapters clusters 282
 - privileges 17
 - setting number of client connections 39
 - shared directories 284
 - Listener Threads (parameter) 105
 - List Management Debug (parameter) 110
 - List subgroups of a user or a group in LDAP queries (parameter) 231
 - LiveAdapter for Microsoft Exchange
 - See also* [Microsoft Exchange](#).
 - configuration parameters 107
 - enabling and disabling 108
 - log level 110
 - port setting 109
 - Public folder base URI 109
 - threads per processor 108
 - LiveAdapter for Reporting, log level 110
 - LiveAdapter for RSS 110
 - LiveAdapter for Sphere
 - enabling and disabling 108
 - log level 110
 - LiveAdapters
 - Business Method Management tab, configuring access to 107
 - clusters and 282
 - Linux 282
 - Windows 2000 and older 282

- LiveAdapters (*continued*)
 - clusters and (*continued*)
 - Windows 2003 284
 - configuration 104
 - for Microsoft Exchange. *See* [LiveAdapter for Microsoft Exchange](#).
 - for RSS, log level 110
 - for Sphere
 - enabling and disabling 108
 - log levels 110
 - overview 5
 - Setup tab, configuring access to 107
- LiveAdapters Admin (LiveServer group) 107
- LiveAdapters Business Analyst (LiveServer group) 107
- LiveBrowser
 - configuring for cross-domain installations 88
 - documentation [xix](#)
 - system requirements 17
 - testing 42
- LiveBrowser Subscribers (statistic) 291
- Liveness wait (parameter) 69, 275
 - WAN clusters and 275
- LiveServer Authoring DN (parameter) 227
- LiveServer Authoring Password 227
- LiveServer DN (parameter) 227
- LiveServer Password (parameter) 227
- LiveServer Rekey page 318
- LiveServers
 - changing desktop installation to a service 32
 - changing service to a desktop application 33
 - configuring for LiveAdapters 104
 - documentation [xviii](#)
 - feeds, certificates and 82
 - login prompt, specifying server for 63
 - overview 3
 - process, stopping 32
 - restarting 50
 - shutting down, knrouter_db and 66
 - starting up
 - database read timeout 71
- LiveServers (*continued*)
 - starting up (*continued*)
 - Linux 38
 - Windows 29
 - status page, accessing 42
 - stopping, Linux 38
 - stopping and restarting, Windows 31
 - system requirements 16
 - version number 291
- liveserver.tcl
 - See also* [configuration](#).
 - upgrades and 18
- LiveServer URL prefix 89
 - slashes and 89
- LiveServer Username (parameter) 231
- LiveServer Version 291
- load balancers
 - clusters and 62, 268, 273
 - host names and 62
- load balancing, clusters and 267
- Location of license file(s) or license manager (parameter) 101
- Log (panel) 302
- Log all request headers (parameter) 303
 - LiveAdapters 106
- Log debug topic locks? (parameter) 303
- /log/ directory 325
 - Linux and 17
- log files
 - access 304
 - configuring 302
 - enabling 305
 - LiveServer startup and 325
 - maximum to keep 304
 - memory statistics 304
 - overview 301
 - performance range 304, 304
 - request headers 303, 304
 - rolling 305
 - setting roll time 305
 - server.log. *See* [server.log](#).
 - topic locks 303
 - viewing 306
 - writing authorization decisions to 154, 226

logging out 46
 logins
 default URL to use 157
 LDAP 226
 forms, custom 154, 158, 228
 requiring HTTPS 60
 Login Style (parameter) 111
 Login user expire after (parameter) 111
 Log Level for Exchange Adapter
 (parameter) 110
 Log Level for LiveAdapter for Reporting
 (parameter) 110
 Log Level for LiveAdapter for RSS
 (parameter) 110
 Log Level for LiveAdapter for Sphere
 (parameter) 110
 Log memory statistics? (parameter) 289, 304
 Logout (link) 47
 Log request headers if forbidden
 (parameter) 304
 Logs (command) 306
 Log Size (bytes) 306
 LSadmin (user)
 AD
 creating 205
 editing privileges 169, 215
 removing from the Domain
 Admins 215
 verifying privileges 221
 ADAM, creating 180
 AD with OpenLDAP, creating 171
 dual OpenLDAP, creating 189
 standalone OpenLDAP, creating 198

M

Macrovision, FLEXIm 101
 Maintain Access Logs (parameter) 304
 Max Backup (parameter) 304
 Maximum Event Size 66
 Maximum response time allowed for any
 command processing (parameter) 304
 Maximum Session Reconnect Time 67
 journals and events and 65

Maximum Tunnel Queue Size 67
 Max Queue Size (field) 126
 Memory 313
 memory
 configuring 313
 for queued events 314
 optimizing with topic swapping 314, 315
 parameters 314
 virtual, threshold level 315
 Memory Cache Topic Period 314, 315
 Memory for Tunnel Buffers 314
 Memory Notice Threshold 315
 metabase URL, Moreover 111
 metadata subtopics 7
 Microsoft Active Directory. *See* [Active Directory \(AD\)](#).
 Microsoft Exchange
 See also [LiveAdapter for Microsoft Exchange](#).
 domain name 109
 host name 109
 passwords 109
 port 109
 protocol 109
 Public folder 109
 sign-in account 109
Microsoft Windows Server 2003 R2 Step-by-Step Guide to Deploying ADAM 179
 modules
 authentication/authorization. *See* [nsperm module](#).
 configuring 95
 filter. *See* [filter modules](#).
 Java 90
 knmcluster. *See* [Multicast](#).
 nsopenssl 61
 configuring 75
 nsperm. *See* [nsperm module](#).
 overview 4
 permissions, selecting 150
 recovery 69, 73
 Monitor 308, 309
 Moreover
 enabling and disabling 108
 metabase URL 111

Moreover Metabase URL (parameter) 111
 Multicast 271–277
 adding nodes 278
 configuring, recommended practices 268
 deleting nodes 281
 editing nodes 280
 initial configuration 271
 parameters 274
 segment address and port 273
 Multicast address (parameter) 275
 Multicast port (parameter) 276
 multi-tiered deployments, configuring 86

N

Name (field) 296
 Process Manager 299
 names
 distinguished
 for searching permission patterns 227
 LDAP 226
 LiveServer 227
 domain
 clusters and 265
 setting 60
 fully qualified, entering 60
 host
 clusters and 265
 setting 62
 LiveServer aliases 59
 user
 default LiveServer 252
 default LiveServer and LDAP 167
 kndaemon 269
 Name Topics 291
 networks, recovery and 266
 nodes. *See* clusters.
 non-ASCII data, passing into LiveServer 6
 notice (alarms) 309
 nsd.exe, stopping 32
 ns_info, Web site for information on 54
 nsopenssl module 61
 configuring 75
 nsperm module

See also permissions.
 refreshing security databases 318
 ns_sendmail 63
 NullIO (field) 297
 Number of processors (parameter) 67

O

off-host routes, SSL and 77
 off-host routes. *See* routes.
 Open Access (policy) 155
 Open Access URLs (parameter) 157, 228
 OpenLDAP
 dual integration 188–196
 administrative user 189
 authorization data 190
 authorization schema 190
 configuring LDAP module 192
 users 191
 validating OpenLDAP servers 188
 validation 196
 standalone integration 197–204
 administrative user 198
 authorization data 200
 authorization schema 199
 configuring for LDAP module 201
 users 199
 validating OpenLDAP server 197
 validation 204
 with AD. *See* Active Directory with OpenLDAP.
 OpenSSL utility 78
 Options for Java VM 93
 Ordering level (parameter) 276
 Other Subscribers 291

P

parameters
 See also individual parameter names.
 about the default values 57
 cross-domain, LiveBrowser 88
 timeout 275
 tuning 64

- parameters (*continued*)
 - units of measure 53
 - using Tcl expressions in 54
- Parameter Value (column), filter management 129
- partition period 111
- Partition Period for Reporting Database (parameter) 111
- Partition-safe (parameter) 277
- partner ID, Sphere 112
- passwd file. *See* users.
- Password (parameter), clusters and 269
- passwords
 - See also* users
 - LDAP, encoding 230
 - LDAP administrator 227
 - Microsoft Exchange server 109
- Pattern Details 241
- pattern_permissions 242
- payloads
 - maximum size, setting 66
 - units of measure 53
- Peak Deliveries 67
- Peak New Connections 68
- Peak Publications 68
- PEM format
 - certificates and 78
 - forwarding certificates and 77
- performance measurement and tuning 288
- Performance range for background operations (parameter) 288, 304
- Performance range for fast operations (parameter) 288, 304
- Perl
 - ADAM and 163
 - AD and 163
 - LDAP and 162
- Permission change poll interval (parameter) 231
- /permissions, using 116
- permissions 149–158
 - 2.0 Compatible Files, refreshing security databases 318
 - access patterns 233–247
 - creating 244
- permissions (*continued*)
 - access patterns (*continued*)
 - LDAP and 168
 - naming 244
 - order of precedence 240
 - shipped with the LiveServer 237
 - sorting 236
 - syntax 238
 - table 236
 - viewing and editing 241
 - default 57
 - Flat Files. *See* Flat Files.
 - for default users and groups 253
 - for LiveAdapters tabs 107
 - LDAP. *See* LDAP.
 - meaning of each 235
 - patterns
 - ADAM and 183
 - OpenLDAP 172, 190, 200
 - standalone AD and 210
 - selecting a permissions module 150
 - topics and 119
 - users 252
- perms file. *See* permissions.
- persistence. *See* recovery.
- Persistence WorkQueue Limit (parameter) 72
- PIDs, Process Manager and 300
- policies, crash dump files 314
- Port (field), LDAP servers 224
- ports
 - 27000 62
 - 443 109
 - 80 62
 - 8000 62
 - 8001 62, 274
 - 8443 63
 - changing 62
 - cluster. *See* clusters.
 - default for requests 42
 - default numbers 62
 - FLEXlm 62, 101
 - LiveAdapter for Microsoft Exchange 109
 - Microsoft Exchange server 109
- Post (access permission) 235

Post logs and alarms as events (parameter) 305
 posts, setting number of 68
 presence, configuring access to 238
 privileges, access. *See permissions.*
 processes 293

- adding to the Process Manager 294
- parameters for 296
- shutting down 297, 300
- starting and stopping 298
- starting or restarting 300

 Process ID (field), Process Manager 300
 Process Manager 293

- adding processes to 294
- configuring 295
- monitoring 298

 processors, feed threads 110
 properties

- event. *See events.*
- route. *See routes.*

 Properties mode. *See topic properties.*
 Properties tab 117
 properties table 120
 Property Editor 121
 Property Name (column) 135

- route properties 145

 Property Name (field), topic properties 122
 Property Value (field), topic properties 122
 protocol, Microsoft Exchange server 109
 Protocol (field) 224
 proxies

- reverse 60
 - Service Network parameters for 60

 proxy servers, HTTP 63
 publications

- definition of term 5
- scenarios 11, 12

 publish, definition of term 5
 Publish (access permission) 235
 Publishers (parameter) 68
 publishing. *See publications.*
 Purpose (field) 297

- Process Manager 300

Q

queues, memory for 314
 queue size

- setting 126
- Spread process 266

R

RDBMS Used (parameter) 111
 Realm (parameter) 63, 231
 reaper

- about 315
- advanced diagnostics 305
- performance
 - completion time 304
 - statistics.xml file and 314
- statistics and 290

 Reaper Interval 315
 Reaper Interval Minimum 315
 recovery 69–73, 266

- clusters and 69, 266
 - database directory 72
- data, directory for 71
- database read timeout 71
- disabling 69, 72
- enabling 69
- files, statistics and 290
- information, moving 324
- interval (time for clearing) 71
- modules 69
 - using custom 73
- parameters 71
 - See also individual parameter names.*
- set_topic_property and 71
- Sleepycat and 69
- wait time 275
- WAN clusters and 275

 Recovery script URL (parameter) 277
 Red Hat Linux. *See Linux.*
 reinstalling LiveServer 18
 Rekey Page, LiveServer 318
 Rekey utility 80
 release notes, accessing 47
 Reload (button) 51

Remove Selected Group 243
 Remove Selected User 243
 Reporting adapter threads per processor (parameter) 111
 reporting. *See* [ESS Reporting](#).
 requests
 POST, setting maximum size 66
 Request Threads (parameter) 105
 req utility 79
 Restart Cluster (button) 50
 Windows 32
 Restarts (field) 297
 Restart Server (button) 50
 Windows 32
 reverse proxies 60
 Service Network parameters for 60
 RFC2252-compliant files 166
 Roll Logs (parameter) 305
 Roll Log Time (parameter) 305
 root certificates. *See* [certificate authority \(CA\) files](#).
 Root Distinguished Name (parameter) 228
 Root of Java Home directory 93
 round robin cluster algorithms 267
 Routed From (field) 142
 Routed To (field) 142
 Routed To (list) 141
 route list table 141
 route properties. *See* [routes](#).
 routes 140–148
 attaching filters to 146
 creating 142
 definition of term 5, 9, 140
 deleting from topics
 deleting all 143
 deleting one 144
 destinations 9
 filter modules and 97
 HTTP and HTTPS protocols and 9
 list of 141
 off-host
 configuring access to 238
 SSL and 77

routes (*continued*)
 properties
 editing, adding, or deleting 145
 viewing 145
 statistics 292
 subscriptions and 9
 topics and 9, 117, 140
 units of measure 53
 URIs and 9
 viewing 140
 Routes mode. *See* [routes](#).
 Routes tab 117
 Route Topics 292
 RPM Package Manager (RPM) 35
 RSA private key
 generating 78
 location of 77
 RSS, LiveAdapter for 110

S

sales contact information xx
 Save Property (button) 124
 scalable connections 266
 scenarios for using LiveServer. *See* [configuration](#).
 Scheduler Threads (parameter) 105
 schema
 authorization
 AD with ADAM 181
 AD with OpenLDAP 172
 dual OpenLDAP 190
 standalone OpenLDAP 199
 searches
 groups, group class 231
 LDAP, referrals and 230
 users, user class 232
 secure sockets layer (SSL) 75–84
 certificate authorities, trusted 82
 certificates
 client-side 76
 deleting 77
 configuring 75
 Domain parameter and 61, 75

- secure sockets layer (SSL) (*continued*)
 - Enable HTTPS and [61, 75](#)
 - forwarding [76](#)
 - certificates [77](#)
 - keys [77](#)
 - module for [61](#)
 - off-host routes and [77](#)
 - parameters [60, 77](#)
 - security
 - authentication/authorization. *See* [authentication/authorization](#).
 - cross-domain installations and [85](#)
 - default setting [57](#)
 - groups. *See* [permissions](#).
 - modules
 - custom [95](#)
 - nsperm. *See* [nsperm module](#).
 - passwords. *See* [users](#).
 - permissions. *See* [permissions](#).
 - perms file. *See* [permissions](#).
 - security modules, cluster nodes and [264](#)
 - Segment Address (parameter) [273](#)
 - Segment Port (parameter) [273](#)
 - Send Test Event (button) [136](#)
 - Send Test Event (button and dialog box) [136](#)
 - Server Alias (Full URL prefix) [59](#)
 - Server Description (parameter) [63](#)
 - server.log [325](#)
 - processes and [297](#)
 - renaming [305](#)
 - writing authorization decisions to
 - Flat Files [154](#)
 - LDAP [226](#)
 - Server log file (base) name (parameter) [305](#)
 - service installation [25](#)
 - changing LiveServer desktop application to [32](#)
 - changing to a desktop application [33](#)
 - Service Interface (parameter), Multicast and [272](#)
 - Service Network [58](#)
 - parameters [60](#)
 - sessions. *See* [journals](#).
 - set_topic_property [65](#)
 - atomic topics and [71](#)
 - severity levels, alarms [309](#)
 - Shared Filesystem Path (parameter) [105](#)
 - Show server name in errors (parameter) [63](#)
 - Shutdown Timeout [297](#)
 - sign-in account, Microsoft Exchange server [109](#)
 - simple tunnels [9](#)
 - single-node systems, Service Network parameters for [60](#)
 - slashes (/)
 - LiveServer URL prefix and [89](#)
 - /name/ and /name conventions [xix](#)
 - Sleepycat's Berkeley DB [69](#)
 - utilities [318, 322](#)
 - recovery and [324](#)
 - SMTP email, sending [63](#)
 - SMTP Server (parameter) [63](#)
 - Snapshot Directory [72](#)
 - snapshots of databases [322](#)
 - special characters in knsettings.conf [54](#)
 - Sphere
 - configuration parameters [107](#)
 - customer ID [111](#)
 - find related articles URL [112](#)
 - LiveAdapter for [110](#)
 - enabling and disabling [108](#)
 - partner ID [112](#)
 - server URL [112](#)
 - threads per processor [111](#)
 - Sphere adapter threads per processor (parameter) [111](#)
 - Sphere Customer ID (parameter) [111](#)
 - Sphere Partner ID (parameter) [112](#)
 - Sphere Server URL (parameter) [112](#)
 - Sphere URL for Find Related Articles (parameter) [112](#)
 - Sphere URL for Search Histogram (parameter) [112](#)
- Spread process, queue size [266](#)
- square brackets. *See* [brackets](#).
- SSL (secure sockets layer). *See* [secure sockets layer \(SSL\)](#).

Stack Size 315
 Start KnowNow LiveServer Service 30
 Start LMGRD on LiveServers? 101
 statistics 290

- gathering 290
- logging 304

 statistic.xml 314
 Stop KnowNow LiveServer Service 31
 strings

- in events, dictionaries and 72

 subscribe, definition of term 5
 Subscribe (access permission) 235
 subscribers, number of 292
 Subscribers (parameter) 68
 subscribing. *See* [subscriptions](#).
 subscriptions

- definition of term 5
- routes and 9
- scenarios 11, 12

 subtopics. *See* [topics](#).
 syntax, in access patterns 238
 Syslog Port (parameter) 295
 system requirements 16

- KnowNow Web site 16

T

Tcl

- email filter 63
- expressions, in parameters 54
- Web site 54

 technical support [xx](#)
 Template for login form (parameter) 228
 Temporary Topic Expires 315
 Temporary Topic Recognize 316
 test events 135
 threads

- ESS Reporting 111
- LiveAdapter for Microsoft Exchange 108
- Sphere 111

 Threshold for feedwatchdog warnings
 (parameter) 112
 time, specifying in the KnowNow System
 Administration 53
 timeout parameters 275

times, specifying for rolling log files 305
 topic expires (field) 126
 Topic Modeler utility. *See* [Topics](#) tab.
 Topic Path (field) 131
 topic properties 117, 119–132

- See also* [topics](#).
- adding 124
- deleting 125
- deleting topics 127
- editing 124
- kn_id and 124
- kn_route_location and 124
- named topics 123
- overview 122
- properties table 120
- viewing 119

 topics 113–148

- See also* [events](#); [Topics](#) tab.
- anonymous users, expiration time 107
- as URIs 6
- atomic 71, 126
- attaching filters to 130
- clearing events from 137
- clusters and 265
- creating 7
- definition of term 5, 6, 332
- deleting 127
 - caution 127
- events and 133
- expanding and collapsing topic views 115
- expiration times 65
- filters. *See* [filter management](#).
- hierarchical structure 114
- hierarchy 7
- idle, memory and 314, 315
- information, saving to disk 71
- journals. *See* [journals](#).
- metadata subtopics 7
- modes 117
- named 123
- naming rules 7
- number of 292
- ownership, changing 255
- properties. *See* [topic properties](#).
- restricting access to 119

- topics (*continued*)
 - routes and 9, 140
 - selecting 115
 - slashes and (/) 89
 - statistics 291
 - subscribing to each other, example 12
 - subtopics
 - creating 125
 - metadata 7
 - statistics 292
 - temporary 315, 316
 - /who 316
 - test events and 135
 - viewing events in 133
 - viewing subtopics 115
 - /who and /what 57
 - topic space
 - definition of term 114
 - exploring 115
 - Topics tab
 - accessing 115
 - Filters mode. *See* [filter management](#).
 - Properties mode. *See* [topic properties](#).
 - Routes mode. *See* [routes](#).
 - Total Subscribers 292
 - Total Subtopics Topics 292
 - Total Topics 292
 - Trace configuration subsystem
 - (parameter) 305, 305
 - Trace LiveServer TCL Interface
 - (parameter) 305
 - Trace network (post & HTTP request)
 - subsystem (parameter) 305
 - Trace statistics-handling (parameter) 306
 - Trace TCL scripts (configuration and permissions) (parameter) 306
 - Trace tunnel (delivery) subsystem
 - (parameter) 306
 - trademark statements ii
 - trailing slashes, LiveServer URL prefix
 - parameter and 89
 - troubleshooting, startup 325
 - trusted certificates. *See* [certificate authority \(CA\) files](#).
 - Trust Login identity (parameter) 232
 - tuning parameters 64
 - Tunnel Error Tolerance 316
 - tunnel routes, definition of term 5
 - tunnels
 - Connectors and 8
 - error threshold 316
 - journals and 8
 - number of 8
 - statistics 291
 - types of 9
 - 2.0 Compatible Files, refreshing security databases 318
- U**
- ulimit 39
 - uninstalling LiveServer 33
 - uninstalling products 19
 - Uninstall KnowNow LiveServer 33
 - units of measure 53
 - unrecognized log entries (alarms) 309
 - Update Filter Parameters (command) 131
 - upgrading LiveServer 18
 - files that are overwritten during 18
 - URIs (uniform resource identifiers), *See* [URLs \(uniform resource locators\)](#).
 - URLs (uniform resource locators)
 - See also* [Web pages](#).
 - ESS Reporting database 110
 - for Microsoft Exchange Public folder 109
 - for Moreover 111
 - for Sphere 112
 - routes and 9
 - topics as URIs 6
 - URL root 333
 - usage scenarios 11–14
 - aggregation 12
 - basic 11
 - User (parameter), clusters and 269
 - User container DN (parameter) 232
 - User identifier attribute (parameter) 232
 - User objectClass (parameter) 232
 - users 250–259
 - adding, finding, and deleting 255, 256

users (*continued*)

- administrative
 - ADAM 180
 - AD with OpenLDAP 171
 - dual OpenLDAP 189
 - standalone AD 205
 - standalone OpenLDAP 198
- AD with ADAM 181
- AD with OpenLDAP 173
- anonymous
 - allowing 155
 - ESS 107
- credentials, LDAP server bind time 230
- default 252
 - required permissions 253
- dual OpenLDAP 191
- ESS, login expiration 111
- ESS Reporting database 107
- groups and 256, 261
- internal 257, 257
 - name 259
 - parameters 258
 - passwords 258
- LDAP and 167, 250
 - container DN 232
 - LDAP user types 254
 - trusting identity of 232
- login forms for
 - Flat Files 158
 - LDAP 228
- Microsoft Exchange server 109
- names, kndaemon 269
- nologin 255
- permissions and privileges. *See* [permissions](#).
- standalone AD 207
- standalone OpenLDAP 199
- Uses Server log 297
- UTF-8 format, headers and 6
- utilities
 - Backup LiveServer Database 321
 - Event Publisher 320
 - genrsa 78

utilities (*continued*)

- OpenSSL 78
- overview 318
- req 79
- Sleepycat's Berkeley DB 318, 322
 - recovery and 324
- Topic Modeler. *See* [Topics](#) tab.

V

Value (column)

- event properties 135
- route properties 145

values

- for parameters, specifying 53

verification of LiveServer 40

- Windows 325

version number of LiveServer 291

View Entire Log 307

virtual memory, threshold level 315

W

WAN Multicast. *See* [Multicast](#).

Web pages

- See also* [URLs \(uniform resource locators\)](#).
- KnowNow xx
- /what 57
- /what/chat, configuring access to 237
- /who 57, 316

wide-area network (WAN) clusters. *See* [Multicast](#).

Windows

- 2000 and older
 - LiveAdapters clusters 282
 - mapped directories 283
- installing LiveServer on. *See* [installation](#).
- removing the LiveServer service 33
- Services, using to start or start LiveServer 30
- verifying that LiveServer is running 325
- Windows 2003
 - LiveAdapters clusters 284
 - mapped directories 284

Workload 64
 parameters 65
write privileges, Linux 17



XML, statistics.xml file 314

