PATROL Console Server and RTserver
Getting Started

Supporting
PATROL Console Server 7.5.00
RTserver 6.6.00

February 14, 2005
Contacting BMC Software

You can access the BMC Software website at http://www.bmc.com. From this website, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

United States and Canada

<table>
<thead>
<tr>
<th>Address</th>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC SOFTWARE INC</td>
<td>713 918 8800</td>
<td>713 918 8000</td>
</tr>
<tr>
<td>2101 CITYWEST BLVD</td>
<td>or 800 841 2031</td>
<td>(01) 713 918 8000</td>
</tr>
<tr>
<td>HOUSTON TX 77042-2827</td>
<td></td>
<td></td>
</tr>
<tr>
<td>USA</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Outside United States and Canada

<table>
<thead>
<tr>
<th>Telephone</th>
<th>Fax</th>
</tr>
</thead>
<tbody>
<tr>
<td>(01) 713 918 8800</td>
<td>(01) 713 918 8000</td>
</tr>
</tbody>
</table>

Copyright 2005 BMC Software, Inc., as an unpublished work. All rights reserved.

BMC Software, the BMC Software logos, and all other BMC Software product or service names are registered trademarks or trademarks of BMC Software, Inc.

All other trademarks belong to their respective companies.

BMC Software considers information included in this documentation to be proprietary and confidential. Your use of this information is subject to the terms and conditions of the applicable End User License Agreement for the product and the proprietary and restricted rights notices included in this documentation.

Restricted rights legend

U.S. Government Restricted Rights to Computer Software. UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure of any data and computer software by the U.S. Government is subject to restrictions, as applicable, set forth in FAR Section 52.227-14, DFARS 252.227-7013, DFARS 252.227-7014, DFARS 252.227-7015, and DFARS 252.227-7025, as amended from time to time. Contractor/Manufacturer is BMC SOFTWARE INC, 2101 CITYWEST BLVD, HOUSTON TX 77042-2827, USA. Any contract notices should be sent to this address.
Customer support

You can obtain technical support by using the Support page on the BMC Software website or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, please see “Before Contacting BMC Software.”

Support website

You can obtain technical support from BMC Software 24 hours a day, 7 days a week at http://www.bmc.com/support_home. From this website, you can

- read overviews about support services and programs that BMC Software offers
- find the most current information about BMC Software products
- search a database for problems similar to yours and possible solutions
- order or download product documentation
- report a problem or ask a question
- subscribe to receive e-mail notices when new product versions are released
- find worldwide BMC Software support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

Support by telephone or e-mail

In the United States and Canada, if you need technical support and do not have access to the web, call 800 537 1813 or send an e-mail message to support@bmc.com. Outside the United States and Canada, contact your local support center for assistance.

Before contacting BMC Software

Before you contact BMC Software, have the following information available so that Customer Support can begin working on your problem immediately:

- product information
  - product name
  - product version (release number)
  - license number and password (trial or permanent)

- operating system and environment information
  - machine type
  - operating system type, version, and service pack or other maintenance level such as PUT or PTF
  - system hardware configuration
  - serial numbers
  - related software (database, application, and communication) including type, version, and service pack or maintenance level

- sequence of events leading to the problem

- commands and options that you used

- messages received (and the time and date that you received them)
  - product error messages
  - messages from the operating system, such as file system full
  - messages from related software
Contents

Chapter 1  Components and Capabilities  15
Features of the PATROL Console Server ................................................. 16
  Storing Management Profiles ......................................................... 16
  Storing KM-related Resources ....................................................... 17
  Processing Monitoring Requests ..................................................... 18
  Authenticating and Impersonating Users .......................................... 19
  Storing User Rights and Permissions .............................................. 19
Features of the RTserver ................................................................. 20
How the PATROL Console Server and RTserver Fit Into PATROL ............ 20
Related Documentation ................................................................. 23

Chapter 2  Installing the PATROL Console Server and RTserver  25
Determining Which Components to Install ......................................... 27
Determining Where to Install Components ......................................... 27
Verifying System Requirements ....................................................... 28
  Supported Platforms ................................................................. 28
  Supported AIX Combinations ....................................................... 30
  Hardware Requirements ............................................................. 30
Determining How to Install Components .......................................... 31
  About the Installation Utility ......................................................... 31
  About the Distribution Server ....................................................... 32
Required Information for Installing the PATROL Console Server .......... 33
  Installation Directory ................................................................. 33
  PATROL Console Server Account ............................................... 33
  Password for the Root Account (Unix Only) .................................... 35
  RTSERVERS Environment Variable ............................................... 35
  Security Information ................................................................. 36
Required Information for Installing the RTserver ................................ 37
  Installation Directory ................................................................. 37
  RTserver Port Number ............................................................... 37
  RTserver Account (Windows Only) ............................................... 37
Required Information for Installing Japanese Resource Files ................ 38
  Installation Directory ................................................................. 38
Upgrading Versus First-time Installation ......................................... 38
  Upgrading the PATROL Console Server ....................................... 39
  Upgrading the RTserver ............................................................ 40
PATROL Console Server and RTserver Installation Worksheets ............ 41
  Worksheet for Installing Both PATROL Console Server and RTserver ... 41
  Worksheet for Installing Only RTserver ......................................... 42
Installing PATROL Console Server and RTserver .......................... 43
Providing Root Access After Running the Installation Utility .......... 54
Uninstalling the PATROL Console Server and RTserver ............. 55
Required Information for Uninstalling the PATROL Console Server . 55
Required Information for Uninstalling the RTserver ................... 55

Chapter 3 Administering and Configuring the PATROL Console Server 57
Preparing to Use the PATROL Console Server .......................... 59
Preparing PATROL Agents for Use with the PATROL Console Server . 60
Enabling PATROL Agent Communications with the PATROL Console Server and RTserver .......................... 61
Configuring the PATROL Console Server for PAM Support on Unix . 63
Choosing a Service ID for the PATROL Console Server ............... 64
Starting and Stopping the PATROL Console Server on Windows ...... 64
Starting and Stopping the PATROL Console Server on Unix .......... 65
Command Line Arguments for the PATROL Console Server ........... 67
Setting Up Users and Groups on the PATROL Console Server ....... 70
Authenticating Users ......................................................... 70
Impersonating Users ......................................................... 71
User Accounts and Groups on the PATROL Console Server ........ 72
Managing Rights ............................................................. 73
Managing Permissions ....................................................... 75
Changing the PATROL Console Server Account or Password ....... 76
Changing the PATROL Console Server Account or Password
Using admincl ............................................................... 77
Manually Changing the PATROL Console Server Account or
Password on Windows ...................................................... 78
Manually Changing the PATROL Console Server Account or
Password on Unix ......................................................... 79
Controlling the RTserver Used ............................................ 80
Configuring PATROL Console Server Options ......................... 81
Editing the Startup Configuration File .................................. 82
Configuring Account Order for Authenticating to PATROL Agents . 83
Controlling Scanning for PATROL Agent Information at Startup .... 85
Configuring Audit Logs ..................................................... 87
Configuring Connection to a Single RTserver Cloud ................. 90
Configuring Connections to Multiple RTserver Clouds ............. 93
Supporting Different Security Levels on Multiple RTserver Clouds . 99
Configuring Management Profiles ....................................... 100
Preventing PATROL Console Server Overload ......................... 109
Collapsing All Top-Level Objects When Opening a Management Profile . 112
Limiting the Number of Computers Per Management Profile ....... 114
Configuring the Known Services Cache ................................ 115
Configuring the Number of Process Thread Pools ...................... 117
Configuring Protocol Compression ...................................... 118
Saving ACLs and Impersonation Files at Regular Intervals .......... 119
Chapter 5  Command Line Utilities With PATROL Console Server

About the Command Line Utilities .................................................. 176
General Command Line Utilities .................................................. 177
  admin_copy ........................................................................ 177
  admincli ............................................................................. 180
  confManager ......................................................................... 194
  sec_encrypt_p3x ................................................................. 198
  stopPatrolService ................................................................. 199
  hg_import ........................................................................ 202
  hg_export ........................................................................... 204

Command Line Utilities for Management Profiles .......................... 206
  lt_counters ........................................................................ 206
  lt_db_util .......................................................................... 216
  lt_import ........................................................................... 218
  lt_export ............................................................................ 222

Chapter 6  Configuring RTservers and Components That Use RTservers

Planning RTserver Cloud Architectures for PATROL Central Infrastructure . 229
Starting and Stopping the RTserver ................................................. 229
  Recommended Order for Starting and Stopping the RTserver .......... 229
Starting and Stopping the RTserver on Windows .............................. 230
Starting and Stopping the RTserver on Unix ................................... 231
Components That Use RTservers .................................................. 232
Logical Connection Names and RTserver Communications ............ 232
The rtserver.cm File .................................................................. 233
  Editing an rtserver.cm File ................................................... 233
  Example of an rtserver.cm File .............................................. 234
Configuring RTservers to Listen for Connections ............................ 236
  Logical Connection Names for Listening for Connections ............. 236
  Using Multiple Logical Connection Names for Listening for Connections 237
Configuring RTserver to Listen for Connections on Computers with Multiple IP Addresses ...................................................... 238
Configuring an RTserver Cloud .................................................... 239
Configuring RTservers for Standalone Mode ................................... 239
Configuring RTservers to Form a Cloud ........................................ 240
  Logical Connection Names for Forming RTserver Clouds .......... 240
  Using Multiple Logical Connection Names When Forming a Cloud 241
  Example: Forming an RTserver Cloud ................................... 241
Configuring RTclients to Connect to RTservers ................................ 243
  Logical Connection Names for Connecting to RTservers ............ 244
  Using Multiple Logical Connection Names to Connect to RTservers 244
Configuring the Number of Client Connections ............................... 245
Specifying RTserver Logging Options .......................................... 246
Configuring Advanced RTserver Options ..................................... 249
Figures

PATROL Console and RTserver in the PATROL Architecture – Single-cloud Configuration ................................................................. 21
PATROL Console and RTserver in the PATROL Architecture – Multi-cloud Configuration ............................................................... 22
Example of RTservers Forming a Cloud ............................................. 242
Example Firewall Setup on Port 2059 ................................................. 265
Example Firewall Setup on Port 2000 ................................................. 266
# Tables

- PATROL Console Server and RTserver Components ............................................. 27
- Platform Support for PATROL Console Server and RTserver ............................ 28
- Support Matrix for AIX 5.x Versions ................................................................. 30
- Basic Hardware Requirements for the PATROL Console Server ......................... 31
- PATROL Console Server Platforms Installable by the Distribution Server ............. 32
- PATROL Console Server Account Requirements ............................................... 34
- Required Tasks for Using PATROL Agents with PATROL Console Server .......... 60
- Command Line Arguments for the PATROL Console Server ............................. 67
- Default PATROL Groups on the PATROL Console Server .................................. 73
- Authentication Order for Connecting to PATROL Agents .................................. 83
- Audit Log Options in the PATROL Console Server Configuration File ................ 88
- Single-cloud Options in the PATROL Console Server Configuration File ............. 91
- Multiple Cloud Options in the PATROL Console Server Configuration File ......... 95
- Management Profile Options in the PATROL Console Server Configuration File ... 101
- Overload Protection Options in the PATROL Console Server Configuration File ... 109
- Management Profile Size Limit Options in the PATROL Console Server Configuration File ................................................................. 115
- KSC Options in the PATROL Console Server Configuration File ...................... 116
- Protocol Compression Option in the PATROL Console Server Configuration File 118
- PATROL Console Server Directories to Back Up for a Full Backup .................... 121
- PATROL Console Server Data and Configuration Files to Back Up ..................... 122
- Online Backup Options in the PATROL Console Server Configuration File ........... 127
- Fields for Specifying a Job Specification String ............................................... 129
- Valid and Invalid Job Specification String Values ........................................... 131
- PATROL Console Server Built-in Commands .................................................... 143
- Environment Variables for Moving PATROL Console Server Directories ............. 153
- Logical Connection Names for Listening for Connections ............................... 236
- Logical Connection Names for Forming RTserver Clouds ............................... 240
- Logical Connection Names for Connecting to RTservers .................................. 244
- RTserver Logging Options ................................................................................. 246
- Advanced RTserver Options ............................................................................. 249
- Options for Running Multiple Servers on the Same Computer ......................... 252
- Service Name Formats ...................................................................................... 269
- Locations for Diagnostic Information on Windows ............................................ 288
- Locations for Diagnostic Information on Unix ................................................. 289
- PATROL Console Server Environment Variables on Windows ......................... 294
- PATROL Console Server Environment Variables on Unix ............................... 295
- Directories and Files for PATROL Console Server on Windows ....................... 296
- Directories and Files for PATROL Console Server on Unix .............................. 301
- Soft Links for PATROL Console Server on Unix ............................................. 307
Components and Capabilities

This chapter provides an overview of the PATROL Console Server and RTserver and their roles in the PATROL architecture.

This chapter contains the following topics:

Features of the PATROL Console Server ................................................. 16
  Storing Management Profiles ......................................................... 16
  Storing KM-related Resources .......................................................... 17
  Processing Monitoring Requests ...................................................... 18
  Authenticating and Impersonating Users ........................................... 19
  Storing User Rights and Permissions ............................................... 19
Features of the RTserver ................................................................. 20
How the PATROL Console Server and RTserver Fit Into PATROL .......... 20
Related Documentation ................................................................. 23
Features of the PATROL Console Server

The PATROL 7.x architecture uses a three-tiered communications model. The PATROL Console Server is in the middle tier between the PATROL Agent and PATROL 7.x consoles.

The PATROL Console Server creates a bridge between PATROL Agents distributed over the enterprise and PATROL 7.x consoles. The PATROL 7.x consoles supported by this release of PATROL Console Server are the following:

- PATROL Central Operator – Microsoft Windows Edition
- PATROL Central Operator – Web Edition
- PATROL Central Administration – Microsoft Windows Edition
- PATROL Central Administration – Web Edition

All the data for the above consoles is collected and saved in data files on the PATROL Console Server. By storing information, such as management profiles, on the PATROL Console Server, users can access information from either edition of PATROL Central Operator and PATROL Central Administration.

The PATROL Console Server performs the following responsibilities:

- stores management profiles
- stores KM-related resources
- processes monitoring requests
- authenticates and impersonates users
- stores user rights and permissions

Storing Management Profiles

Management profiles are used in PATROL Central Operator to monitor the PATROL environment. These management profiles are stored on the PATROL Console Server so that they can be accessed from any instance of PATROL Central Operator on any computer.

Storing management profiles on the PATROL Console Server enables users to access them from multiple locations.

Users can open a management profile as read-only or for read/write. Multiple users can open the same management profile at the same time, if they all open the management profile as read-only.
By default, only the owner of the management profile, PATROL administrators, or users with the FORCE_PROFILE_CLOSE_PRIV right can open a management profile for read/write. Only one user can have the management profile open for read/write at a time. However, other users can still open the management profile as read-only at the same time.

If the owner of a management profile, a PATROL administrator, or a user with the FORCE_PROFILE_CLOSE_PRIV right attempts to open a management profile for read/write while it is open as read/write by someone else, PATROL Console Server will force it closed on the other user. The other user may then reopen it as read-only.

Management profiles contain a variety of data, including the following:

- which PATROL Agents to connect to and which KMs to load
- user preferences for charts, gauges, column order in list views, and view types such as icons, lists, and details
- user-defined charts, custom views, and folders

For a detailed list of what is stored in a management profile, see the PATROL Central Operator – Microsoft Windows Edition online Help or PATROL Central Operator – Web Edition online Help.

**Storing KM-related Resources**

The PATROL Console Server stores the following resources for Knowledge Modules (KMs):

- KM icons
- KM online Help files

These icons and Help files are accessed through PATROL Central Operator.
Processing Monitoring Requests

The PATROL Console Server performs the following tasks related to monitoring your PATROL environment:

- provides PATROL Central Operator with information in management profiles
- forwards requests for executing KM commands from PATROL Central Operator to PATROL Agents
- forwards system output from PATROL Agents to PATROL Central Operator
- executes managed system queries across multiple PATROL Agents
- executes event filters across multiple PATROL Agents
- executes state change actions as defined in management profiles

The PATROL Console Server also caches information used by PATROL Central Operator, such as:

- static agent attributes such as PATROL Agent versions, operating system names and versions, and preloaded KM lists
- KM menu and InfoBox commands

For more information about monitoring, see the PATROL Central Operator – Microsoft Windows Edition online Help and PATROL Central Operator – Web Edition online Help.
Authenticating and Impersonating Users

The PATROL Console Server performs the following tasks relating to user authentication and impersonation:

- provides a centralized location for all users to log on to access PATROL Agents
- stores console credentials and impersonation data for user accounts in one central location
- provides a centralized location where user logons and actions can be audited

Centralized user authentication and impersonation allows PATROL administrators to manage which users can access which PATROL Agents, without requiring users to log on to each PATROL Agent or storing passwords on individual consoles.

For more information about user authentication and impersonation, see “Setting Up Users and Groups on the PATROL Console Server” on page 70. For more information about administering and troubleshooting impersonation and authentication, see the PATROL Central Administration – Microsoft Windows Edition online Help or PATROL Central Operator – Web Edition online Help.

Storing User Rights and Permissions

The PATROL Console Server performs the following tasks relating to user rights and permissions:

- allows centralization of right assignments and access control definitions for both maintenance and enforcement purposes
- stores right information defined through PATROL Central Administration
- stores user permissions information (access control list definitions) as defined through PATROL Central Administration

For more information about user authorization, see “Setting Up Users and Groups on the PATROL Console Server” on page 70. For more information about administering and troubleshooting user authorization, see the PATROL Central Administration – Microsoft Windows Edition online Help or PATROL Central Operator – Web Edition online Help.
Features of the RTserver

The Real Time server (RTserver) delivers application-related data between the following PATROL components:

- PATROL Agent
- PATROL Central Operator
- PATROL Central Administration
- PATROL Console Server

The Distribution Server also uses the RTserver for its internal communications.

You can install more than one RTserver to form an RTserver cloud. RTserver clouds provide flexibility and load balancing.

You can also create multiple RTserver clouds, with each cloud linked to the PATROL Console Server. This configuration is recommended when the total number of PATROL Agents cannot be safely managed by a single RTserver cloud.

For more information about RTservers, see Chapter 6, “Configuring RTservers and Components That Use RTservers.”

How the PATROL Console Server and RTserver Fit Into PATROL

Figure 1 on page 1-21 and Figure 2 on page 1-22 show the relationships between the PATROL Console Server, an RTserver cloud, and other PATROL components.

When a PATROL 7.x console requests information about a managed system running a PATROL Agent, the request is first passed through the RTserver cloud to the PATROL Console Server. If the PATROL Console Server has the requested data, it sends the information back to the PATROL 7.x console through the RTserver cloud.

If the PATROL Console Server does not have the requested information, it passes the request through the RTserver cloud to the PATROL Agent. The PATROL Agent then sends the requested information back through the RTserver cloud to the PATROL Console Server, and the PATROL Console Server passes the information to the PATROL 7.x console through the RTserver cloud.
Figure 1  PATROL Console and RTserver in the PATROL Architecture – Single-cloud Configuration

Console Systems

Web browser

PATROL Central – Microsoft Windows Edition
- PATROL Central Operator
- PATROL Central Administration

RTserver Cloud

PATROL Central – Web Edition
- PATROL Central Operator
- PATROL Central Administration
- Other console modules

PATROL Console Server

Managed Systems

PATROL Agents
(version 3.5 or later)
Install PATROL solutions (KMs) on each system.

PATROL products and solutions may require additional files installed throughout the infrastructure.
Figure 2  PATROL Console and RTserver in the PATROL Architecture – Multi-cloud Configuration

Console Systems

- Web browser

Common Services

- PATROL Console Server
- RTserver Cloud

Managed Systems

- PATROL Central – Microsoft Windows Edition
  - PATROL Central Operator
  - PATROL Central Administration
  - Other console modules

- PATROL Central – Web Edition
  - PATROL Central Operator
  - PATROL Central Administration
  - Other console modules

PATROL products and solutions may require additional files installed throughout the infrastructure.
Related Documentation

The PATROL Console Server and RTserver are supported by the following documents:

- *PATROL Console Server and RTserver Getting Started* (this document)
- *PATROL Installation Reference Manual*
- *PATROL Security Release Notes*
- *PATROL Security User Guide*
- *PATROL Central Infrastructure Best Practices Guide*

For more information about PATROL components that use the PATROL Console Server and RTserver, see the following related documents:

- *PATROL Central Operator – Microsoft Windows Edition Getting Started*
- PATROL Central Operator – Microsoft Windows Edition online Help
- PATROL Central Administration – Microsoft Windows Edition online Help
- *PATROL Central Operator – Web Edition Getting Started*
- PATROL Central Operator – Web Edition online Help
- PATROL Central Administration – Web Edition online Help
- *PATROL Agent Reference Manual*
Installing the PATROL Console Server and RTserver

This chapter provides information that you need to install the PATROL Console Server and RTserver.

This chapter contains the following topics:

- Determining Which Components to Install ........................................... 27
- Determining Where to Install Components .......................................... 27
- Verifying System Requirements ................................................................. 28
  - Supported Platforms ............................................................................. 28
  - Supported AIX Combinations ................................................................. 30
  - Hardware Requirements ........................................................................ 30
- Determining How to Install Components ............................................... 31
  - About the Installation Utility ................................................................. 31
  - About the Distribution Server ................................................................ 32
- Required Information for Installing the PATROL Console Server ........ 33
  - Installation Directory ........................................................................... 33
  - PATROL Console Server Account ......................................................... 33
  - Password for the Root Account (Unix Only) ........................................ 35
  - RTSERVERS Environment Variable ......................................................... 35
  - Security Information .............................................................................. 36
- Required Information for Installing the RTserver .................................. 37
  - Installation Directory ........................................................................... 37
  - RTserver Port Number ........................................................................... 37
  - RTserver Account (Windows Only) ....................................................... 37
- Required Information for Installing Japanese Resource Files ............... 38
  - Installation Directory ........................................................................... 38
- Upgrading Versus First-time Installation ............................................... 38
  - Upgrading the PATROL Console Server ............................................. 39
  - Upgrading the RTserver ....................................................................... 40
- PATROL Console Server and RTserver Installation Worksheets ............ 41
  - Worksheet for Installing Both PATROL Console Server and RTserver . 41
  - Worksheet for Installing Only RTserver ............................................... 42
Installation Tasks ................................................................. 42
Upgrading the RTserver ......................................................... 42
Installing PATROL Console Server and RTserver ....................... 43
Providing Root Access After Running the Installation Utility ............... 54
Uninstalling the PATROL Console Server and RTserver ..................... 55
Required Information for Uninstalling the PATROL Console Server .......... 55
Required Information for Uninstalling the RTserver .......................... 55
Determining Which Components to Install

Table 1 lists the components that can be installed as part of the PATROL Console Server and RTserver.

**Table 1  PATROL Console Server and RTserver Components**

<table>
<thead>
<tr>
<th>Component</th>
<th>Comments</th>
</tr>
</thead>
</table>
| PATROL Console Server | This is the primary component for the PATROL Console Server. During installation, this component is called one of the following, depending on your platform:  
  - PATROL Console Server for Microsoft Windows  
  - PATROL Console Server for AIX  
  - PATROL Console Server for Linux  
  - PATROL Console Server for Solaris |
| RTserver       | During installation, this component is called one of the following, depending on your platform:  
  - RTserver for Microsoft Windows  
  - RTserver for AIX  
  - RTserver for Linux  
  - RTserver for Solaris |

Determining Where to Install Components

The overall size of your PATROL environment is the primary factor that determines where you install the PATROL Console Server and RTserver. It also determines how many PATROL Console Servers and RTservers you install. A small environment might use a single PATROL Console Server and RTserver on the same computer. A large or complex environment might require multiple RTservers, or even multiple PATROL Console Servers.

For more information about determining where to install components and how many you need of each, see the PATROL Central Infrastructure Best Practices Guide.
Verifying System Requirements

This section describes system requirements for installing the PATROL Console Server and RTserver.

Supported Platforms

Table 2 lists all the supported platforms and whether they are supported for PATROL Console Server, RTserver, or both. This table also lists the languages supported for each platform. Newly supported platforms are in bold and are identified with New.

Table 2  Platform Support for PATROL Console Server and RTserver (Part 1 of 2)

<table>
<thead>
<tr>
<th>Platform</th>
<th>Version or Service Pack Level</th>
<th>Languages</th>
<th>PATROL Console Server</th>
<th>RTserver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat Enterprise Linux</td>
<td>AS 2.1 (Intel x86) (formerly known as Advanced Server 2.1)</td>
<td>English and CTKJ</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>ES 2.1 (Intel x86)</td>
<td>English and CTKJ</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>AS 3.0 (Intel x86)</td>
<td>English and CTKJ</td>
<td>Yes</td>
<td>New</td>
</tr>
<tr>
<td></td>
<td>AS 3.0 (Itanium)</td>
<td>English and CTKJ</td>
<td>Yes</td>
<td>New</td>
</tr>
<tr>
<td></td>
<td>ES 3.0 (Intel x86)</td>
<td>English and CTKJ</td>
<td>Yes</td>
<td>New</td>
</tr>
<tr>
<td></td>
<td>ES 3.0 (Itanium)</td>
<td>English and CTKJ</td>
<td>Yes</td>
<td>New</td>
</tr>
<tr>
<td>AIX</td>
<td>5.1 (32 &amp; 64-bit)</td>
<td>English and CTKJ</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>5.2 (32 &amp; 64-bit)</td>
<td>English and CTKJ</td>
<td>Yes</td>
<td>New</td>
</tr>
<tr>
<td></td>
<td>5.3 (32 &amp; 64-bit)</td>
<td>English and CTKJ</td>
<td>New</td>
<td>New</td>
</tr>
<tr>
<td>Solaris</td>
<td>2.7 / 7 (32 &amp; 64-bit)</td>
<td>English and CTKJ</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>2.8 / 8 (32 &amp; 64-bit)</td>
<td>English and CTKJ</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>2.9 / 9 (32 &amp; 64-bit)</td>
<td>English and CTKJ</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Table 2  Platform Support for PATROL Console Server and RTserver (Part 2 of 2)

<table>
<thead>
<tr>
<th>Platform</th>
<th>Version or Service Pack Level</th>
<th>Languages</th>
<th>PATROL Console Server</th>
<th>RTserver</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows Server 2003</td>
<td>Enterprise Edition (Intel x86)</td>
<td>English and CTKJ</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Web Edition (Intel x86)</td>
<td>English and CTKJ</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Standard Edition (Intel x86)</td>
<td>English and CTKJ</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Datacenter Edition (Intel x86)</td>
<td>English and CTKJ</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Windows 2000</td>
<td>Advanced Server – SP2, SP3, or SP4</td>
<td>English and CTKJ</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Datacenter Server – SP2, SP3, or SP4</td>
<td>English and CTKJ</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Server – SP2, SP3, or SP4</td>
<td>English and CTKJ</td>
<td>Yes</td>
<td>Yes</td>
<td></td>
</tr>
</tbody>
</table>

1. CTKJ = Simplified Chinese, Traditional Chinese, Korean, and Japanese
2. For AIX 5.1, OS patch level 5100-05 or later is required.
3. For AIX 5.2, OS patch level 5200-04 or later is required.
4. For Solaris 2.8 32-bit, patch 108434-14 or later is required.
5. For Solaris 2.8 64-bit, patch 108435-14 or later is required.
6. If you install PATROL Console Server on a Windows 2000 Service Pack 2 system, the Q305227 hot patch must be applied. For information on obtaining the hot patch without upgrading service packs, see [http://support.microsoft.com](http://support.microsoft.com).

**NOTE**

The PATROL Console Server is installed as a native 64-bit application for the following platforms:

- Red Hat Enterprise Linux AS 3.0 (Itanium)
- Red Hat Enterprise Linux ES 3.0 (Itanium)
- AIX 5.1 64-bit
- AIX 5.2 64-bit
- AIX 5.3 64-bit
- Solaris 2.8
- Solaris 2.9
Supported AIX Combinations

PATROL Console Server and RTserver on AIX 5.x support the same 32/64 hardware, kernel, and ABI combinations as the PATROL Agent.

PATROL Console Server runs as a 64-bit application only in 64-bit kernel and 64-bit API mode. In all other modes, PATROL Console Server runs as a 32-bit application.

Table 3 lists the supported AIX combinations.

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Kernel</th>
<th>ABI</th>
</tr>
</thead>
<tbody>
<tr>
<td>32</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>32</td>
</tr>
<tr>
<td>64</td>
<td>32</td>
<td>64</td>
</tr>
<tr>
<td>64</td>
<td>64</td>
<td>64</td>
</tr>
</tbody>
</table>

Hardware Requirements

This section contains information about the hardware requirements for the PATROL Console Server and RTserver.

Hardware requirements for the PATROL Console Server and RTserver vary, depending upon the size of your environment. For more information on scalability considerations, see the PATROL Central Infrastructure Best Practices Guide.

This guide is available to further assist you with implementing the PATROL infrastructure components. This guide is available on the customer support pages for PATROL Central Operator – Microsoft Windows Edition 7.5.00 and PATROL Central Operator – Web Edition 7.1.10.

Table 4 on page 31 lists the basic PATROL Console Server hardware requirements for an environment consisting of 100 PATROL Agents, five PATROL consoles, one PATROL Console Server, and one RTserver. This table defines requirements for the PATROL Console Server only; no minimum requirements exist for the RTserver.

**NOTE**

The following requirements are based on testing in a specific environment. Your environment may have different requirements. For information about hardware recommendations for different environments, see the PATROL Central Infrastructure Best Practices Guide.
Determining How to Install Components

The components covered in this document were designed to be installed by using the BMC Software installation utility or the Distribution Server.

About the Installation Utility

The BMC Software installation utility runs in a Web browser. You can use the installation utility to perform a local installation or uninstallation. The installation utility includes the following features. For more information, see the PATROL Installation Reference Manual.

- You can install to remote computers in your environment by creating an installable product image that can be transferred to and installed locally on those computers.

- You can install to a computer that does not have a Web browser by launching the installation utility from a command line and specifying the \texttt{-serveronly} command line option. This option starts the installation Web server on the computer that does not have a browser, and you can then connect to that Web server by using a browser on another computer.

Details for installing products locally are included in this chapter. For details about creating, distributing, and installing installable images, see the PATROL Installation Reference Manual.

To get the most current information about possible installation issues, always refer to the latest version of the PATROL Console Server and RTserver Release Notes, the Installation Utility Reference Manual, and the Installation Utility Release Notes before using the installation utility.

### Table 4 Basic Hardware Requirements for the PATROL Console Server

<table>
<thead>
<tr>
<th>Resource</th>
<th>Minimum Requirement</th>
<th>Recommended Requirement</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor</td>
<td>Single processor, Intel Pentium III at 800 MHz (Linux and Windows)</td>
<td>Dual processor, Intel Pentium III at 800 MHz (Linux and Windows)</td>
</tr>
<tr>
<td></td>
<td>Single processor, Sun Sparc processor 300 MHz or higher (Solaris)</td>
<td>Dual processor, Sun Sparc processor 450 MHz or higher (Solaris)</td>
</tr>
<tr>
<td></td>
<td>Single processor, IBM pSeries POWER3 II at 450 MHz (AIX)</td>
<td>Dual processor, IBM pSeries POWER3 II at 375 MHz (AIX)</td>
</tr>
<tr>
<td>Server Memory</td>
<td>512 MB</td>
<td>1 GB</td>
</tr>
<tr>
<td>Hard Drive Space</td>
<td>300 MB</td>
<td>300 MB or more</td>
</tr>
</tbody>
</table>
About the Distribution Server

The Distribution Server is a BMC Software product for distributing products from a central server to multiple computers. For details about using the Distribution Server, see the Distribution Server Getting Started.

The PATROL Console Server and RTserver may be installed using Distribution Server 7.1.15 or 7.1.20, depending on the platform support provided by the Distribution Server.

Table 5 lists the PATROL Console Server platforms that may or may not be installed by the various versions of the Distribution Server. All RTservers are installable by the Distribution Server.

Table 5  PATROL Console Server Platforms Installable by the Distribution Server

<table>
<thead>
<tr>
<th>PATROL Console Server Platform</th>
<th>DS 7.1.15</th>
<th>DS 7.1.20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red Hat Linux 2.1, 3.0</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>AIX 5.1, 5.2</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>AIX 5.3</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Solaris 2.7, 2.8, 2.9</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows Server 2003</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows 2000</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

If you choose to use the Distribution Server, you will need the same product-specific information that is used for installing products locally with the installation utility. See “Required Information for Installing the PATROL Console Server” on page 33 and “Required Information for Installing the RTserver” on page 37.
You need to know the information in this section before installing the PATROL Console Server.

**Installation Directory**

The base installation directory is the location where you install all products that you select. Additional directories will be created under the base installation directory. This directory must meet the following requirements:

- The installation directory must be the same installation directory used by other BMC Software products, such as the PATROL Agent, on the same computer.

- On Unix, if the installation directory is on an NFS file system (instead of the local file system), the lockd daemon must be enabled.

- The installation directory cannot contain any multi-byte characters, such as those found in some non-English languages.

On Windows, the default installation directory is `C:\Program Files\BMC Software`. On Unix, the default installation directory is `/opt/bmc`.

You cannot change this directory after the installation.

**PATROL Console Server Account**

Before you install the PATROL Console Server, you must create an operating system account for the PATROL Console Server to use. Although you may select an existing user account, BMC Software recommends that you create a separate user account for the PATROL Console Server.

This account is referred to as the PATROL default account in the installation utility. When you install the PATROL Console Server, you must specify the user name and password of this account.

On Unix, you must also be logged on as this account when you run the installation utility.
Table 6 lists the Windows and Unix requirements for the PATROL Console Server account.

Table 6  PATROL Console Server Account Requirements

<table>
<thead>
<tr>
<th>Platform</th>
<th>Account Requirements</th>
</tr>
</thead>
</table>
| Windows  | This account must have the following user rights:  
- log on as a service  
- act as part of operating system  
- replace a process level token  
- log on locally (Windows 2000)  
- allow log on locally (Windows Server 2003)  

The installation program automatically assigns the listed user rights.  

This account can be a domain user account or a local account. However, if you want domain users to be able to access the PATROL Console Server, this account must be a domain account. The domain account does not need to be a Domain Admin or a privileged account at the domain level.  

The PATROL Console Server account no longer has to belong to the local administrators group. Previously, you could not start the PATROL Console Server service unless the account was a member of the local administrators group.  

Note: To install the PATROL Console Server, you must be logged onto the operating system with an account that has local administrator rights to perform the installation.  

Unix  
This account must have read, write, and execute permissions for the base installation directory.  

Do not use root for this account.|

The PATROL Console Server uses this account to open local PATROL Console Server files.  

This account also always has all rights and permissions for the PATROL Console Server. For more information about rights and permissions, see “Managing Rights” on page 73 and “Managing Permissions” on page 75.  

For information about changing this account after the installation, see “Changing the PATROL Console Server Account or Password” on page 76.
Password for the Root Account (Unix Only)

The installation program needs root access in order to install PATROL Console Server on Unix. You can provide root access in either of the following ways:

- run the `config_cserver.sh` script as root after running the installation utility
- specify the password for the root account when prompted for it by the installation utility

For more information, see “Providing Root Access After Running the Installation Utility” on page 54.

The installation program needs root access to do the following:

- run install-time scripts that require root permission
- set the owner of the PATROL Console Server executable file to root and grant the `setuid` permission for the executable

The PATROL Console Server executable requires root access for the following:

- to authenticate local users
- to spawn processes (such as action methods in management profiles) that are run as other accounts (such as the account for the management profile)

The root password is not stored or used after the installation.

RTSERVERS Environment Variable

The PATROL Console Server uses the RTSERVERS environment variable to connect to an RTserver.

The default value is `tcp:localhost:2059`, which tells the PATROL Console Server to connect to an RTserver on the same computer as the PATROL Console Server with the default port 2059.

If you install the PATROL Console Server at the same time as the RTserver, the port number in the variable should be the same as the port number specified for the RTserver. See “RTserver Port Number” on page 37.
If you want the PATROL Console Server to use an RTserver on a different computer or with a different port, you must modify the RTSERVERS environment variable with the correct host name and port number.

For more information, see Chapter 6, “Configuring RTservers and Components That Use RTservers.”

Security Information

You must set the level of security that you want to use. The recommended level is Basic Security, the default.

For more information, see the PATROL Security User Guide.

NOTE

If you plan on installing advanced security, BMC Software strongly recommends that you thoroughly review the contents of the PATROL Security User Guide for information about the configuration requirements of each security level.

Interoperability of Different Security Levels

PATROL Console Server can interoperate with PATROL Agents and PATROL Central consoles at security levels 2–4. However, there is one limitation. If a PATROL Agent has a security level of 4, then the PATROL Console Server also has to have a security level of 4.

The interoperability between security levels 2–4 applies to connections within a single RTserver cloud. For example, you can have PATROL Agents in the same RTserver cloud at different security levels 2–4 that communicate with a PATROL Console Server at security level 4.

The PATROL Console Server can also support separate security levels for each RTserver cloud in a multi-cloud configuration. This feature allows each RTserver cloud to have its own security level. For example, you can have one RTserver cloud at security level 0, another RTServer cloud at security level 1, and yet another RTserver cloud at a combination of security levels 2–4.

You can define and configure multiple RTserver clouds after you install the PATROL Console Server. (The installation uses a single-cloud configuration by default.)

For more information about configuring multiple RTserver clouds, see “Configuring Connections to Multiple RTserver Clouds” on page 93.
For more information about using per-cloud security levels, see “Supporting Different Security Levels on Multiple RTserver Clouds” on page 99.

For more information about security levels, see the PATROL Security User Guide.

Required Information for Installing the RTserver

You need to know the information in this section before installing the RTserver.

Installation Directory

This installation directory is the same as the one for the PATROL Console Server.

For more information, see “Installation Directory” on page 33.

RTserver Port Number

You specify the port number that the RTserver will use. The default port number is 2059.

If you install the PATROL Console Server at the same time as the RTserver, this port number should be the same as the port number in the RTSERVERS environment variable. See “RTSERVERS Environment Variable” on page 35.

For information about changing this port number after the installation, see “Changing the Port Number of an RTserver” on page 267.

RTserver Account (Windows Only)

In this release, the RTserver runs as the local system account. Therefore, you do not need a specific account when you install the RTserver. However, you must use an account with local administrative rights to run the installation program so that the RTserver can be installed as a Windows service.
Required Information for Installing Japanese Resource Files

The Japanese resource files provide message strings that have been localized for a Japanese environment. Install the Japanese resource files only if you are running in a Japanese environment.

You need to know the information in this section before installing the Japanese resource files.

Installation Directory

The installation directory is the same as the one for the PATROL Console Server.

For more information, see “Installation Directory” on page 33.

Upgrading Versus First-time Installation

In general, upgrading the PATROL Console Server or the RTserver has the same requirements and procedures as performing a first-time installation.

Before you upgrade, BMC Software strongly recommends that you create a full backup of your installation directory. Since the file format for several files have changed in this release, you will not be able to revert to a previous release unless you have a backup.
Upgrading the PATROL Console Server

For the PATROL Console Server, all data is preserved when you install the new version. If you choose to change the PATROL Console Server account when upgrading, ensure that the new account has the same file permissions as the previous account.

Although all of your data is preserved when you install the new version of PATROL Console Server, you may have to open and save the management profiles at least once before you can use the management profiles in read-only mode. Therefore, BMC Software strongly recommends that you back up your current installation before you install PATROL Console Server 7.5.00.

If the management profiles have not been opened and saved at least once, then users will not be able to open them in read-only mode. The following sections highlight the differences based on the version of PATROL Central Operator that is used to connect to PATROL Console Server 7.5.00.

**PATROL Central Operator 7.1.x and 7.2.x**

The internal file format used for the 7.5.00 management profiles has changed relative to the 7.2.x releases. After you upgrade to PATROL Console Server 7.5.00, you will have to convert each management profile to the new format before you can use the management profiles in read-only mode. To do this, open each profile in read/write mode first.

If a file is not converted and you try to open it in read-only mode, the open will fail and you will receive the following message:

*Cannot upgrade management profile ‘profileName’ on the fly when opened read-only. Please open as r/w first before opening read-only.*

**PATROL Central Operator 7.5.00**

PATROL Console Server 7.5.00 introduces the one-writer/N-reader feature that works with PATROL Central Operator 7.5.00. After you upgrade to PATROL Console Server 7.5.00, you will have to open and save each management profile in read/write mode in order to create a read-only copy that can be shared by multiple users.

If a management profile from an older version of PATROL Console Server has not been opened and saved at least once using PATROL Console Server 7.5.00, then the management profile will not be available as a read-only management profile for PATROL Central Operator 7.5.00 users.
Upgrading the RTserver

For the RTserver, the existing `rtserver.cm` file is automatically backed up when you install the new version. A new template file, `rtserver.tmpl`, is installed, and the existing `rtserver.cm` file is merged with the template file to create the new `rtserver.cm` file, preserving the existing configuration options.

Since logging is enabled in this release with the RTserver trace facility as an out-of-the-box setting, the `setopt trace_file` and `setopt time_format` options are overwritten. If you previously enabled these settings in your RTserver 6.2 installation, compare the new settings with your previous settings and manually update the settings that you want to keep. For more information about RTserver logging capabilities, see “Specifying RTserver Logging Options” on page 246.

The `setopt max_client_conns` option does not merge into the new `rtserver.cm` file. If you want additional client connections, you must manually edit this variable in the new `rtserver.cm` file. However, you can safely replace your 6.2.00 RTservers with 6.6.00 RTservers without any configuration changes, and you do not have to upgrade all of the RTservers in your RTserver cloud at one time.

RTserver 6.6.00 supports significantly more clients than RTserver 6.2.00. However, before configuring your environment to take advantage of the improved scalability and multi-cloud configurations, see the PATROL Central Infrastructure Best Practices Guide.

For more information about the `rtserver.cm` file, see Chapter 6, “Configuring RTservers and Components That Use RTservers.”
PATROL Console Server and RTserver Installation Worksheets

Use the following worksheets to record information for your PATROL Console Server and RTserver installations.

Worksheet for Installing Both PATROL Console Server and RTserver

Use this worksheet to record information for your PATROL Console Server and RTserver installations.

<table>
<thead>
<tr>
<th>General Information</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the computer name?</td>
<td></td>
</tr>
<tr>
<td>Where do you want to install BMC Software products?</td>
<td>The default is C:\Program Files\BMC Software (Windows) or /opt/bmc (Unix).</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>PATROL Console Server Information</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>What security level do you want to use?</td>
<td>Circle one:</td>
</tr>
<tr>
<td>The default is basic.</td>
<td></td>
</tr>
<tr>
<td>What are the user name and password for the PATROL Console Server account?</td>
<td></td>
</tr>
<tr>
<td>What is the Root password (Unix only)?</td>
<td></td>
</tr>
<tr>
<td>Overwrite the current security configuration (keys, certificates, and trusted roots)?</td>
<td>Circle one:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>RTserver Information</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>What is the port number for the RTserver to use?</td>
<td>The default is 2059.</td>
</tr>
<tr>
<td>Note that you specify this port number for both the PATROL Console Server and the RTserver.</td>
<td></td>
</tr>
</tbody>
</table>
Worksheet for Installing Only RTserver

Use this worksheet to record information for your RTserver installation.

<table>
<thead>
<tr>
<th>What is the computer name?</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Where do you want to install BMC Software products?</td>
<td></td>
</tr>
<tr>
<td>The default is <code>C:\Program Files\BMC Software</code> (Windows) or <code>/opt/bmc</code> (Unix).</td>
<td></td>
</tr>
<tr>
<td>What is the port number for the RTserver to use?</td>
<td>The default is 2059.</td>
</tr>
</tbody>
</table>

## Installation Tasks

This section contains the following tasks:

<table>
<thead>
<tr>
<th>Task</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upgrading the RTserver</td>
<td>page 42</td>
</tr>
<tr>
<td>Installing PATROL Console Server and RTserver</td>
<td>page 43</td>
</tr>
<tr>
<td>Providing Root Access After Running the Installation Utility</td>
<td>page 54</td>
</tr>
</tbody>
</table>

## Upgrading the RTserver

This task describes how to upgrade the RTserver if you enabled any of the trace-related options in your 6.2.0x installation.

In the RTserver 6.2.0x releases, the RTserver trace facility was not enabled in the default configuration. If you never added any of the trace-related configuration options—`trace_file`, `trace_level`, and `time_format`—then you can simply install the new version of RTserver. For more information about installing the RTserver, see “Installing PATROL Console Server and RTserver” on page 43.
To Upgrade the RTserver

1. Install the RTserver.

   For instructions, see “Installing PATROL Console Server and RTserver” on page 43.

2. Locate the backup of the `rtserver.cm` file that was created by the installation utility.

   The backed-up file is located in the following directory:

   - `%RTHOME%\standard\` (Windows)
   - `$RTHOME/standard/` (Unix)

   The file name is `rtserver.cm~x`, where `x` indicates the backup number. For example, after the first upgrade, the backup file name is `rtserver.cm~1`. After the second upgrade, the backup file name is `rtserver.cm~2`.

3. Compare the new settings with your previous settings. Copy the settings that you want to keep from the backed-up `rtserver.cm~x` file to the new `rtserver.cm` file.

   For more information about RTserver logging options, see “Specifying RTserver Logging Options” on page 246.

   For more information about configuring the RTserver and the `rtserver.cm` file, see Chapter 6, “Configuring RTservers and Components That Use RTservers.”

Installing PATROL Console Server and RTserver

This section describes how to install the PATROL Console Server and RTserver.

You can download the PATROL Console Server and the RTserver separately from the EPD site. If you do this, you can install the components separately. You can also combine the downloaded files into a single image and install all of the components at the same time.

**WARNING**

Never combine downloaded PATROL Console Server files with downloaded files from any other product besides RTserver.

This task assumes that you are performing a local installation of both the PATROL Console Server and RTserver at the same time.
For more information about how to run the installation utility, see “About the Installation Utility” on page 31.

If you install only the PATROL Console Server or only the RTserver, you might be asked for less configuration information. If you install other products or components at the same time, you will be asked for additional configuration information for them.

**Before You Begin**

The computer must meet the system requirements. For more information, see “Verifying System Requirements” on page 28.

You must know the required information for the installation you want to perform. For more information, see “Required Information for Installing the PATROL Console Server” on page 33 and “Required Information for Installing the RTserver” on page 37.

All PATROL-related applications and services, such as an existing PATROL Console Server or PATROL Agent, must be stopped.

On Windows 2000 and Windows Server 2003, the services MMC snap-in must be closed.

You must be logged on as the appropriate account:

- On Unix, you must be logged on as the PATROL Console Server account. For more information, see “PATROL Console Server Account” on page 33.

- On Windows, you must be logged onto the operating system with an account that has local administrator rights to perform the installation.

If you use pop-up blocker software to prevent pop-up windows from being displayed in your Web browser, you must temporarily disable the software on the computer on which you want to install PATROL Console Server and RTserver to run the installation utility. The procedures and requirements for disabling pop-up blocker software vary depending on the software that you are using. Consult the documentation provided with the pop-up blocker software for instructions.
To Install PATROL Console Server and RTserver

1. From the PATROL Central Operator – Microsoft Windows Edition and Infrastructure CD or from an electronically downloaded installation (EPD) image, start the installation utility.

   - On Windows, run setup.exe.
   - On Unix, run /setup.sh.

   **NOTE**
   For more information about starting the installation utility, including any optional command line arguments, see the PATROL Installation Reference Manual.

The Welcome to the Installation Utility page is displayed.
2 To start the installation, click Next.

The Review License Agreement page is displayed.

3 Review the license agreement. If you accept it, choose Accept. Then click Next.

The Select Installation Option page is displayed.
Choose **Install products on this computer now**. Then click **Next**.

**NOTE**
For more information about creating an installable image, see the *PATROL Installation Reference Manual*.

The Specify Installation Directory page is displayed.

Specify the location where you want to install BMC Software products. Then click **Next**.

**NOTE**
For more information about the installation directory, see “**Installation Directory**” on page 33.

The Select Products and Components to Install page is displayed.
6. Expand the PATROL Console Server and PATROL RTserver folders and select the following components:

- PATROL Console Server for platform name
- RTserver for platform name

For a list of the different component names, see Table 1 on page 27.

You can select individual components. For more information, see “Determining Which Components to Install” on page 27.

7. Click Next.

For Unix computers, the Provide the System Root Account Properties page is displayed. Continue with step 8 on page 49.

For Windows computers, the Provide the PATROL Default Account Properties page is displayed. Continue with step 9 on page 49.
8 Type the root password. Then click Next.

For more information, see “Password for the Root Account (Unix Only)” on page 35.

The Provide the PATROL Default Account Properties page is displayed.

9 Type the user name and password of the PATROL Console Server account. Then click Next.

For more information, see “PATROL Console Server Account” on page 33.

The Select Level of Security page is displayed.
10 Select security level that you want to use and whether you want to overwrite the current security configuration. Then click Next.

For more information, see “Security Information” on page 36.

The Provide Startup Information for the PATROL Console Server page is displayed.

11 Select whether you want to have the PATROL Console Server started automatically after the installation is complete. Then click Next.

If you selected to install the RTserver, the Provide Startup Information for the RTserver page is displayed. Continue with step 12 on page 51.

If you did not select to install the RTserver, the RTSERVERS Variable Properties page is displayed. Continue with step 13 on page 52.
12 Complete the Provide Startup Information for the RTserver page. Then click Next.

- Accept the default RTserver port number or type a different one. For more information, see “RTserver Port Number” on page 37.

**NOTE**

If you are upgrading the RTserver, you will not be prompted for a port number.

- Select whether you want to have the RTserver started automatically after the installation is complete.

If you selected to install the PATROL Console Server, the RTSERVERS Variable Properties page is displayed. Continue with step 13 on page 52.

If you did not select to install the PATROL Console Server, the Review Selections and Install page is displayed. Continue with step 14 on page 52.
13 Type the value for the RTSERVERS environment variable. Then click **Next**.

For more information, see “RTSERVERS Environment Variable” on page 35.

**NOTE**

If you changed the RTSERVERS environment variable after your last installation, you must type the correct value when prompted for the RTSERVERS environment variable during the upgrade. By default, the RTSERVERS environment variable that you specified during the last installation is displayed.

The Review Selections and Install page is displayed.

14 Review your selections. Then click **Start Install**.

The Installation Status page is displayed.
15 When the installation status reaches 100 percent, the Stop Install button changes to the Next button. Click Next.

The Installation Results page is displayed.

16 Click View Log to review the installation log file. Click Next, then click Finish to exit the installation utility.

The components are now installed on your system.
Providing Root Access After Running the Installation Utility

On Unix, if you install PATROL Console Server but do not specify the correct root password when running the installation utility, you must run the `config_cserver.sh` script after the installation.

**Before You Begin**

You must have installed the PATROL Console Server without the root password. For more information, see “Password for the Root Account (Unix Only)” on page 35.

You must know the following:

- the PATROL Console Server account
- the security level to be used
- whether you want to overwrite the existing security files

For more information about security, see the *PATROL Security User Guide*.

**To Run the config_cserver.sh Script**

1. Change to the `$BMC_ROOT/Patrol7` directory.
2. Set the user to root.
3. Enter the following command:

   ```bash
   ./config_cserver.sh defaultAccount securityLevel overwriteOption
   ```

   where the following table describes the parameters:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>defaultAccount</td>
<td>This is the account to be used for the PATROL Console Server. For more information, see “PATROL Console Server Account” on page 33.</td>
</tr>
<tr>
<td>securityLevel</td>
<td>This is the security level set in the security files in patrol.d. Possible values are 0, 1, 2, 3, or 4. For more information, see “Security Information” on page 36.</td>
</tr>
<tr>
<td>overwriteOption</td>
<td>This parameter specifies whether to overwrite the existing security files. Possible values are TRUE or FALSE. For more information, see “Security Information” on page 36.</td>
</tr>
</tbody>
</table>
Uninstalling the PATROL Console Server and RTserver

You can uninstall the PATROL Console Server and RTserver by running the installation utility in uninstall mode.

For detailed uninstallation instructions, see the PATROL Installation Reference Manual.

Required Information for Uninstalling the PATROL Console Server

You must know the following information prior to uninstalling the PATROL Console Server:

- the installation directory
- root password (Unix only)
- whether you want to delete the default PATROL groups (patadm, patpop, patop, patscadm, patwatch)
- whether you want to remove operating system user rights from the PATROL Console Server account (Windows only)

Required Information for Uninstalling the RTserver

Other than the installation directory, you do not need to know any other special information to uninstall the RTserver.
Administering and Configuring the PATROL Console Server

This chapter provides information about administering and configuring the PATROL Console Server.

This chapter contains the following topics:

Preparing to Use the PATROL Console Server ........................................... 59
   Preparing PATROL Agents for Use with the PATROL Console Server ........ 60
   Enabling PATROL Agent Communications with the PATROL Console Server and RTserver ......................................................... 61
   Configuring the PATROL Console Server for PAM Support on Unix ........ 63
   Choosing a Service ID for the PATROL Console Server ....................... 64
   Starting and Stopping the PATROL Console Server on Windows .......... 64
   Starting and Stopping the PATROL Console Server on Unix ................ 65
   Command Line Arguments for the PATROL Console Server ................. 67
Setting Up Users and Groups on the PATROL Console Server ................. 70
   Authenticating Users ................................................................. 70
   Impersonating Users .................................................................. 71
   User Accounts and Groups on the PATROL Console Server ............... 72
   Managing Rights ...................................................................... 73
   Managing Permissions .................................................................. 75
Changing the PATROL Console Server Account or Password .................. 76
   Changing the PATROL Console Server Account or Password Using admincli .......................................................... 77
   Manually Changing the PATROL Console Server Account or Password on Windows ......................................................... 78
   Manually Changing the PATROL Console Server Account or Password on Unix .......................................................... 79
Controlling the RTserver Used ............................................................... 80
Configuring PATROL Console Server Options ...................................... 81
   Editing the Startup Configuration File ......................................... 82
   Configuring Account Order for Authenticating to PATROL Agents ....... 83
   Controlling Scanning for PATROL Agent Information at Startup .......... 85
   Configuring Audit Logs .................................................................. 87
Preparation to Use the PATROL Console Server

This section includes information for preparing to use the PATROL Console Server. This section contains the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Preparing PATROL Agents for Use with the PATROL Console Server</td>
<td>page 60</td>
</tr>
<tr>
<td>Enabling PATROL Agent Communications with the PATROL Console Server and RTserver</td>
<td>page 61</td>
</tr>
<tr>
<td>Configuring the PATROL Console Server for PAM Support on Unix</td>
<td>page 63</td>
</tr>
<tr>
<td>Choosing a Service ID for the PATROL Console Server</td>
<td>page 64</td>
</tr>
<tr>
<td>Starting and Stopping the PATROL Console Server on Windows</td>
<td>page 64</td>
</tr>
<tr>
<td>Starting and Stopping the PATROL Console Server on Unix</td>
<td>page 65</td>
</tr>
<tr>
<td>Command Line Arguments for the PATROL Console Server</td>
<td>page 67</td>
</tr>
</tbody>
</table>
Preparing to Use the PATROL Console Server

Preparing PATROL Agents for Use with the PATROL Console Server

Depending on the version of the PATROL Agent, you might have to prepare it for use with the PATROL Console Server.

Table 7 lists the corresponding tasks for each version of the PATROL Agent.

<table>
<thead>
<tr>
<th>PATROL Agent Version</th>
<th>Required Configuration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Earlier than 3.5.00</td>
<td>The PATROL Console Server cannot communicate with these earlier versions of the PATROL Agent. If you want to use these versions with the PATROL Console Server, you must upgrade the PATROL Agent to at least version 3.5.00.</td>
</tr>
<tr>
<td>3.5.00</td>
<td>You must manually enable this version of the PATROL Agent to communicate with the PATROL Console Server and RTserver. See “Enabling PATROL Agent Communications with the PATROL Console Server and RTserver” on page 61. If these PATROL Agents have Operator ACLs, you must remove those ACLs, as they can cause connections from the PATROL Console Server to fail. For instructions, see the PATROL Agent Reference Manual.</td>
</tr>
<tr>
<td>3.5.01 to 3.5.10</td>
<td>You must manually enable these versions of the PATROL Agent to communicate with the PATROL Console Server and RTserver. See “Enabling PATROL Agent Communications with the PATROL Console Server and RTserver” on page 61.</td>
</tr>
<tr>
<td>3.5.30 or later</td>
<td>None.</td>
</tr>
</tbody>
</table>

You configure the PATROL Agent to communicate with the RTserver during the installation of the PATROL Agent. No additional configuration is required on the PATROL Agent, unless you want to change the RTserver.

PATROL Agent versions 3.5.32.08 or later support a configuration variable that can be used to assign an RTserver to a PATROL Agent. The name of the variable is /AgentSetup/rtServers. PATROL Configuration Manager (PCM) can be used to modify and distribute this variable.
Enabling PATROL Agent Communications with the PATROL Console Server and RTserver

The RTSERVERS environment variable must be set on the PATROL Agent computer for the PATROL Agent to communicate with the PATROL Console Server and RTserver. This task describes how to use the config_rtservers.psl script to set the RTSERVERS environment variable on PATROL Agent computers.

**TIP**

For PATROL Agent versions 3.5.32.08 or later, use the /AgentSetup/rtServers configuration variable to control the RTserver configuration of a PATROL Agent. PATROL Configuration Manager (PCM) can be used to define a rule for this variable that can be distributed to several PATROL Agents.

The config_rtservers.psl script can be used to change the RTSERVERS environment variable for any version of the PATROL Agent. However, it is only recommended for use with PATROL Agent versions 3.5.00 through 3.5.30.

**Before You Begin**

You must have access to PATROL Console for Windows 3.5.20 or later or PATROL Console for Unix 3.5.20 or later.

You must know the value that you want to use for the RTSERVERS environment variable. For more information, see “Configuring RTclients to Connect to RTservers” on page 243.

**NOTE**

The config_rtservers.psl script reinitializes the PATROL Agent.

**To Run the config_rtservers.psl Script**

1. Start PATROL Console for Unix or PATROL Console for Windows.

   You do not have to be in developer mode; however, the account used to connect to each PATROL Agent must be the same as the PATROL default account for the PATROL Agent or the PATROL_ADMIN account.

2. Select the PATROL Agents that require the update.
3 From the PATROL Main menu, choose Commands => All Computers => PSL Task.

The PSL Task window is displayed.

4 In the PSL Task window, click the Open File icon.

The Open File window is displayed.

5 In the Open File window, navigate to the following directory:

- %PATROL_HOME%\lib\psl (Windows)
- $PATROL_HOME/lib/psl (Unix)

6 Select the config_rtservers.psl script.

7 Click the Open button.

The script is displayed in the PSL Task window.

8 Edit the RTSERVERS= line of the script to reflect the desired RTservers.

The script is commented with directions and examples for you to follow. Edit only those items of note in the Customer Controlled Variables section of the script.

9 After completing edits to the config_rtservers.psl script, click the Apply to Selected button in the PSL Task window.

Each PATROL Agent task executes the script. The results are displayed in individual task output windows.

**NOTE**

After the RTSERVERS environment variable is created and updated for a PATROL Agent, the PATROL Agent reinitializes itself.
Configuring the PATROL Console Server for PAM Support on Unix

This task describes how to configure the PATROL Console Server for pluggable authentication module (PAM) support on Unix.

**NOTE**

PATROL Console Server supports PAM on AIX 5.2 or later, with certain patches. For information on the required patches, see the *PATROL Security User Guide*.

**To Configure the PATROL Console Server for PAM Support on Unix**

1. Locate the *site.plc* file in the `/etc/patrol.d/security_policy_v3.0` directory.

2. Edit the *site.plc* file to include the following line under the `[authenticator]` section, where `serviceName` specifies the PAM service to use:

   ```
   [authenticator]
   provider=pam
   service=serviceName
   ```

   On Linux, PAM services usually are configured in the `/etc/pam.d` directory. On AIX and Solaris, PAM services are configured in the `/etc/pam.conf` file.

   For more information about specifying authentication providers and services, see the *PATROL Security User Guide*.

3. Save and close the *site.plc* file.

4. If the PATROL Console Server is running, stop and restart it.
Choosing a Service ID for the PATROL Console Server

The service ID of the PATROL Console Server is how the PATROL Console Server is identified to the RTserver and other PATROL components, such as PATROL Central Operator.

By default, the service ID of the PATROL Console Server is the same as the host name of the computer.

However, you might want to use a different service ID for the PATROL Console Server for the following reasons:

- to make failover easier
- to make it easier to identify the purpose of the PATROL Console Server when there are multiple PATROL Console Servers

You set the service ID of the PATROL Console Server by using the `-id` command line argument. For more information about using this command line argument, see “Command Line Arguments for the PATROL Console Server” on page 67. For more information about service IDs, see “Choosing a Service ID” on page 269.

NOTE

Several of the files and directories used by the PATROL Console Server include the service ID of the PATROL Console Server. The PATROL Console Server uses its service ID to determine which files to use. If you change the service ID of the PATROL Console Server and want to continue using the same data files, see “Changing the Service ID of the PATROL Console Server” on page 136.

Starting and Stopping the PATROL Console Server on Windows

By default, the PATROL Console Server is started automatically as a service when you install it; however, you can start it manually. This section describes how to start and stop the PATROL Console Server and verify that it is running on Windows.

To Start or Stop the PATROL Console Server as a Service on Windows

Start or stop the PATROL Console Server service.

If you want to use any optional command line arguments, add them as startup parameters for the service. For more information about command line arguments, see “Command Line Arguments for the PATROL Console Server” on page 67.
To Start the PATROL Console Server From the Command Line on Windows

1. Open a command window.
2. Run `cserver.exe`, including any optional command line arguments.

   For more information about command line arguments, see “Command Line Arguments for the PATROL Console Server” on page 67.

To Stop the PATROL Console Server From the Command Line on Windows

Press Ctrl+C.

To Verify that the PATROL Console Server is Running on Windows

1. Press Ctrl+Alt+Delete.
2. Click the Task Manager button.
3. On the Processes tab, look for the `cserver.exe` process.

If a Problem Occurs

If the PATROL Console Server fails to start, see “PATROL Console Server Fails to Start on Windows” on page 273.

Starting and Stopping the PATROL Console Server on Unix

By default, the PATROL Console Server is started automatically when you install it; however, you can start it manually. This section describes how to start and stop the PATROL Console Server and verify that it is running on Unix.

To Manually Start the PATROL Console Server on Unix

1. Change to the `$PATROL_ROOT` directory.
2. Enter the following command, followed by any optional command line arguments:

   `./start_cserver.sh`
To verify that the PATROL Console Server is running on Unix

1. Enter the following command:
   ```bash
   ps -ef | grep cserver
   ```
   2. Look for the `cserver` process.

To manually stop the PATROL Console Server on Unix

1. Change to the `$PATROL_ROOT` directory.
2. Enter the following command:
   ```bash
   ./stop_cserver.sh
   ```

If a Problem Occurs

<table>
<thead>
<tr>
<th>Problem</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATROL Console Server Fails to Start on Unix</td>
<td>page 274</td>
</tr>
<tr>
<td>PATROL Console Server on an NFS File System Fails to Start on Unix</td>
<td>page 274</td>
</tr>
<tr>
<td>On Unix, PATROL Console Server Stops After You Log Out</td>
<td>page 275</td>
</tr>
</tbody>
</table>
Command Line Arguments for the PATROL Console Server

Table 8 lists the command line arguments for the PATROL Console Server.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-debug</td>
<td>Enables debug messages in the PATROL Console Server log file</td>
</tr>
<tr>
<td>-detach</td>
<td><em>(optional)</em> Unix only: The -detach option runs the PATROL Console Server as a detached, non-interactive daemon</td>
</tr>
<tr>
<td></td>
<td>You do not have to specify -detach. This option is used by default if no option is specified.</td>
</tr>
<tr>
<td>-nodetach</td>
<td><em>(optional)</em> Unix only: The -nodetach option runs the PATROL Console Server as an interactive process, such that the PATROL Console Server is stopped if the terminal that launched the PATROL Console Server is closed</td>
</tr>
<tr>
<td></td>
<td>If this option is not specified, the -detach option is used by default.</td>
</tr>
<tr>
<td>-help or -h</td>
<td>Displays a list of command line arguments</td>
</tr>
<tr>
<td>-id serviceId</td>
<td>Specifies the service ID used by the PATROL Console Server to identify itself to the RTserver and other PATROL components such as PATROL Central Operator</td>
</tr>
<tr>
<td></td>
<td>The specified service ID string is converted to uppercase.</td>
</tr>
<tr>
<td></td>
<td>For more information, see “Choosing a Service ID for the PATROL Console Server” on page 64 and “Components That Use RTservers” on page 232.</td>
</tr>
<tr>
<td></td>
<td>To save the service ID in the Windows registry, use -install -id serviceId.</td>
</tr>
</tbody>
</table>
Preparing to Use the PATROL Console Server

Table 8 Command Line Arguments for the PATROL Console Server (Part 2 of 3)

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
</table>
| -install [-name windowsServiceName] [-id serviceId] [-display displayName] [-displayDesc description] [-manual] [user [password]] | Windows only: Installs the PATROL Console Server as a service, or changes an existing PATROL Console Server service. 

windowsServiceName is the name used to register the service with the Windows Service Control Manager (SCM). The default is cserver.

If the windowsServiceName is not cserver and -id is not specified, then the windowsServiceName is also used as the RTserver service ID. For example, cserver -install -name my_cserver registers itself as a Windows service named my_cserver, and when the PATROL Console Server starts, it advertises itself in the RTserver cloud as PATROL_CSERVER_MY_CSERVER.

serviceId is the PATROL Console Server service ID saved in the Windows registry. If -id is not specified, the default depends on the -name argument. If -name is not specified, the default PATROL Console Server service ID is windowsServiceName.

For example, cserver -install -name my_cserver -id gopher registers itself as a Windows service named my_cserver, but when the PATROL Console Server is started, it advertises itself in the RTserver cloud as PATROL_CSERVER_GOPHER.

The serviceId specified in the Windows registry is not used when:

- you start the PATROL Console Server from the command line
- you manually specify the -id command line argument when you start the PATROL Console Server as a service

displayName is the string displayed for the service in the Windows SCM. If -display is not specified, then the default value for displayName depends on the -name argument. If -name is not specified, or is specified as cserver, then the default display name is PATROL Console Server. If -name is specified as something other than cserver, then the default display name is PATROL Console Server (windowsServiceName).

description is the descriptive string registered with the service.

-manual indicates that the service should be installed with a startup type of manual. The default startup type is automatic.

user and password specify the user ID and encrypted password for the PATROL Console Server service. The default is the local system account. To encrypt the password, you can use the sec_encrypt_p3x utility. For more information about the sec_encrypt_p3x utility, see “sec_encrypt_p3x” on page 198.
Preparing to Use the PATROL Console Server

Chapter 3 Administering and Configuring the PATROL Console Server

Table 8 Command Line Arguments for the PATROL Console Server (Part 3 of 3)

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-nostreamlog</td>
<td>Disables log messages to stderr Normally, the PATROL Console Server echoes log messages to stderr if it is run in the foreground from an OS command prompt. Specifying this option disables the echo of those messages to stderr. The messages are always recorded in the log file. This argument has no effect if the -detach argument is used.</td>
</tr>
<tr>
<td>-remove [-name windowsServiceName]</td>
<td>Windows only: Same as -uninstall If you use this argument on Windows 2000 or Windows Server 2003, ensure the Services MMC snap-in is closed. If it is not closed, then the service is not removed until the computer is rebooted. If a name is not specified, then cserver is used as the default.</td>
</tr>
<tr>
<td>-rtserver RTserverNames</td>
<td>Defines the list of RTserver names used by the PATROL Console Server to locate an RTserver The value of this argument overrides the value specified by the RTSERVERS environment variable. For more information about the RTSERVERS environment and the format of the list of RTserver names, see Chapter 6, “Configuring RTservers and Components That Use RTservers.”</td>
</tr>
<tr>
<td>-uninstall [-name windowsServiceName]</td>
<td>Windows only: Removes the PATROL Console Server as a service If you use this argument on Windows 2000 or Windows Server 2003, ensure the Services MMC snap-in is closed. If it is not closed, then the service is not removed until the computer is rebooted. If a name is not specified, then cserver is used as the default.</td>
</tr>
<tr>
<td>-v</td>
<td>Displays the PATROL Console Server version number and information about the build tag, date, and platform</td>
</tr>
</tbody>
</table>
Setting Up Users and Groups on the PATROL Console Server

The PATROL 7.x architecture uses operating system groups and users on the PATROL Console Server for the following:

- establishing the identity of users throughout PATROL (authentication and impersonation)
- controlling which functionality users can access (rights) and which objects users can access (permissions)

This section provides a brief overview of these features. For more information and specific tasks relating to these features, see the following documents:

- PATROL Central Administration – Microsoft Windows Edition online Help
- PATROL Central Administration – Web Edition online Help
- PATROL Central Operator – Microsoft Windows Edition online Help
- PATROL Central Operator – Web Edition online Help
- PATROL Central Operator – Microsoft Windows Edition Getting Started
- PATROL Central Operator – Web Edition Getting Started

Authenticating Users

Authentication is the process of establishing a user’s identity. Anyone who wants to use a PATROL Console Server, including users of PATROL Central Operator and PATROL Central Administration, must be authenticated as valid users by the PATROL Console Server.

The PATROL Console Server can authenticate only users with valid operating system accounts.

On Windows, if the PATROL Console Server account is a local account, the PATROL Console Server can authenticate only local accounts. If it is a domain account, the PATROL Console Server can authenticate both local and domain accounts.
Impersonating Users

Other programs in the PATROL architecture, such as the PATROL Agent, also allow access to only authenticated users. The PATROL Console Server facilitates authenticating users on PATROL Agents with a process called impersonation.

Impersonation is the process of allowing PATROL 7.x console users to access PATROL Agents that do not recognize the user’s account. The PATROL Console Server performs impersonation services by providing PATROL Agents with account information on behalf of the console user. If a PATROL Agent does not recognize the original user account, the PATROL Console Server consults its impersonation table that maps users to alias accounts that are recognized by PATROL Agents. The PATROL Console Server then provides the PATROL Agent with the alias account that is configured for the user.

For example, the authentication process for a user of PATROL Central Operator is as follows:

1. First, the user logs on to the PATROL Console Server through PATROL Central Operator, providing an account known to the PATROL Console Server. The PATROL Console Server authenticates this account. This account is the console account.

2. The PATROL Console Server provides the PATROL Agent with the console user’s account. If the PATROL Agent recognizes the console user’s account, the authentication process is successfully completed.

3. If the PATROL Agent does not recognize the console account, the PATROL Console Server consults its impersonation table for an alias account for the user or the user group to which the user belongs. The impersonation table lookup is based on the PATROL Agent’s name and managed system group membership.

4. If there is an alias account for the user or managed system group to which the user belongs, the PATROL Console Server provides it to the PATROL Agent. If the PATROL Agent recognizes the alias account, the authentication process is successfully completed.

5. If there is no alias account, or if the PATROL Agent does not recognize the alias account, the user is prompted for an account to use.

6. If the PATROL Agent recognizes the account the user enters, the authentication process is successfully completed. Otherwise, the user cannot access the PATROL Agent.
The process outlined illustrates the default impersonation process. You can configure the order in which accounts are selected for impersonation. For more information, see “Configuring Account Order for Authenticating to PATROL Agents” on page 83.

You manage the impersonation table for a PATROL Console Server through PATROL Central Administration or through the admincli command line utility. For information about creating, editing, and modifying impersonation tables, see the PATROL Central Administration – Microsoft Windows Edition online Help or the PATROL Central Administration – Web Edition online Help. For more information about the admincli command line utility, see “admincli” on page 180.

User Accounts and Groups on the PATROL Console Server

NOTE

In previous releases of PATROL Central, the term “privileges” was used to refer to controlling access to particular functions or operations, and the term “rights” was used to refer to controlling access to particular objects. In the 7.5.00 release of PATROL Central, this terminology was updated to be more in line with other BMC products. Now, the term “rights” is used to refer to controlling access to functions or operations, and the term “permissions” is used to refer to controlling access to particular objects.

The PATROL 7 architecture uses operating system user accounts and groups on the PATROL Console Server to control rights and permissions for PATROL. A right allows a user to access specific functionality. A permission allows a user to access specific PATROL objects. A PATROL user cannot access functionality or objects without the appropriate rights and permissions.

Users inherit the rights and permissions of the groups to which they belong. Usually it is easier to add users to the appropriate groups in the operating system than to administer rights and permissions directly for each user.

When you install the PATROL Console Server, default PATROL groups are created, if they do not already exist. When the user connects to the PATROL Console Server from a console, the user logs on with an operating system account. The PATROL Console Server uses the operating system account to identify the user, to which groups the user belongs, and which PATROL rights and permissions the user has.

Setting Up Users and Groups on the PATROL Console Server
Table 9 lists the default PATROL groups with pre-defined rights and permissions.

Table 9  Default PATROL Groups on the PATROL Console Server

<table>
<thead>
<tr>
<th>Group</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>patop</td>
<td>ordinary PATROL operators</td>
</tr>
<tr>
<td>patpop</td>
<td>PATROL power operators</td>
</tr>
<tr>
<td>patwatch</td>
<td>PATROL watchers</td>
</tr>
<tr>
<td></td>
<td>By default, members of this group have permission</td>
</tr>
<tr>
<td></td>
<td>only to watch console objects</td>
</tr>
<tr>
<td>patadm</td>
<td>PATROL administrators</td>
</tr>
<tr>
<td>patscadm</td>
<td>PATROL security administrators</td>
</tr>
</tbody>
</table>

You set up user accounts and groups in the operating system of the PATROL Console Server. You change rights and permissions of groups or individual users with PATROL Central Administration.

Managing Rights

**NOTE**

In previous releases of PATROL Central, the term “privileges” was used to refer to controlling access to particular functions or operations, and the term “rights” was used to refer to controlling access to particular objects. In the 7.5.00 release of PATROL Central, this terminology was updated to be more in line with other BMC products. Now, the term “rights” is used to refer to controlling access to functions or operations, and the term “permissions” is used to refer to controlling access to particular objects.

The PATROL Console Server uses rights to control access to functionality. PATROL consoles use these rights to control their graphical user interfaces, such as controlling the functioning of graphical use interface (GUI) objects. Regardless of whether the user interface is restricted, the PATROL Console Server enforces these rights for all appropriate actions generated by the console.

Some of the rights cover functionality provided by the PATROL Console Server and accessed only through PATROL Central Operator consoles, such as functionality involving management profiles. These rights are checked by the PATROL Console Server and cannot be bypassed.
Other rights cover functionality provided by the PATROL Agent. For example, if a user does not have the CLOSE EVENT right, the PATROL Console Server does not forward a request to close an event from PATROL Central Operator to the PATROL Agent. Having the PATROL Console Server check these rights allows a centralized way of managing them for PATROL Central Operator consoles. However, the PATROL Agent itself does not check these rights on the PATROL Console Server. Therefore, the same user can perform the operation from PATROL Console for Windows 3.5, PATROL Console for Unix 3.5, or PEM API. To prevent users of these earlier consoles or the PEM API from performing these operations, you must modify the configuration file for each PATROL Agent. For more information, see the \textit{PATROL Agent Reference Manual}.

The only rights used by the PATROL Console Server itself are the LOGON and SHUTDOWN rights. By default, members of the patop, patpop, patwatch, patadm, and patscadm groups have the LOGON right; members of patadm also have the SHUTDOWN right.

For a list of specific rights used by PATROL Central Operator or PATROL Central Administration, see their respective Help files.

You manage the right assignments for a PATROL Console Server through PATROL Central Administration or through the admincli command line utility. For information about assigning rights to users or groups, see the PATROL Central Administration – Microsoft Windows Edition online Help or the PATROL Central Administration – Web Edition online Help. For more information about the admincli command line utility, see “admincli” on page 180.

\textbf{NOTE}

The PATROL Console Server account (see “PATROL Console Server Account” on page 33) always has all rights.
Managing Permissions

**NOTE**

In previous releases of PATROL Central, the term “privileges” was used to refer to controlling access to particular functions or operations, and the term “rights” was used to refer to controlling access to particular objects. In the 7.5.00 release of PATROL Central, this terminology was updated to be more in line with other BMC products. Now, the term “rights” is used to refer to controlling access to functions or operations, and the term “permissions” is used to refer to controlling access to particular objects. However, to maintain backwards compatibility with existing releases, the command line arguments for admincli were not modified.

The PATROL Console Server uses permissions to control access to objects, such as management profiles, managed systems, managed system groups, and application classes. The PATROL Console Server stores who has which permissions in an access control database.

For a list of permissions used by PATROL Central Operator and what they mean for each object, see the PATROL Central Operator – Microsoft Windows Edition online Help or the PATROL Central Operator – Web Edition online Help.

You manage the assignments of permissions for a PATROL Console Server through PATROL Central Administration or through the admincli command line utility. For information about allowing and denying permissions to users or groups, see the PATROL Central Administration – Microsoft Windows Edition online Help or the PATROL Central Administration – Web Edition online Help. For more information about the admincli command line utility, see “admincli” on page 180.

**NOTE**

The PATROL Console Server account (see “PATROL Console Server Account” on page 33) always has all permissions for all objects.
Changing the PATROL Console Server Account or Password

You specify the user name and password of the PATROL Console Server account at installation. For more information, see “PATROL Console Server Account” on page 33.

The user name and password are stored in the `patrol.conf` file, which is located in the following directory:

- `%BMC_ROOT%\common\patrol.d\patrol.conf` (Windows)
- `/etc/patrol.d/patrol.conf` (Unix)

On Windows, the user name and password are also stored in the PATROL Console Server service.

Following is an example of the `[CSERVER]` section of the `patrol.conf` file. In this example, the PATROL Console Server account is `patrol`, and the encrypted password is `FB0A195D60626300`.

```
[ CSERVER ]

defaultAccount = patrol:FB0A195D60626300
```

This section contains the following tasks:

<table>
<thead>
<tr>
<th>Task</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changing the PATROL Console Server Account or Password Using admincli</td>
<td>page 77</td>
</tr>
<tr>
<td>Manually Changing the PATROL Console Server Account or Password on Windows</td>
<td>page 78</td>
</tr>
<tr>
<td>Manually Changing the PATROL Console Server Account or Password on Unix</td>
<td>page 79</td>
</tr>
</tbody>
</table>
Changing the PATROL Console Server Account or Password Using admincli

This task describes how to change the PATROL Console Server account or password on Windows or Unix using the admincli command line utility. To use this utility, the PATROL Console Server must be running.

To Change the PATROL Console Server Account or Password Using admincli

1. Start the PATROL Console Server.
2. In the operating system, create the new account or change the password.
3. If you create a new account, ensure that the account meets all of the requirements.
   On Windows, manually assign the necessary user permissions that are normally assigned by the installation. For more information, see “PATROL Console Server Account” on page 33.
4. Add the new account to all of the PATROL groups (patadm, patpop, patop, patscadm, patwatch).
5. Run the admincli command line utility with the necessary options to change the default account.
   For more information about the admincli command line utility, see “admincli” on page 180.
   For example:
   ```bash
   % admincli -c consoleServiceID -user username -pwd encryptedPassword -table config -op set -def_user username -def_pass encryptedPassword
   ```
6. Restart the PATROL Console Server.
Manually Changing the PATROL Console Server Account or Password on Windows

This task describes how to manually change the PATROL Console Server account or password on Windows.

**To Manually Change the PATROL Console Server Account or Password on Windows**

1. Stop the PATROL Console Server.
2. In the operating system, create the new account or change the password.
3. If you create a new account, manually assign the necessary user permissions that are normally assigned by the installation program, and ensure that the account meets all other requirements for the account.

   For more information, see “PATROL Console Server Account” on page 33.

4. Add the new account to all of the PATROL groups (patadm, patpop, patop, patscadm, patwatch).
5. Update the user name or password used by the PATROL Console Server service.

   You can use either of the following methods to update the account information for the service:

   - Update the user name or password by using the Services MMC snap-in.
   - Run the PATROL Console Server with the -install command line argument to install the service with the new user name and password. For more information about these command line arguments, see “Command Line Arguments for the PATROL Console Server” on page 67.

6. Update the user name or password in the `patrol.conf` file.
   
   A. Encrypt the new password by using the `sec_encrypt_p3x` command line utility.

   ```
   %BMC_ROOT%/common\bmc\bin\Windows-x86\sec_encrypt_p3x.exe -patrol_conf plainTextPassword
   ```

   For more information about the `sec_encrypt_p3x` command line utility, see “sec_encrypt_p3x” on page 198.
Changing the PATROL Console Server Account or Password

B Locate the *patrol.conf* file in the `%BMC_ROOT%\common\patrol.d\` directory and open it in a text editor.

C In the *patrol.conf* file, locate the `[CSERVER]` section, and replace the user name and encrypted password in the `defaultAccount` variable.

You can also edit the *patrol.conf* file by using the confManager utility. For more information about confManager, see "confManager" on page 194.

7 Restart the PATROL Console Server.

Manually Changing the PATROL Console Server Account or Password on Unix

This task describes how to manually change the PATROL Console Server account or password on Unix while the PATROL Console Server is not running.

To Manually Change the PATROL Console Server Account or Password on Unix

1 Stop the PATROL Console Server.

2 In the operating system, create the new account or change the password.

If you create a new account, ensure that it meets all of the requirements for the PATROL Console Server account listed under “PATROL Console Server Account” on page 33.

3 Add the new account to all of the PATROL groups (patadm, patpop, patop, patscadm, patwatch).

4 In the operating system for the PATROL Console Server, ensure that the new account has access to the files for the PATROL Console Server.

You might have to change the permissions or ownership of the files and directories so that the new account can read and write to the appropriate PATROL Console Server files.
5 Update the user name or password in the `patrol.conf` file.

A Encrypt the new password by using the `sec_encrypt_p3x` command line utility.

```bash
$BMC_ROOT/common/bmc/bin/target/sec_encrypt_p3x -patrol_conf
plainTextPassword
```

For more information about the `sec_encrypt_p3x` command line utility, see “`sec_encrypt_p3x`” on page 198.

B Locate the `patrol.conf` file in the `/etc/patrol.d/` directory and open it in a text editor.

C In the `patrol.conf` file, locate the `[CSERVER]` section, and replace the user name and encrypted password in the `defaultAccount` variable.

You can also edit the `patrol.conf` file by using the `confManager` utility. For more information about `confManager`, see “`confManager`” on page 194.

6 Restart the PATROL Console Server.

---

**Controlling the RTserver Used**

By default, the PATROL Console Server connects to a single RTserver cloud, but you can configure it to connect to more than one RTserver cloud.

Typically, a single cloud is sufficient for anywhere between 500–700 PATROL Agents, depending on the monitored environment and your RTserver hardware. However, in cases where a single cloud is not sufficient, you may configure additional RTserver clouds to accommodate more PATROL Agents. For more information, see the PATROL Central Infrastructure Best Practices Guide.

In a default configuration with a single RTserver cloud, you can specify which RTserver the PATROL Console Server uses. There are several methods you can use to do this.

For more information about these methods and how to configure the PATROL Console Server to use a single RTserver cloud, see “Configuring Connection to a Single RTserver Cloud” on page 90.

For more information about configuring PATROL Console Server to use multiple RTserver clouds, see “Configuring Connections to Multiple RTserver Clouds” on page 93.

For more information about specifying the RTserver, see Chapter 6, “Configuring RTservers and Components That Use RTservers.”
You can configure the PATROL Console Server to better suit the needs of your particular installation by editing options that are stored in an ASCII configuration file. This file is the `acfg_cserver_serviceID.mof` file, where `serviceID` is a unique instance of the PATROL Console Server.

The `acfg_cserver_serviceID.mof` file is located in the following directory:

- `%BMC_ROOT%\Patrol7\config\cserver` (Windows)
- `$BMC_ROOT/Patrol7/config/cserver` (Unix)

For more information about service IDs, see “Choosing a Service ID for the PATROL Console Server” on page 64.

Do not modify any options that are not documented in this book, in a Technical Bulletin, or in Release Notes.

This section contains the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Editing the Startup Configuration File</td>
<td>page 82</td>
</tr>
<tr>
<td>Configuring Account Order for Authenticating to PATROL Agents</td>
<td>page 83</td>
</tr>
<tr>
<td>Controlling Scanning for PATROL Agent Information at Startup</td>
<td>page 85</td>
</tr>
<tr>
<td>Configuring Audit Logs</td>
<td>page 87</td>
</tr>
<tr>
<td>Configuring Connection to a Single RTserver Cloud</td>
<td>page 90</td>
</tr>
<tr>
<td>Configuring Connections to Multiple RTserver Clouds</td>
<td>page 93</td>
</tr>
<tr>
<td>Supporting Different Security Levels on Multiple RTserver Clouds</td>
<td>page 99</td>
</tr>
<tr>
<td>Configuring Management Profiles</td>
<td>page 100</td>
</tr>
<tr>
<td>Preventing PATROL Console Server Overload</td>
<td>page 109</td>
</tr>
<tr>
<td>Collapsing All Top-Level Objects When Opening a Management Profile</td>
<td>page 112</td>
</tr>
<tr>
<td>Limiting the Number of Computers Per Management Profile</td>
<td>page 114</td>
</tr>
<tr>
<td>Configuring the Known Services Cache</td>
<td>page 115</td>
</tr>
<tr>
<td>Configuring the Number of Process Thread Pools</td>
<td>page 117</td>
</tr>
<tr>
<td>Configuring Protocol Compression</td>
<td>page 117</td>
</tr>
<tr>
<td>Saving ACLs and Impersonation Files at Regular Intervals</td>
<td>page 119</td>
</tr>
</tbody>
</table>
Editing the Startup Configuration File

The PATROL Console Server has a startup configuration file for each service ID that it uses. You can edit this startup configuration file to customize how the PATROL Console Server behaves.

Modify only the `acfg_cserver_ServiceID.mof` file in the `config/cserver` directory. Do not modify any of the `.mof` files in the `lib/mof` directory.

To Edit the Startup Configuration File

1. Stop the PATROL Console Server.
2. Locate the `acfg_cserver_ServiceID.mof` file in the following directory:
   - `%BMC_ROOT%\Patrol7\config\cserver\` (Windows)
   - `$BMC_ROOT/Patrol7/config/cserver/` (Unix)
   
   **NOTE**
   
   Make a backup copy of the file before editing it.

3. Open the file in a text editor.
4. Edit the file as necessary.
   
   **NOTE**
   
   If the option does not already exist, you can add the option. The option can be added anywhere between the `{}` braces that define the subsection. Lines that begin with `//` are treated as comments.

   For more information about the options that you can configure, see any of the topics on page 81.

5. Save and close the file.
6. If you are using advanced security, resign the file.
   
   For more information about signing files, see the *PATROL Security User Guide*.

7. Restart the PATROL Console Server.
   
   The new configuration takes effect when the PATROL Console Server is restarted.
Configuring Account Order for Authenticating to PATROL Agents

When the PATROL Console Server connects to a PATROL Agent it must provide a user account known to the PATROL Agent computer. The PATROL Console Server can attempt to connect by using the following accounts:

- the console account, which was used to log on to the PATROL Console Server
- an account from the impersonation table
- an account provided by the user

If the PATROL Console Server attempts to connect to a PATROL Agent by using an account that is not valid on the PATROL Agent, the account can become locked out, depending on the account policies. You can minimize the chances that the PATROL Console Server will use an invalid account by controlling the order in which the PATROL Console Server uses these accounts.

The order in which the PATROL Console Server uses these accounts is determined by the `mlmAuthOrder` option in the startup configuration file. Table 10 lists the available `mlmAuthOrder` options.

**Table 10  Authentication Order for Connecting to PATROL Agents (Part 1 of 2)**

<table>
<thead>
<tr>
<th><code>mlmAuthOrder</code> option</th>
<th>Order of Accounts</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1. console account</td>
<td>This is the default option.</td>
</tr>
<tr>
<td></td>
<td>2. account from the impersonation table</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3. account provided by the user</td>
<td></td>
</tr>
</tbody>
</table>
Table 10 Authentication Order for Connecting to PATROL Agents (Part 2 of 2)

<table>
<thead>
<tr>
<th>mlmAuthOrder option</th>
<th>Order of Accounts</th>
<th>Comments</th>
</tr>
</thead>
</table>
| 1                   | 1. account from the impersonation table  
                      2. console account  
                      3. account provided by the user | Choose this option if the impersonation table is set up for most cases and you want to use the console account when the impersonation table does not contain a valid account. |
| 2                   | 1. account from the impersonation table  
                      2. account provided by the user | Choose this option to never use the console account to authenticate to the PATROL Agent. |

For example, if you want the PATROL Console Server to use the impersonation table first and the console account second, enter the line in bold in the startup configuration file.

For instructions about how to edit the `acfg_cserver_ServiceID.mof` file, see “Editing the Startup Configuration File” on page 82.

```plaintext
instance of Acfg_8_0_0_Communication
{
    cosPath = "AgentSetup/communication";
    
    enableMLMAuth = true;
    
    // use authentication order #1 to look in the impersonation table first
    mlmAuthOrder = 1;
}
```

For more information about how the impersonation table works, see the PATROL Central Administration – Microsoft Windows Edition Help or PATROL Central Administration – Web Edition Help.
Controlling Scanning for PATROL Agent Information at Startup

The PATROL Console Server scans PATROL Agents for information such as PATROL Agent version, operating system, ports, host names, and other attributes. Normally the PATROL Console Server scans for this information at startup so that it is available when you add PATROL Agents to a management profile.

However, if the account used to gather this information is not valid on the PATROL Agent computer or if the password is incorrect for the PATROL Agent computer, the account can become locked out, depending on the account policies.

To avoid problems with accounts becoming locked out, you can disable scanning PATROL Agents for information, or you can specify which accounts to use on each PATROL Agent.

The `autoScanServices` and `kscRemoteUserContext` options in the startup configuration file determine whether scanning is enabled and which accounts to use.

If scanning PATROL Agents is enabled (autoScanServices = true), the PATROL Console Server consults its impersonation table to determine which account to use for the virtual user name specified by the `kscRemoteUserContext` option in the startup configuration file. If `kscRemoteUserContext` is set to an empty string (""), the virtual user name is the user name for the PATROL Console Server Account. The default value of `kscRemoteUserContext` is `ksc`.

If there is no matching row in the impersonation table for the virtual user name and a particular PATROL Agent, the PATROL Console Server cannot authenticate to the PATROL Agent, and thus, cannot gather information about that PATROL Agent.

For example, if you want the PATROL Console Server to scan for PATROL Agent information and use the virtual user name `ksc` in the impersonation table, enter the lines in bold in the startup configuration file.

For instructions about how to edit the `acfg_cserver_ServiceID.mof` file, see “Editing the Startup Configuration File” on page 82.

```plaintext
instance of Acfg_7_1_0_KSC
{
    cosPath = "AgentSetup/ksc";
    enabled = true;
    autoScanServices = true;
    kscRemoteUserContext = "ksc";
};
```
**TIP**

To *suppress* scanning for PATROL Agent information, set `autoScanServices` to false.

To *enable* scanning for PATROL Agent information, set `autoScanServices` to true, and set `kscRemoteUserContext` to the virtual user name to use in the impersonation table.

To ensure that the entries in the impersonation table for this virtual user name are used only for scanning for PATROL Agent information, choose a name that is not an actual account on the PATROL Console Server. If you choose an actual account on the PATROL Console Server, that user will have access to the specified PATROL Agents.

If you enabled scanning, use PATROL Central Administration to update the impersonation table for the virtual user name that is specified by `kscRemoteUserContext`.

For more information about updating the impersonation table, see “Example 1: An Impersonation Table With Wildcard Users” on page 86 and “Example 2: An Impersonation Table With Named Users” on page 87. For more information about how the impersonation table works, see the PATROL Central Administration – Microsoft Windows Edition Help or PATROL Central Administration – Web Edition Help.

**Example 1: An Impersonation Table With Wildcard Users**

Any rows in the impersonation table with the wildcard character for the user also apply to the virtual user name specified by the `kscRemoteUserContext` option.

If a PATROL Agent is in a row with the wildcard character for the user, you do not need to create an additional row for that PATROL Agent with the virtual user name.

<table>
<thead>
<tr>
<th>User/Group</th>
<th>User</th>
<th>Service Type</th>
<th>Service Name</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>*</td>
<td>Managed System</td>
<td>MYAGENT_3181</td>
<td>MyAlias1</td>
</tr>
<tr>
<td>User</td>
<td>*</td>
<td>Managed System</td>
<td>*</td>
<td>MyAlias2</td>
</tr>
</tbody>
</table>
Example 2: An Impersonation Table With Named Users

If the impersonation table contains rows for specific named users, those rows do not apply to the virtual user name used by the PATROL Console Server to gather PATROL Agent information at startup.

If you want the PATROL Console Server to gather PATROL Agent information for those PATROL Agents, duplicate those rows, substituting the specific named users with the virtual user name (\textit{ksc} by default).

<table>
<thead>
<tr>
<th>User/Group</th>
<th>User/Group Name</th>
<th>Service Type</th>
<th>Service Name</th>
<th>Alias</th>
</tr>
</thead>
<tbody>
<tr>
<td>User</td>
<td>MyUser</td>
<td>Managed System</td>
<td>MYAGENT_3181</td>
<td>MyAlias1</td>
</tr>
<tr>
<td>User</td>
<td>\textit{ksc}</td>
<td>Managed System</td>
<td>MYAGENT_3181</td>
<td>MyAlias1</td>
</tr>
<tr>
<td>User</td>
<td>MyUser</td>
<td>Managed System</td>
<td>*</td>
<td>MyAlias2</td>
</tr>
<tr>
<td>User</td>
<td>\textit{ksc}</td>
<td>Managed System</td>
<td>*</td>
<td>MyAlias2</td>
</tr>
</tbody>
</table>

Configuring Audit Logs

By default, audit messages are written to the PATROL Console Server application log, which is the existing PATROL Console Server log. These messages are identified by the prefix \textit{AUDIT}. However, you can have audit messages written to a separate dedicated audit log file. The type of audit log used is determined by the \textit{useAuditLog} option.

Audit messages are written for the following:

- authentication attempts, including failed authentications
- disconnects for authenticated clients of the PATROL Console Server
- successful lookups in the impersonation table
- opening and closing of the audit log file
- server shutdowns
- attempts to open DLL files
- attempts to validate digital signatures
- security level and policy used by the server
additions, deletions, and changes to right assignments in the access control backend

- additions, deletions, and changes to permission assignments in the access control backend

- additions, deletions, and changes to the impersonation and alias tables

- additions, deletions, and changes to managed system groups and managed systems in the group

- suspending and resuming KM parameters

- customizing KM application classes and parameters

- KM menu commands

- snooze events

The audit level is determined by the `auditLevel` option.

You can also manage the size of log files by periodically starting a new log file, renaming existing log files as necessary, and deleting extra log files. This file-aging configuration is determined by the `maxMessagesInAppLog`, `maxAppLogFileSize`, `maxMessagesInAuditLog`, and `maxAuditLogFileSize` options.

The audit level is determined by the `auditLevel` option.

Table 11 lists the audit log options in the configuration file. These options are found in the instance of `Acfg_8_0_0_Generic`.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>maxMessagesInAppLog</code></td>
<td>defines the maximum number of messages recorded in the application log before file aging occurs</td>
</tr>
<tr>
<td></td>
<td>The default value is 0. A value of 0 indicates that file aging does not occur based on the number of messages in the application log file.</td>
</tr>
<tr>
<td><code>maxAppLogFileSize</code></td>
<td>defines the maximum number of bytes recorded in the application log before file aging occurs</td>
</tr>
<tr>
<td></td>
<td>The default value is 0. A value of 0 indicates that file aging does not occur based on the byte size of the application log file.</td>
</tr>
<tr>
<td><code>auditLevel</code></td>
<td>controls the audit level</td>
</tr>
<tr>
<td></td>
<td>The default value is 1. The valid range is 1–3. Level 1 provides the least amount of logging and includes tracking user connections. Levels 2 and 3 provide an increased level of auditing.</td>
</tr>
</tbody>
</table>
If both the `maxMessagesIn<logtype>Log` and `max<logtype>LogFileSize` options have non-zero values, the `maxMessagesIn<logtype>Log` option will be used to determine file aging.

If both the `maxMessagesIn<logtype>Log` and `max<logtype>LogFileSize` options are set to zero, file aging occurs when the PATROL Console Server is started.

For example, if you want the PATROL Console to write audit messages to a separate audit log file with the highest level of auditing, and you want file aging to occur after 100 messages are written to the log, enter the lines in bold in the startup configuration file.

For instructions about how to edit the `acfg_cserver_ServiceID.mof` file, see “Editing the Startup Configuration File” on page 82.

### Table 11  Audit Log Options in the PATROL Console Server Configuration File (Part 2 of 2)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>useAuditLog</td>
<td>specifies the destination of the audit messages</td>
</tr>
<tr>
<td></td>
<td>The default value is false. A value of false specifies that audit messages</td>
</tr>
<tr>
<td></td>
<td>are written to the PATROL Console Server application log. A value of true</td>
</tr>
<tr>
<td></td>
<td>specifies that audit messages are written to a separate dedicated audit log</td>
</tr>
<tr>
<td></td>
<td>file.</td>
</tr>
<tr>
<td></td>
<td>The dedicated audit log is located in the <code>PATROL_ROOT_LOG</code> directory and is</td>
</tr>
<tr>
<td></td>
<td>assigned a file extension of <code>.audit_log</code>.</td>
</tr>
<tr>
<td>maxMessagesInAuditLog</td>
<td>defines the maximum number of messages recorded in the audit log</td>
</tr>
<tr>
<td></td>
<td>before file aging occurs</td>
</tr>
<tr>
<td></td>
<td>The default value is 0. A value of 0 indicates that file aging does not</td>
</tr>
<tr>
<td></td>
<td>occur based on the number of messages in the audit log file.</td>
</tr>
<tr>
<td></td>
<td>This option is valid only if the <code>useAuditLog</code> option is set to true.</td>
</tr>
<tr>
<td>maxAuditLogFileSize</td>
<td>defines the approximate maximum number of bytes recorded in the audit log</td>
</tr>
<tr>
<td></td>
<td>before file aging occurs</td>
</tr>
<tr>
<td></td>
<td>The maximum number of bytes is not exact; rollover occurs around the</td>
</tr>
<tr>
<td></td>
<td>number defined.</td>
</tr>
<tr>
<td></td>
<td>The default value is 0. A value of 0 indicates that file aging does not</td>
</tr>
<tr>
<td></td>
<td>occur based on the byte size of the audit log file.</td>
</tr>
<tr>
<td></td>
<td>This option is valid only if the <code>useAuditLog</code> option is set to true.</td>
</tr>
</tbody>
</table>
Configuring Connection to a Single RTserver Cloud

You can configure multiple RTservers to form an RTserver cloud. Using an RTserver cloud provides flexibility and load balancing. Typically, a single cloud composed of two RTservers is sufficient for anywhere between 500–700 PATROL Agents, depending on the monitored environment and your RTserver hardware. For more information, see the PATROL Central Infrastructure Best Practices Guide.

However, in cases where a single cloud configuration is not sufficient, use the multiple cloud configuration to accommodate more PATROL Agents. For more information, see “Configuring Connections to Multiple RTserver Clouds” on page 93.

In a default configuration with a single RTserver cloud, you can specify which RTserver the PATROL Console Server uses by using either of the following methods, which are listed in order of precedence:

1. -rtserver command line option (if specified)

   For more information, see “Command Line Arguments for the PATROL Console Server” on page 67.

2. RTSERVERS environment variable

   This method is the default method, and the environment variable is set at installation.
3. \textit{rtServerLocators} option

This option is specified in the \texttt{acfg\_cserver\_ServiceID.mof} file.

\textbf{Table 12} lists the options you can edit to configure the PATROL Console Server to communicate with a single RTserver cloud. These options are found in the instance of \texttt{Acfg\_8\_0\_0\_Communication}.

For the majority of cases, the default values are sufficient. The default values should only be modified to address a specific issue.

For instructions about how to edit the \texttt{acfg\_cserver\_ServiceID.mof} file, see “Editing the Startup Configuration File” on page 82.

\begin{table}[h]
\centering
\begin{tabular}{|l|l|}
\hline
Option & Description \\
\hline
rtServerLocators & defines the set of RTserver locator strings that are used to connect to the RTserver cloud, based on the same syntax used for the RTSERVERS environment variable \\
& The default value is tcp:localhost:2059,tcp:bmcrtserver:2059,auto:*2059. \\
& The value for this option is used only if the RTSERVERS environment variable is not defined. \\
& The order to locate the default RTserver cloud setting is: \\
& 1. -rtserver command line argument \\
& 2. value of the RTSERVERS environment variable \\
& 3. value of the rtServerLocators option \\
\hline
\end{tabular}
\end{table}
Configuring PATROL Console Server Options

Table 12  Single-cloud Options in the PATROL Console Server Configuration File (Part 2 of 3)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serviceName</td>
<td>defines the name used by the PATROL Console Server to advertise itself in an RTserver cloud. The default value is PATROL_CSERVER_%{id}, where %{id} represents a macro that is replaced with the PATROL Console Server service ID to construct the full service name. The default value used for substitution is the host name. However, if the -id command line is specified, the value in the -id command line is used. For more information about service IDs, see “Choosing a Service ID” on page 269. Typically, the value of the serviceName option does not need to be changed. However, there are a few cases in which the value can be changed to enter an explicit service name, such as PATROL_CSERVER_ACME_WIDGETS. If an explicit value is specified, then the -id command line argument will have no effect on the service name that is used to advertise the PATROL Console Server in the RTserver cloud.</td>
</tr>
<tr>
<td>serviceNameAliases</td>
<td>defines a list of additional service names used by the PATROL Console Server to identify itself in an RTserver cloud. The names in this list are advertised in addition to the value specified by the serviceName option. The default value is PATROL_CCSERVER_%{id}, which is used by the Common Connect Back End.</td>
</tr>
<tr>
<td>conTimeout</td>
<td>defines the connection timeout for the RTserver cloud in milliseconds. The default value is 20000, which is equivalent to 20 seconds. This option applies to the initial negotiation that happens when a socket connection is established to the RTserver. The negotiation is a two-phase process. The first phase is to connect to the RTserver, and the second phase is to negotiate and receive a list of existing RTservers from the RTserver cloud. Until a connection is established, the first phase may be repeated x times if the RTServer locators string has x RTservers, with each attempt getting a full timeout. Once phase one completes successfully for an RTserver, then the second phase countdown begins.</td>
</tr>
</tbody>
</table>
Configuring PATROL Console Server Options

Chapter 3 Administering and Configuring the PATROL Console Server 93

If one or more instances of the configuration class Acfg_8_0_0_RtCloudConnection, which is used to configure multiple clouds, are defined, then the values for configuring a single cloud are ignored.

### Configuring Connections to Multiple RTserver Clouds

Connecting to more than one RTserver cloud allows you to view a much larger number of PATROL Agents in a single management profile. Typically, you can have between 400–750 PATROL Agents on a single cloud, depending on the monitored environment and your RTserver hardware. If you want to monitor more PATROL Agents than recommended for a single cloud in your environment, you can create several RTserver clouds to accommodate more PATROL Agents. For more information, see the PATROL Central Infrastructure Best Practices Guide.

To configure the PATROL Console Server to connect to more than one RTserver cloud at the same time, you will have to add additional instances to the acfg_cserver_ServiceID.mof file.

### To Enable Multiple Clouds

1. In the acfg_cserver_ServiceID.mof file, add an instance of Acfg_8_0_0_RtCloudConnection for each cloud to which the PATROL Console Server will connect.

   **NOTE**

   Add instances of Acfg_8_0_0_RtCloudConnection at the end of the acfg_cserver_ServiceID.mof file.

   Each instance of Acfg_8_0_0_RtCloudConnection must have a unique name. These names are used to identify the different RTserver clouds in various log messages.

### Table 12 Single-cloud Options in the PATROL Console Server Configuration File (Part 3 of 3)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>keepAliveTimeout</td>
<td>defines the connection timeout for heartbeat messages sent to the RTserver in milliseconds</td>
</tr>
<tr>
<td></td>
<td>The default value is 12000, which is equivalent to 120 seconds.</td>
</tr>
<tr>
<td></td>
<td>If the RTserver fails to respond within two of these keep-alive intervals, then it is assumed that the RTserver is no longer available.</td>
</tr>
<tr>
<td></td>
<td>If the value is set to zero, then no keep-alive messages are sent.</td>
</tr>
</tbody>
</table>

If one or more instances of the configuration class Acfg_8_0_0_RtCloudConnection, which is used to configure multiple clouds, are defined, then the values for configuring a single cloud are ignored.
If only one instance of `Acfg_8_0_0_RtCloudConnection` is defined, then the PATROL Console Server uses that instance as the only cloud to which it connects.

**NOTE**

*Do not* create two or more instances of `Acfg_8_0_0_RtCloudConnection` for the same physical cloud.

Refer to the following example for the minimum instance definition, where `name` is a unique string used to identify each cloud, and `rtLocatorString` is the standard locator string format (tcp:hostname:port#):

```plaintext
instance of Acfg_8_0_0_RtCloudConnection
{
  cosPath = "AgentSetup/communication/<name>";
  rtServerLocators = "<rtLocatorString>";
};
```

2 Define the options you need.

Table 13 on page 95 lists the options you can add to configure the PATROL Console Server to communicate with multiple RTserver clouds. These options must be added under the instance of `Acfg_8_0_0_RtCloudConnection`.

**TIP**

There is no required order when you add options; however, it is recommended to list `cosPath` first.

The existing default values are sufficient for the majority of cases. The default values should only be modified to address a specific issue.

The minimum options required to define an RTserver cloud connection are `cosPath` and `rtServerLocators`.

For instructions about how to edit the `acfg_cserver_ServiceID.mof` file, see “Editing the Startup Configuration File” on page 82.
### Table 13  Multiple Cloud Options in the PATROL Console Server Configuration File (Part 1 of 3)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>rtServerNamespaceId</td>
<td>defines the namespace identifier that is registered with the RTserver cloud</td>
</tr>
<tr>
<td></td>
<td>The PATROL Console Server will only communicate with PATROL Central and PATROL Agents that use the same namespace identifier.</td>
</tr>
<tr>
<td></td>
<td>The default value is <strong>cos</strong>.</td>
</tr>
<tr>
<td>rtServerLocators</td>
<td>defines the set of RTserver locator strings that are used to connect to the RTserver cloud, based on the same syntax used for the RTSERVERS environment variable</td>
</tr>
<tr>
<td></td>
<td>You can include more than one RTserver in this string for backup purposes. All RTservers in this string must be a part of the same cloud.</td>
</tr>
<tr>
<td></td>
<td>The default value is &quot;.&quot;.</td>
</tr>
<tr>
<td>useCloudSpecificCommSecurity</td>
<td>defines whether the PATROL Console Server reads and uses an RTserver cloud-specific security policy or the default security policy.</td>
</tr>
<tr>
<td></td>
<td>Using an RTserver cloud-specific security policy allows different RTserver clouds to operate at different security levels. For more information, see “Supporting Different Security Levels on Multiple RTserver Clouds” on page 99.</td>
</tr>
<tr>
<td></td>
<td>The possible values for this option are:</td>
</tr>
<tr>
<td></td>
<td>- true—If you specify <code>true</code>, then the PATROL Console Server reads and uses the security policy that is specific to the RTserver.</td>
</tr>
<tr>
<td></td>
<td>- false—If you specify <code>false</code>, then the PATROL Console Server reads and uses the default security policy.</td>
</tr>
<tr>
<td></td>
<td>The default value is false.</td>
</tr>
</tbody>
</table>
Configuring PATROL Console Server Options

Table 13  Multiple Cloud Options in the PATROL Console Server Configuration File (Part 2 of 3)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>serviceName</td>
<td>defines the service name used by the PATROL Console Server to advertise itself in the cloud that is identified by this entry</td>
</tr>
<tr>
<td></td>
<td>The possible values for this option are:</td>
</tr>
<tr>
<td></td>
<td>■ &quot; &quot;—If you specify an empty string, then the default service name is used. The default service name is the value specified by Acfg_8_0_0_Communication::serviceName.</td>
</tr>
<tr>
<td></td>
<td>■ none—If you specify the string none, then the PATROL Console Server will not advertise itself in this cloud. This is useful in situations where you want to have a dedicated RTserver cloud for the DMZ, but you do not want PATROL Agents in that DMZ to have visibility to the rest of the enterprise.</td>
</tr>
<tr>
<td></td>
<td>■ service_name—If you specify a specific service name, then this value is used in advertising instead of the default, and the default aliases are ignored.</td>
</tr>
<tr>
<td></td>
<td>The default value is ” ”.</td>
</tr>
<tr>
<td>serviceNameAliases</td>
<td>defines a list of additional service names that are advertised by the PATROL Console Server in the cloud identified by this entry</td>
</tr>
<tr>
<td></td>
<td>If the serviceName option is not specified, then this value is appended to the alias list defined by Acfg_8_0_0_Communication::serviceNameAliases.</td>
</tr>
<tr>
<td></td>
<td>If the serviceName option is none, then this value is ignored.</td>
</tr>
<tr>
<td></td>
<td>If the serviceName option is given a specific value, then this value is the only alias list, and Acfg_8_0_0_Communication::serviceNameAliases is ignored.</td>
</tr>
</tbody>
</table>
### Configuring PATROL Console Server Options

#### Chapter 3 Administering and Configuring the PATROL Console Server

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>conTimeout</td>
<td>defines the connection timeout, in milliseconds, for the RTserver cloud&lt;br&gt;The default value is 20000, which is equivalent to 20 seconds.&lt;br&gt;This option applies to the initial negotiation that happens when a socket connection is established to the RTserver. The negotiation is a two-phase process. The first phase is to connect to the RTserver, and the second phase is to negotiate and receive a list of existing RTservers from the RTserver cloud.&lt;br&gt;Until a connection is established, the first phase may be repeated x times if the RTserver locators string has x RTservers, with each attempt getting a full timeout. Once phase one completes successfully for an RTserver, then the second phase countdown begins.</td>
</tr>
<tr>
<td>keepAliveTimeout</td>
<td>defines the connection timeout, in milliseconds, for heartbeat messages sent to the RTserver&lt;br&gt;The default value is 120000, which is equivalent to 120 seconds.&lt;br&gt;If the RTserver fails to respond within two of these keep-alive intervals, then it is assumed that the RTserver is no longer available.&lt;br&gt;If the value is set to zero, then no keep-alive messages are sent.</td>
</tr>
</tbody>
</table>
Example of a Multi-cloud Configuration

There are numerous configurations you can construct based on the multiple cloud capability.

In this example, there are three clouds:

- Cloud 1, named Consoles, contains PATROL Central Operator consoles
- Cloud 2, named Agents – 1, contains one group of PATROL Agents
- Cloud 3, named Agents – 2, contains a second group of PATROL Agents

All three clouds have a primary and secondary RTserver. All three clouds have a different security level. The PATROL Console Server connects to all three clouds based on the following definition:

```plaintext
instance of Acfg_8_0_0_RtCloudConnection
{
    cosPath="Agent Setup/communication/Consoles";
    rtServerLocators="tcp:consoleRTA:2059, tcp:consoleRTB:2059";
    useCloudSpecificCommSecurity=true;
};

instance of Acfg_8_0_0_RtCloudConnection
{
    cosPath="Agent Setup/communication/Agents – 1";
    rtServerLocators="tcp:agentRT1A:2059, tcp:agentRT1B:2059";
    useCloudSpecificCommSecurity=true;
};

instance of Acfg_8_0_0_RtCloudConnection
{
    cosPath="Agent Setup/communication/Agents – 2";
    rtServerLocators="tcp:agentRT2A:2059, tcp:agentRT2B:2059";
    useCloudSpecificCommSecurity=true;
};
```
Supporting Different Security Levels on Multiple RTserver Clouds

PATROL Console Server can support different security levels (0–4) for each RTserver cloud. You can configure whether an RTserver cloud uses an RTserver cloud-specific security policy or the default security policy. To do this you must:

1. Define the useCloudSpecificCommSecurity option in the configuration file for each RTserver cloud with a unique security level.

2. Create the proper client/server stanzas in the security policy file based on the name of the RTserver cloud that you defined in the configuration file.

   For Unix, the security policy file is:
   `/etc/patrol.d/security_policy_v3.0/cserver_rtCloudName.plc`

   For Windows, the security policy file is stored in the following registries:

   - `\HKEY_LOCAL_MACHINE\SOFTWARE\BMC Software\PATROL\SecurityPolicy_v3.0\cserver\rtCloudName`
   - `\HKEY_LOCAL_MACHINE\SOFTWARE\BMC Software\PATROL\SecurityPolicy_v3.0\cserver\rtCloudName\server`

   For per-cloud security levels of 0–2, you can use a simple template and specify the required security level and the new log filename to use. For per-cloud security levels of 3 or 4, you will have to perform custom modifications.

   PATROL Console Server provides templates you can use. You will have to fill in the TARGET value, which is the installation path. These templates are located in the following directory:

   - `%BMC_ROOT%\Patrol7\security` (Windows)
   - `$BMC_ROOT/Patrol7/security` (Unix)

   For more information about editing security policies, see the PATROL Security User Guide.
The following example illustrates a sample stanza for security levels 0–2:

```
Filename: cserver-<name>.plc
[client]
logfile=cserver_<name>_client.log
security_level=2

[server]
logfile=cserver_<name>_server.log
security_level=2
```

### Configuring Management Profiles

A management profile is a view of your PATROL environment that is stored on the PATROL Console Server and accessed by using PATROL Central Operator. A management profile contains the following information:

- any managed systems (PATROL Agents) that you have added
- any Knowledge Modules (KMs) that you have loaded
- other miscellaneous preferences and settings

You can make changes to a management profile to better suit your needs. Table 14 on page 101 lists the configuration options that are directly related to management profiles. These options must be defined in the instance of `Acfg_7_1_0_ConsoleServer`.

For instructions about how to edit the `acfg_cserver_ServiceID.mof` file, see “Editing the Startup Configuration File” on page 82.
Table 14  Management Profile Options in the PATROL Console Server Configuration File (Part 1 of 7)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Options for overload protection:</td>
<td></td>
</tr>
<tr>
<td>■ percentAddressSpaceLimit</td>
<td></td>
</tr>
<tr>
<td>■ maxVirtualMemorySize</td>
<td></td>
</tr>
<tr>
<td>■ maxProfilesOpen</td>
<td></td>
</tr>
<tr>
<td>■ maxReadWriteProfilesOpen</td>
<td></td>
</tr>
<tr>
<td>■ maxObjectCount</td>
<td></td>
</tr>
<tr>
<td>■ maxComputerCount</td>
<td></td>
</tr>
<tr>
<td>For information about these options, see Table 15 on page 109.</td>
<td></td>
</tr>
<tr>
<td>For more information about using these options, see “Preventing PATROL Console Server Overload” on page 109.</td>
<td></td>
</tr>
<tr>
<td>Options for limiting the number computers per management profile:</td>
<td>For information about these options, see Table 16 on page 115.</td>
</tr>
<tr>
<td>■ privilegedUserMaxComputersPerProfile</td>
<td>For more information about using these options, see “Limiting the Number of Computers Per Management Profile” on page 114.</td>
</tr>
<tr>
<td>■ unprivilegedUserMaxComputersPerProfile</td>
<td></td>
</tr>
<tr>
<td>newLayoutPreloadList</td>
<td>defines the .mof files that will be used to add selected user-defined items to new management profiles</td>
</tr>
<tr>
<td>By default, this file loads the following items:</td>
<td></td>
</tr>
<tr>
<td>■ predefined agent queries</td>
<td></td>
</tr>
<tr>
<td>■ predefined LEM filters</td>
<td></td>
</tr>
<tr>
<td>■ predefined Agent Selection Filters</td>
<td></td>
</tr>
<tr>
<td>For more information about using this option, see “Creating New Default Managed System Queries, Event Filters, and Chart Schemes” on page 141.</td>
<td></td>
</tr>
<tr>
<td>autoSaveMode</td>
<td>defines whether the PATROL Console Server automatically saves the changes that are made to read/write management profiles</td>
</tr>
<tr>
<td>The default value is true. A value of true indicates that changes are automatically saved on a regular interval defined by the autoSaveTimer option.</td>
<td></td>
</tr>
<tr>
<td>If the value is set to false, changes are saved only when the management profile is closed.</td>
<td></td>
</tr>
</tbody>
</table>
### Configuring PATROL Console Server Options

#### Option

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>autoSaveTimer</td>
<td>defines the interval, in seconds, that opened read/write management profiles are saved</td>
</tr>
<tr>
<td></td>
<td>The default value is 450, which is equivalent to 7.5 minutes.</td>
</tr>
<tr>
<td></td>
<td>This option is used only if the autoSaveMode option is set to true.</td>
</tr>
<tr>
<td>collapseTopLevelChildrenLimit</td>
<td>defines whether the PATROL Console Server collapses all top-level objects, such as PATROL Agents, when you open a management profile</td>
</tr>
<tr>
<td></td>
<td>The default value is 10.</td>
</tr>
<tr>
<td></td>
<td>For more information about using this option, see “Collapsing All Top-Level Objects When Opening a Management Profile” on page 112.</td>
</tr>
<tr>
<td>useP3xLabelsWhenObjectNotHome</td>
<td>defines the way PATROL Agent objects are labeled in the PATROL Console Server</td>
</tr>
<tr>
<td></td>
<td>The default value is false. A value of false indicates that style labels do not change. However, the user can hover on the icon and see the details.</td>
</tr>
<tr>
<td></td>
<td>If the value is set to true, then objects outside their home location are relabeled to indicate the instance and managed system to which they belong.</td>
</tr>
<tr>
<td></td>
<td>This option is valid only when an object is moved from its default location either by the user or by the KM.</td>
</tr>
<tr>
<td>checkAccessControlForParameters</td>
<td>defines whether access control (ACL) checks are performed for parameters</td>
</tr>
<tr>
<td></td>
<td>The default value is false. A value of false indicates that ACL checks are not performed.</td>
</tr>
<tr>
<td></td>
<td>A value of true indicates that ACL checks are performed. Depending on the KMs and ACLs involved, enabling this option may affect your system’s performance.</td>
</tr>
</tbody>
</table>
### Table 14  Management Profile Options in the PATROL Console Server Configuration File (Part 3 of 7)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>verboseConsoleMessages</td>
<td>defines whether certain messages sent by the PATROL Console Server to the PATROL Central Operator message window are suppressed.</td>
</tr>
<tr>
<td></td>
<td>Suppressed messages are informational only, such as Starting to load KM &lt;name&gt;, or KM &lt;KMname&gt; loaded on agent &lt;PATROLAgentname&gt;.</td>
</tr>
<tr>
<td></td>
<td>The default value is false. A value of false indicates that these messages are not sent.</td>
</tr>
<tr>
<td></td>
<td>A value of true indicates that these messages are sent.</td>
</tr>
<tr>
<td>msgSuppressMask</td>
<td>defines whether some of the messages that are displayed by a PATROL Agent’s system output window in the management profiles are suppressed.</td>
</tr>
<tr>
<td></td>
<td>Suppressing these messages helps to generate less network traffic after RTserver failovers and PATROL Console Server restarts.</td>
</tr>
<tr>
<td></td>
<td>The default value is 0x01. A value of 0x01 indicates that KM loading messages are suppressed.</td>
</tr>
<tr>
<td></td>
<td>To see all of the messages, set the value to 0.</td>
</tr>
<tr>
<td></td>
<td>This option only works with PATROL Agent version 3.6.00 or later.</td>
</tr>
</tbody>
</table>
Configuring PATROL Console Server Options

104 PATROL Console Server and RTserver Getting Started

computerConnectBatchSize defines the number of managed systems to which the PATROL Console Server attempts to connect before pausing.

This option works with the computerConnectBatchInterval option to control the rate at which the PATROL Console Server attempts to connect to different managed systems when a management profile is first loaded. This is useful after an RTserver failover or a PATROL Console Server restart, when multiple operators connecting concurrently can trigger a peak load condition as the PATROL Console Server attempts to load several management profiles at once.

Setting the value of the computerConnectBatchSize option and the computerConnectBatchInterval option to 0 causes the PATROL Console Server to immediately attempt to connect to all PATROL Agents in the management profile.

The default value is 60. The valid range is 0–2^{32} power (4GB).

Lower values mean less time to connect to all PATROL Agents in a profile, which increases the network and CPU load. Higher values mean longer times to connect to all PATROL Agents in a management profile.

Table 14 Management Profile Options in the PATROL Console Server Configuration File (Part 4 of 7)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>computerConnectBatchSize</td>
<td>defines the number of managed systems to which the PATROL Console Server attempts to connect before pausing. This option works with the computerConnectBatchInterval option to control the rate at which the PATROL Console Server attempts to connect to different managed systems when a management profile is first loaded. This is useful after an RTserver failover or a PATROL Console Server restart, when multiple operators connecting concurrently can trigger a peak load condition as the PATROL Console Server attempts to load several management profiles at once. Setting the value of the computerConnectBatchSize option and the computerConnectBatchInterval option to 0 causes the PATROL Console Server to immediately attempt to connect to all PATROL Agents in the management profile. The default value is 60. The valid range is 0–2^{32} power (4GB). Lower values mean less time to connect to all PATROL Agents in a profile, which increases the network and CPU load. Higher values mean longer times to connect to all PATROL Agents in a management profile.</td>
</tr>
</tbody>
</table>
**computerConnectBatchInterval** defines the duration of the pause, in seconds, between PATROL Console Server attempts to connect to managed systems.

This option works with the \textit{computerConnectBatchSize} option to control the rate at which the PATROL Console Server attempts to connect to different managed systems when a management profile is first loaded. This is useful after an RTserver failover or a PATROL Console Server restart, when multiple operators connecting concurrently can trigger a peak load condition as the PATROL Console Server attempts to load several management profiles at once.

Setting the value of the \textit{computerConnectBatchSize} option and the \textit{computerConnectBatchInterval} option to 0 causes the PATROL Console Server to immediately attempt to connect to all PATROL Agents in the management profile.

The default value is 5. Depending on load conditions, the actual length of the pause is 1 to 5 times the set value.

The valid range is $0–2^{32}$ power (4GB).

Lower values mean less time to connect to all PATROL Agents in a profile, which increases the network and CPU load. Higher values mean longer times to connect to all PATROL Agents in a management profile.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>computerConnectBatchInterval</td>
<td>defines the duration of the pause, in seconds, between PATROL Console Server attempts to connect to managed systems. This option works with the computerConnectBatchSize option to control the rate at which the PATROL Console Server attempts to connect to different managed systems when a management profile is first loaded. This is useful after an RTserver failover or a PATROL Console Server restart, when multiple operators connecting concurrently can trigger a peak load condition as the PATROL Console Server attempts to load several management profiles at once. Setting the value of the computerConnectBatchSize option and the computerConnectBatchInterval option to 0 causes the PATROL Console Server to immediately attempt to connect to all PATROL Agents in the management profile. The default value is 5. Depending on load conditions, the actual length of the pause is 1 to 5 times the set value. The valid range is $0–2^{32}$ power (4GB). Lower values mean less time to connect to all PATROL Agents in a profile, which increases the network and CPU load. Higher values mean longer times to connect to all PATROL Agents in a management profile.</td>
</tr>
</tbody>
</table>
### Table 14 Management Profile Options in the PATROL Console Server Configuration File (Part 6 of 7)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>computerConnectWhenReachableInterval</td>
<td>defines the time allowed, in seconds, after the PATROL Agent connects to the RTserver namespace before the PATROL Console Server connects and makes requests to load KMs. Most of the production PATROL Agents are configured to have KMs preloaded. Connecting to the managed systems before the KMs are fully instantiated will cause unnecessary network traffic due to the various creation and state change events that are generated as the first instances of each KM are created. Delaying the connection helps the PATROL Agent instantiate all the preloaded KMs, allowing for more efficient retrieval of initial status information. The default value is 45. The valid range is $0–2^{32}$ power (4GB). Set the value of this option so that it is greater than the majority of the discovery intervals in the namespace.</td>
</tr>
</tbody>
</table>

---

**Configuring PATROL Console Server Options**

106 PATROL Console Server and RTserver Getting Started
### Configuring PATROL Console Server Options

#### Chapter 3 Administering and Configuring the PATROL Console Server 107

**Table 14 Management Profile Options in the PATROL Console Server Configuration File (Part 7 of 7)**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>queryBatchSize</td>
<td>defines the number of managed systems that are queried before pausing. This option works with the queryBatchInterval option to control the rate at which the PATROL Console Server sends managed object queries to different managed systems. This helps to manage the load on the RTserver cloud and the PATROL Console Server. This option only applies to “all object” queries and parameter-level queries where no additional filtering has been defined. The default value is 100. The valid range is $0–2^{32}$ power (4GB). Lower values mean faster query responses, which increases the network and server load. Higher values mean longer times to finish all queries.</td>
</tr>
<tr>
<td>queryBatchInterval</td>
<td>defines the duration of the pause, in seconds, between each batch of queries. This option works with the queryBatchSize option to control the rate at which the PATROL Console Server sends managed object queries to different managed systems. This helps to manage the load on the RTserver cloud and the PATROL Console Server. This option only applies to “all object” queries and parameter-level queries where no additional filtering has been defined. The default value is 5. The valid range is $0–2^{32}$ power (4GB). Lower values mean faster query responses, which increases the network and server load. Higher values mean longer times to finish all queries.</td>
</tr>
</tbody>
</table>
The following options can have a significant effect on performance:

- computerConnectBatchSize
- computerConnectBatchInterval
- computerConnectWhenReachableInterval
- queryBatchSize
- queryBatchInterval

Therefore, consider the following issues before making changes to the values of the options listed:

- If you do not have experience with performance problems related to connect times or query times, do not modify these values.

- If you modify these values, you may have to experiment with different combinations to achieve the proper balance for your environment between network and server load versus the completion time for the corresponding task.

- In some environments, improper settings for these values may cause the PATROL Console Server and the RTserver to be overwhelmed, causing them to be unable to recover in certain failover scenarios.
Preventing PATROL Console Server Overload

When the workload exceeds the level that the PATROL Console Server can manage, the PATROL Console Server can run out of memory to handle new requests, or it can run out of CPU resources to handle the incoming requests or updates from PATROL Agents. You can prevent this by editing the options for overload protection in the startup configuration file.

Table 15 lists the overload protection options in the configuration file. These options must be defined in the instance of Acfg_7_1_0_ConsoleServer.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| percentAddressSpaceLimit | defines the maximum percentage of virtual memory used by the PATROL Console Server before triggering the overload condition  
The default value is 75. The valid range is 0–100. A value of 0 indicates that no limit is defined.  
The actual size of the address space available to the application depends on the operating system. On 32-bit operating systems, the size is typically 2 or 3 GB.  
The limits on supported 32-bit platforms are:  
- Windows—2GB (3GB if /3GB boot option is specified where applicable for Windows 2000Advanced Servers or Windows 2003 servers)  
- Solaris—3.75GB (you need to specify using ulimit, default value (which prints as unlimited) corresponds to 2GB.  
- Linux—3GB on generic Linux 2.4 systems; 4GB on RH AS 3.0  
- AIX—3.25GB on AIX 5.1 and 5.2  
This option is ignored on 64-bit platforms. |
| maxVirtualMemorySize | defines the maximum virtual memory, in units of 1024 bytes, used by the PATROL Console Server before triggering the overload condition  
The default value is 0. A value of 0 indicates that no limit is defined.  
This option is valid on both 32- and 64-bit platforms. |
If any of the overload protection options are met, PATROL Console Server is considered to be running in overload mode, and the following will occur:

- Namespace variables are set to indicate that the PATROL Console Server is overloaded and the reason for the overload.

- A warning message describing the overloaded condition is written to the PATROL Console Server application log file or the dedicated audit log file, depending on your audit log configuration.

- Requests for the following actions are rejected:
  - opening existing profiles
  - creating new profiles
  - expanding profiles that are already open
  - running Event Manager and PATROL Agent queries when the filter is too generic

### Table 15  Overload Protection Options in the PATROL Console Server Configuration File (Part 2 of 2)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>maxProfilesOpen</td>
<td>defines the maximum number of profiles that can be open concurrently in either read-only or read/write mode and limits the total number of user sessions served by the PATROL Console Server. The default value is 0. A value of 0 indicates that no limit is defined.</td>
</tr>
<tr>
<td>maxReadWriteProfilesOpen</td>
<td>defines the maximum number of profiles that can be open concurrently in read/write mode. The default value is 0. A value of 0 indicates that no limit is defined.</td>
</tr>
<tr>
<td>maxObjectCount</td>
<td>defines the maximum number of managed objects allowed at any one time in all open profiles. The default value is 0. A value of 0 indicates that no limit is defined.</td>
</tr>
<tr>
<td>maxComputerCount</td>
<td>defines the maximum number of managed systems allowed at any one time in all open profiles. The default value is 0. A value of 0 indicates that no limit is defined.</td>
</tr>
</tbody>
</table>
When the PATROL Console Server transitions out of overload mode, the following will occur:

- Namespace variables are cleared to indicate that the PATROL Console Server is no longer overloaded and the reason.

- An informational message indicating that the PATROL Console Server is no longer overloaded is written to the PATROL Console Server application log file or the dedicated audit log file, depending on your audit log configuration.

- Requests to open existing profiles, create new profiles, and expand existing profiles are accepted.

The following example illustrates different settings for the various overload protection options.

For instructions about how to edit the `acfg_cserver_ServiceID.mof` file, see “Editing the Startup Configuration File” on page 82.

```plaintext
instance of Acfg_7_1_0_ConsoleServer
{
    cosPath = "AgentSetup/consoleserver";
    .
    .
    percentAddressSpaceLimit = 70;
    maxVirtualMemorySize = 5000000;
    maxProfilesOpen = 40;
    maxReadWriteProfilesOpen = 10;
    maxObjectCount = 5000000;
    maxComputerCount = 1000;
    .
    .
};
```
Collapsing All Top-Level Objects When Opening a Management Profile

You can control whether the PATROL Console Server collapses all top-level objects, such as PATROL Agents, when you open a management profile. When a management profile contains many top-level objects, collapsing them when opening the management profile provides the following benefits:

- The management profiles open faster.
- It is easier to scroll through the list of top-level objects.
- PATROL Central Operator does not spend time drawing objects that are not of immediate interest.
- The PATROL Console Server does not send data that is not of immediate interest.

Whether the PATROL Console Server collapses all top-level objects is determined by the following:

- the number of top-level objects in the management profile
- the value of the `collapseTopLevelChildrenLimit` option in the startup configuration file

If `collapseTopLevelChildrenLimit` is zero, or if it is greater than the number of top-level objects in the management profile, the previous state of each object (expanded or collapsed) is maintained when you open the management profile. Any top-level objects that were expanded when you last closed the management profile remain expanded, and any top-level objects that were collapsed remain collapsed.

However, if `collapseTopLevelChildrenLimit` is greater than zero but less than the number of top-level objects in the management profile, all top-level objects are automatically collapsed when the management profile is opened.

The default value of `collapseTopLevelChildrenLimit` is 10.
For example, if you want the PATROL Console Server to never collapse all top-level objects when opening a management profile, enter the line in bold in the startup configuration file.

For instructions about how to edit the `acfg_cserver_ServiceID.mof` file, see “Editing the Startup Configuration File” on page 82.

```c_cpp
// Console Server component configuration
instance of Acfg_7_1_0_ConsoleServer
{
    cosPath = "AgentSetup/consoleServer";
    enabled = true;
    collapseTopLevelChildrenLimit = 0;
};
```

**TIP**

To always collapse all top-level objects when opening all management profiles, set `collapseTopLevelChildrenLimit` to 1.

To never collapse all top-level objects when opening a management profile, set `collapseTopLevelChildrenLimit` to 0.

---

**EXAMPLE**

Suppose you have a management profile with 50 PATROL Agents directly under the PATROL Main Map.

- If `collapseTopLevelChildrenLimit` is set to 0, any PATROL Agents that were expanded remain expanded when you open the management profile.

- If `collapseTopLevelChildrenLimit` is set to 20, all of the PATROL Agents are collapsed when you open the management profile because there are more than 20 PATROL Agents.

- If `collapseTopLevelChildrenLimit` is set to 100, any PATROL Agents that were expanded remain expanded when you open the management profile because there are fewer than 100 PATROL Agents.
Limiting the Number of Computers Per Management Profile

You can limit the size of management profiles that a user can create. This ensures that management profiles do not exceed a predefined size.

You can define the number of PATROL Agents that a user can add to a management profile by setting the `privilegedUserMaxComputersPerProfile` option or the `unprivilegedUserMaxComputersPerProfile` option in the configuration file.

The option that you use depends on whether the user has the ALLOW MAXIMUM COMPUTERS PER PROFILE right. This right is granted to all of the PATROL user groups that also have the right to create profiles: patadm, patpop, and patop. However, you can always assign this right to individual users or groups that are configured in your system.

For users who have the ALLOW MAXIMUM COMPUTERS PER PROFILE right, set the `privilegedUserMaxComputersPerProfile` option in the configuration file.

For users who do not have the ALLOW MAXIMUM COMPUTERS PER PROFILE right, set the `unprivilegedUserMaxComputersPerProfile` option in the configuration file.

If both options are set and the user does not have the ALLOW MAXIMUM COMPUTERS PER PROFILE right, the value of `unprivilegedUserMaxComputersPerProfile` is capped with the value of `privilegedUserMaxComputersPerProfile`.

Table 16 on page 115 lists the options you can set in the configuration file to limit the number of computers per management profile. These options must be defined in the instance of `Acfg_7_1_0_ConsoleServer`.

For instructions about how to edit the `acfg_cserver_ServiceID.mof` file, see “Editing the Startup Configuration File” on page 82.
Configuring the Known Services Cache

The Known Services Cache (KSC) maintains a list of all the services in the RTserver cloud. Each time the PATROL Console Server starts, it checks the RTserver cloud for new services. It also continuously monitors the RTserver cloud for new services after the PATROL Console Server is started. For each service found, the KSC maintains a small list of attributes for that service, such as PATROL Agent version, operating system, ports, and host names. This information is used by PATROL Central Operator and PATROL Central Administration.

Table 17 on page 116 lists the KSC options in the configuration file. These options must be defined in the instance of Acfg_7_1_0_KSC.

For instructions about how to edit the acfg_cserver_ServiceID.mof file, see “Editing the Startup Configuration File” on page 82.
## Table 17  KSC Options in the PATROL Console Server Configuration File

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>autoScanServices</td>
<td>defines whether PATROL Agents are scanned to collect information</td>
</tr>
<tr>
<td></td>
<td>The default value is true.</td>
</tr>
<tr>
<td></td>
<td>If the value is set to true, scanning of PATROL Agents is enabled. If the</td>
</tr>
<tr>
<td></td>
<td>value is set to false, scanning of PATROL Agents is suppressed.</td>
</tr>
<tr>
<td></td>
<td>For more information about the <code>autoScanServices</code> option, see</td>
</tr>
<tr>
<td></td>
<td>“Controlling Scanning for PATROL Agent Information at Startup” on page 85.</td>
</tr>
<tr>
<td>kscRemoteUserContext</td>
<td>defines the virtual user name that is used to locate entries in the</td>
</tr>
<tr>
<td></td>
<td>impersonation table for authenticating with PATROL Agents</td>
</tr>
<tr>
<td></td>
<td>The default value is <code>ksc</code>. If the value is set to an empty string (&quot; &quot;),</td>
</tr>
<tr>
<td></td>
<td>the virtual user name is the user name for the PATROL Console Server</td>
</tr>
<tr>
<td></td>
<td>Account.</td>
</tr>
<tr>
<td></td>
<td>For more information about the <code>kscRemoteUserContext</code> option, see</td>
</tr>
<tr>
<td></td>
<td>“Controlling Scanning for PATROL Agent Information at Startup” on page 85.</td>
</tr>
<tr>
<td>autoSaveInterval</td>
<td>defines the frequency, in seconds, at which the KSC is saved to disk</td>
</tr>
<tr>
<td></td>
<td>The default value is 300, which is equivalent to 5 minutes.</td>
</tr>
<tr>
<td>cleanupInterval</td>
<td>defines the frequency, in seconds, at which the KSC checks for stale</td>
</tr>
<tr>
<td></td>
<td>services to remove from the cache</td>
</tr>
<tr>
<td></td>
<td>The default value is 86400, which is equivalent to 1 day.</td>
</tr>
<tr>
<td>retentionInterval</td>
<td>defines the period of time, in seconds, a service can remain stale before</td>
</tr>
<tr>
<td></td>
<td>it is removed from the cache</td>
</tr>
<tr>
<td></td>
<td>The default value is 604800, which is equivalent to 7 days.</td>
</tr>
</tbody>
</table>
Configuring the Number of Process Thread Pools

PATROL Console Server runs a pool of threads to handle requests that come from clients, such as PATROL Central consoles and command line utilities. These threads usually do not handle data coming from PATROL Agents, so the preconfigured value of 4 is sufficient most of the time.

You can control whether additional threads are created to handle concurrent requests from various sources and improve response time. This is beneficial in large environments and when you host 50–100 users.

**NOTE**

Creating additional threads is only applicable on computers with four or more processors, which are capable of true multiprocessing.

To configure the PATROL Console Server to create additional threads, set the value of the `threadPoolSize` option in the configuration file. It is recommended to increase the `threadPoolSize` option to 8 or 12.

The `threadPoolSize` option is found in the instance of `Acfg_8_0_0_Communication`.

For instructions about how to edit the `acfg_cserver_ServiceID.mof` file, see “Editing the Startup Configuration File” on page 82.

For example, if you want the PATROL Console Server to create 8 additional threads, configure the `acfg_cserver_ServiceID.mof` file as follows:

```plaintext
instance of Acfg_8_0_0_Communication
{
  threadPoolSize = 8;
};
```
Configuring Protocol Compression

Protocol compression improves performance and alleviates disconnection problems for remote consoles in a non-local wide area network (WAN) that are connected to the PATROL Console Server. This is especially true for large queries of events or managed objects.

**NOTE**

Protocol compression is available for PATROL Central Operator – Microsoft Windows Edition 7.5.00 and for command line utilities that are delivered with PATROL Console Server 7.5.00.

You can set the `compressionThreshold` option to define the minimum size of messages to which the PATROL Console Server applies compression.

To completely disable protocol compression in the PATROL Console Server, define the environment variable COS_COMPRESSIONTYPE with a value of 0. For more information about how to set environment variables for the PATROL Console Server, see “Setting Environment Variables for the PATROL Console Server” on page 139.

Table 18 lists the protocol compression option in the configuration file. This option is found in the instance of `Acfg_8_0_0_Communication`.

For instructions about how to edit the `acfg_cserver_ServiceID.mof` file, see “Editing the Startup Configuration File” on page 82.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>compressionThreshold</td>
<td>defines the minimum size, in bytes, of messages that are compressed</td>
</tr>
<tr>
<td></td>
<td>Only messages that are larger than the set threshold will be compressed.</td>
</tr>
<tr>
<td></td>
<td>A value of zero means that all eligible messages are compressed.</td>
</tr>
<tr>
<td></td>
<td>The default is 4096.</td>
</tr>
</tbody>
</table>
Saving ACLs and Impersonation Files at Regular Intervals

The PATROL Console Server saves ACL and impersonation file changes made in the PATROL Central Administration console at regular intervals. You can set the interval at which the data is saved by defining the *autoSaveInterval* option in the configuration file.

The *autoSaveInterval* option defines the frequency, in seconds, at which the ACL or impersonation files are saved to disk. The default value is 300, which is equivalent to 5 minutes.

To save ACL data, define the *autoSaveInterval* option in the instance of **Acfg_8_0_0_AccessControl**. For example, if you want the PATROL Console Server to save ACL data every 10 minutes, configure the **acfg_cserver_ServiceID.mof** file as follows:

```plaintext
instance of Acfg_8_0_0_AccessControl
{
    cosPath = "AgentSetup/ac";
    enabled = true;
    autoSaveInterval = 600
};
```

To save impersonation and alias data, define the *autoSaveInterval* option in the instance of **Acfg_8_0_0_Impersonate**. For example, if you want the PATROL Console Server to save impersonation data every 3 minutes, configure the **acfg_cserver_ServiceID.mof** file as follows:

```plaintext
instance of Acfg_8_0_0_Impersonate
{
    cosPath = "AgentSetup/impersonate";
    enabled = true;
    autoSaveInterval = 180
};
```

For instructions about how to edit the **acfg_cserver_ServiceID.mof** file, see “Editing the Startup Configuration File” on page 82.
Backing Up and Restoring the PATROL Console Server

This section describes how to manually back up and restore the PATROL Console Server. Other backup mechanisms include the online backup capability and the admincli command line utility. For more information about online backup, see “Online Backups” on page 125. For more information about the admincli command line utility, see “admincli” on page 180.

This section contains the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATROL Console Server Files to Back Up</td>
<td>page 121</td>
</tr>
<tr>
<td>Backing Up the PATROL Console Server</td>
<td>page 123</td>
</tr>
<tr>
<td>Restoring the PATROL Console Server</td>
<td>page 123</td>
</tr>
</tbody>
</table>

Once you have a full backup, you can refresh the existing backup. For more information about refreshing an existing backup, see “Refreshing a Backup Copy of the PATROL Console Server” on page 171.
PATROL Console Server Files to Back Up

This section lists the files to back up for the PATROL Console Server. For instructions on how to back up and restore these files, see “Backing Up the PATROL Console Server” on page 123 and “Restoring the PATROL Console Server” on page 123.

Files to Back Up for a Full Backup

Table 19 lists the directories to back up for a full backup of PATROL Console Server.

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows</td>
<td></td>
</tr>
<tr>
<td>%BMC_ROOT%</td>
<td>This directory contains all files installed and created by the PATROL Console Server and other PATROL services.</td>
</tr>
<tr>
<td>Unix</td>
<td></td>
</tr>
<tr>
<td>/etc/patrol.d</td>
<td>This directory contains security files for all PATROL products.</td>
</tr>
<tr>
<td>/opt/bmc (if $BMC_ROOT is not /opt/bmc)</td>
<td>This directory contains symbolic links that help setuid programs find dynamically loaded libxxxx.so.7.2 files.</td>
</tr>
<tr>
<td>$BMC_ROOT</td>
<td>This directory contains all files installed and created by the PATROL Console Server and other PATROL services.</td>
</tr>
</tbody>
</table>

Data and Configuration Files to Back Up

In the following situations, you may want to back up only the data and configuration files, rather than perform a full backup:

- to copy files to another machine
- to restore files if corrupted
- to restore the old files if permissions and right assignments caused errors
- if for any reason you want to have the minimal set of data required to make PATROL Console Server functional
Table 20 lists the directories that contain data and configuration files to back up.

**NOTE**
If you are running an advanced security level (levels 1 through 4), all files in these directories are signed. When you restore a file, ensure that the corresponding file with a .sgn extension is also restored. For more information about signed files, see the PATROL Security User Guide.

<table>
<thead>
<tr>
<th>Directory</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%BMC_ROOT%Patrol7\config\cserver</td>
<td>This directory contains information such as ACLs, impersonation database, configurations such as timeouts, site-specific settings for profiles, and the number of threads in the pool.</td>
</tr>
<tr>
<td>%BMC_ROOT%Patrol7\etc</td>
<td>This directory contains Secure Key Store files which store aliases (user.password mappings) used by impersonation support.</td>
</tr>
<tr>
<td>%BMC_ROOT%Patrol7\log\cserver</td>
<td>This directory contains several subdirectories, including subdirectories that contain all management profiles for the PATROL Console Server.</td>
</tr>
<tr>
<td>$BMC_ROOT/Patrol7/config/cserver</td>
<td>This directory contains information such as ACLs, impersonation database, configurations such as timeouts, site-specific settings for profiles, and the number of threads in the pool.</td>
</tr>
<tr>
<td>$BMC_ROOT/Patrol7/etc</td>
<td>This directory contains Secure Key Store files which store aliases (user.password mappings) used by impersonation support.</td>
</tr>
<tr>
<td>$BMC_ROOT%/Patrol7/log/cserver</td>
<td>This directory contains several subdirectories, including subdirectories that contain all management profiles for the PATROL Console Server.</td>
</tr>
</tbody>
</table>
Backing Up the PATROL Console Server

This task describes how to back up the PATROL Console Server.

To Back Up the PATROL Console Server

1. Stop the PATROL Console Server.

   **NOTE**

   Do not back up the files while PATROL Console Server is running. During shutdown, the PATROL Console Server writes the last known state to many of its files.

2. Back up the desired directories and files.

   See “PATROL Console Server Files to Back Up” on page 121.

3. Restart the PATROL Console Server.

Restoring the PATROL Console Server

This task describes how to restore the PATROL Console Server.

Before You Begin

You must know the service ID of the original PATROL Console Server. For more information, see “Choosing a Service ID for the PATROL Console Server” on page 64.

You must have a backup of the PATROL Console Server files. For more information, see “PATROL Console Server Files to Back Up” on page 121.

To Restore the PATROL Console Server

1. Install the same version of the PATROL Console Server as was used originally, including any patches.

   For more information, see Chapter 2, “Installing the PATROL Console Server and RTserver.”
2 Stop the PATROL Console Server.

**NOTE**
Do not restore the files while PATROL Console Server is running.

3 Replace the backup files to the appropriate directories.

For more information, see “PATROL Console Server Files to Back Up” on page 121.

4 Restart the PATROL Console Server so that it uses the same service ID as the original service ID.

Use one of the following methods:

- If the hostname of the new PATROL Console Server is different from the hostname of the original PATROL Console Server, then use the -id option.

  For more information about using the -id option, see “Command Line Arguments for the PATROL Console Server” on page 67.

- If the hostname of the new PATROL Console Server is different from the hostname of the original PATROL Console Server and you do not want to use the -id option, then rename the backed up files and directories to the new hostname using the following steps:

  1. Stop the old PATROL Console Server.

  2. Run the admin_copy utility to copy all of the needed files with new names into a new directory tree and to copy over the new PATROL Console Server computer.

     For more information about the admin_copy utility, see “admin_copy” on page 177.

  3. Start the new PATROL Console Server.

- If the hostname of the new PATROL Console Server is the same as the hostname of the original PATROL Console Server, then restart the new PATROL Console Server without the -id option.

  For more information about service IDs, see “Choosing a Service ID for the PATROL Console Server” on page 64.
Online Backups

You can run online backups to save your PATROL Console Server configuration data and management profiles. Online backups generate much of the same output as running the admin_copy command line utility. However, there are several advantages to running an online backup. With an online backup, you can:

- run the backup without shutting down the PATROL Console Server
- back up management profiles that are in use
- back up the PATROL Console Server configuration file, `acfg_cserver_serviceId.mof`
- back up the Known Services Cache

You can run online backups in two modes:

- Full backup—All PATROL Console Server configuration data and management profiles are backed up, regardless of whether they have changed since the last backup.
- Incremental backup—All PATROL Console Server configuration data is backed up; however, management profiles are backed up only if they have changed since the last backup.

All online backup activities are logged in the PATROL Console Server application log file and in the backup log file, `backup.log`, which is located at the root of the backup directory that you specified. The `backup.log` files are rotated in a similar fashion as the PATROL Console Server application log file, where you have access to the last five `backup.log` files in the same backup directory.

The `backup.log` file indicates whether the online backup is successful. If the backup is successful, the following message is displayed at the end of the log file:

```
INFORM: 3/16/2005 9:55:02 AM:::Backup operation completed successfully
```

**NOTE**

If you use a tool to monitor the backup directory to copy the results to an offline backup archive, check the status of the online backup before archiving the data.

There are two ways you can request online backups:

- Manual requests, which are run using the admincli command line utility. You can run manually requested online backups in full and incremental mode.
- Scheduled requests, which are run by configuring the configuration file. You can run scheduled online backups only in incremental mode.
Only one online backup operation can run at a given time. If another online backup request is received when another online backup operation is already running, that request is logged and skipped.

If shutdown of the PATROL Console Server is requested during an online backup operation, the online backup aborts as soon as possible and the PATROL Console Server shuts down. However, if an online backup of a large management profile has already started, shutdown of the PATROL Console Server is delayed until the online backup of the file is completed.

### Manually Requesting an Online Backup

Only users with the ALLOW THE USER TO INITIATE ONLINE BACKUPS right can manually request an online backup. By default, only the patadm group has this right. However, you can grant this right to individual users to meet your requirements.

For more information about how to manually request an online backup with the admincli command line utility, see “Options for Manually Requesting an Online Backup” on page 191.

### Scheduling Online Backups

To set up a regularly scheduled online backup, you must make changes in the PATROL Console Server configuration file, `acfg_cserver_ServiceID.mof`. Therefore, only PATROL administrators with access to physical directory structures of the PATROL Console Server can schedule online backups.

PATROL Console Server supports two independent backup areas with two independent backup schedules. You have the choice of specifying a primary online backup, a secondary online backup, or both.

Table 21 on page 127 lists the configuration options that are directly related to scheduling online backups. These options are found in the instance of `Acfg_8_0_0_Generic`. By default, there is no online backup configured, and all schedules and backup directory definitions are empty.

For instructions about how to edit the `acfg_cserver_ServiceID.mof` file, see “Editing the Startup Configuration File” on page 82.
### Table 21  Online Backup Options in the PATROL Console Server Configuration File

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>primaryBackupPath</td>
<td>defines the primary backup path</td>
</tr>
<tr>
<td></td>
<td>The directory you specify must exist and should be writable by the default user account of the PATROL Console Server.</td>
</tr>
<tr>
<td></td>
<td>The PATROL Console Server does not create a top-level directory for backup, but it creates any subdirectories needed under the top-level directory.</td>
</tr>
<tr>
<td></td>
<td>If you do not specify a backup directory when manually requesting an online backup, the primaryBackupPath is used.</td>
</tr>
<tr>
<td></td>
<td>The default is an empty string.</td>
</tr>
<tr>
<td>secondaryBackupPath</td>
<td>defines the secondary backup path</td>
</tr>
<tr>
<td></td>
<td>The directory you specify must exist and should be writable by the default user account of the PATROL Console Server.</td>
</tr>
<tr>
<td></td>
<td>The PATROL Console Server does not create a top-level directory for backup, but it creates any subdirectories needed under the top-level directory.</td>
</tr>
<tr>
<td></td>
<td>The default is an empty string.</td>
</tr>
<tr>
<td>primaryBackupSchedule</td>
<td>defines the job specification string for the primary backup</td>
</tr>
<tr>
<td></td>
<td>For information about specifying the job specification string, see “The Job Specification String” on page 128.</td>
</tr>
<tr>
<td></td>
<td>The default is an empty string.</td>
</tr>
<tr>
<td>secondaryBackupSchedule</td>
<td>defines the job specification string for the secondary backup</td>
</tr>
<tr>
<td></td>
<td>For information about specifying the job specification string, see “The Job Specification String” on page 128.</td>
</tr>
<tr>
<td></td>
<td>The default is an empty string.</td>
</tr>
</tbody>
</table>
For example, if you want to schedule a backup to run every morning at 1 AM and your backup directory is `/my_backup_filesystem/Patrol7/primaryBackup/`, configure the `acfg_cserver_Service1D.mof` file as follows:

```csharp
instance of Acfg_B_0_0_Generic {
    primaryBackupPath = "/my_backup_filesystem/Patrol7/
    primaryBackup/";

    // every morning at 1 am;
    primaryBackupSchedule = "+ 0100 * * * * * *"
};
```

**The Job Specification String**

Job scheduling parameters are supplied to the Scheduler through a schedule specification string, which is used by the Scheduler to control the job submitted to the Scheduler queue. It specifies the active or inactive state of the job run times; that is, you can specify that the job run during the times specified or that the job does not run during the specified times. The string also specifies start and end times so that you can control when a job is run.

You can combine schedule strings into simple but relatively powerful control constructs by logically adding multiple specification strings.

The string consists of nine parts, or fields:

```
Active Start End Interval Weekday Month Day Number # Comments
```

The fields must be separated by at least one space or tab. Everything from the pound sign (#) to the end of the line is treated as a comment. An asterisk (*) in a field selects all possible values or acts as a placeholder, depending on the context.
Table 22 lists the fields used to specify the job specification string.

Table 22  Fields for Specifying a Job Specification String (Part 1 of 3)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Active | specifies whether to run or *not* run an online backup during the times specified in the string  
This field must contain one of the following single characters:  
- + (plus sign)—indicates that the remaining time specification values be interpreted as times to run the job.  
- - (minus sign)—indicates that the remaining time specification values be interpreted as times *not* to run the job. |
| Start  | specifies when to start the online backup  
The start time is specified in a 24-hour format with no seconds: hhmm.  
If you specify *, execution of the online backup begins as soon as possible and is treated as a one-time job.  
To run the online backup at the same interval all the time, specify a start time of 0000 and an end time of 2400. |
| End    | specifies when to end the online backup  
To run the online backup at the same interval all the time, specify a start time of 0000 and an end time of 2400. |
| Interval | specifies the interval to run the online backup  
The interval is specified as hh:mm:ss, mm:ss, or ss. For example, all of the following specify a period of two minutes: 00:02:00, 2:00, and 120.  
Leading zeroes are not required.  
If you specify *, a one-time job is run. To run a job periodically, you must specify a time value. |
### Table 22  Fields for Specifying a Job Specification String (Part 2 of 3)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
</table>
| Weekday | specifies the day of the week to run the online backup  

The days of the week can be specified numerically, where 0 represents Sunday and 6 represents Saturday, or by using the locale-specific character representation of the weekday, such as Fri.

You can combine multiple weekdays on a single line by specifying an inclusive range, such as Mon-Fri, or by specifying a list of values, such as Mon, Wed, Fri.

You can also combine ranges and individual values. For example, Mon-Wed, Fri.

| Month | specifies the month to run the online backup  

The months can be specified numerically, where 1 represents January and 12 represents December, or by using the locale-specific character representation of the month, such as Jan.

You can combine multiple months on a single line by specifying an inclusive range, such as Jan-Dec, or by specifying a list of values, such as Mar, Jun, Sep, Dec.

You can also combine ranges and individual values. For example, Jan-Mar, Jun, Sep-Dec.

| Day | specifies the day of the month to run the online backup  

The range of valid days are 1 to the last day of the month. Note that specifying Feb 31 may result in an error or failure to run.

To specify the last day of the month, you can use -1. You can also use -2 to specify the second to last day of the month, and so on.

You can combine multiple days on a single line by specifying an inclusive range, such as 1-10, or by specifying a list of values, such as 15, -1.

You can also combine ranges and individual values. For example, 1-10, 15, 20-31.
Table 22  Fields for Specifying a Job Specification String (Part 3 of 3)

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number</td>
<td>specifies the number of times to run the online backup</td>
</tr>
<tr>
<td></td>
<td>This field is not used for one-time online backups.</td>
</tr>
<tr>
<td>Comments</td>
<td>allows the user to add comments for the job specification string</td>
</tr>
<tr>
<td></td>
<td>Everything from the pound sign (#) to the end of the line is treated as a comment.</td>
</tr>
</tbody>
</table>

Table 23 lists valid and invalid job specification string values.

Table 23  Valid and Invalid Job Specification String Values (Part 1 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Valid Values</th>
<th>Invalid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active</td>
<td>+</td>
<td>Any alphanumeric character</td>
</tr>
<tr>
<td></td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Start/End</td>
<td>0000-2400</td>
<td>*</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&lt;0000</td>
</tr>
<tr>
<td></td>
<td></td>
<td>&gt;2400</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-0000 - -2400</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0-9</td>
</tr>
<tr>
<td></td>
<td></td>
<td>00-99</td>
</tr>
<tr>
<td></td>
<td></td>
<td>000-999</td>
</tr>
<tr>
<td>Interval</td>
<td>hh:mm:ss</td>
<td>-seconds</td>
</tr>
<tr>
<td></td>
<td>mm:s</td>
<td></td>
</tr>
<tr>
<td></td>
<td>ss</td>
<td></td>
</tr>
<tr>
<td></td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>
Table 23  Valid and Invalid Job Specification String Values (Part 2 of 2)

<table>
<thead>
<tr>
<th>Field</th>
<th>Valid Values</th>
<th>Invalid Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Weekday</td>
<td>0-6</td>
<td>-0 -6</td>
</tr>
<tr>
<td></td>
<td>Sun, Mon, Tue, Wed, Thu, Fri, Sat</td>
<td>&gt;6</td>
</tr>
<tr>
<td></td>
<td>Full names</td>
<td>&lt;0</td>
</tr>
<tr>
<td></td>
<td>Locale-specific names</td>
<td>M, Tu, W, Th, T, F, Sa, Su, S, Thur, Tues</td>
</tr>
<tr>
<td>Month</td>
<td>1-12</td>
<td>&lt;1</td>
</tr>
<tr>
<td></td>
<td>Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sept, Oct, Nov, Dec</td>
<td>&gt;12</td>
</tr>
<tr>
<td></td>
<td>Full names</td>
<td>-1 - -12</td>
</tr>
<tr>
<td></td>
<td>Locale-specific names</td>
<td></td>
</tr>
<tr>
<td>Day</td>
<td>-1--31</td>
<td>&gt;31</td>
</tr>
<tr>
<td></td>
<td>1-31</td>
<td>14--14 (will fail on the 16th of any month that has 31 days)</td>
</tr>
<tr>
<td></td>
<td>13--13</td>
<td></td>
</tr>
<tr>
<td></td>
<td>*</td>
<td></td>
</tr>
<tr>
<td>Number</td>
<td>0</td>
<td>&lt;=0</td>
</tr>
<tr>
<td></td>
<td>*</td>
<td></td>
</tr>
</tbody>
</table>

**Examples of Valid Job Specification Strings**

The following example runs an online backup once at noon. If the current time is before noon, the online backup is run at noon; however, if the current time is after noon, the online backup runs at noon the following day. After the run is complete, the job is removed from the queue and will not be rescheduled.

```
+1200 * * * * * 1 # One-time job run at noon
```
The following example runs an online backup at 1:00 AM on the fifteenth and last days of every month for one year. After the run is complete for each month of the year, the job is removed from the queue and will not be rescheduled.

```
+ 0100 * * * 15,-1 1 # Every month on the 15th and last day
```

The following example runs an online backup as soon as possible, four times, every 10 seconds. After the fourth run is complete, the job is removed from the queue and will not be rescheduled.

```
+ * * 10 * * * 4 # Run 4 times ASAP every 10 seconds
```

The following example runs an online backup as soon as possible, three times, at one-hour intervals. After the third run is complete, the job is removed from the queue and will not be rescheduled.

```
+ * * 1:00:00 * * * 3 # Run 3 times ASAP every hour
```

The following example runs an online backup three times, at one-hour intervals, starting at noon. If the current time is before noon, the online backup is run at noon; however, if the current time is after noon, the online backup runs at noon the following day. After the third run is complete, the job is removed from the queue and will not be rescheduled. It is not possible to schedule a periodic job that has a fixed number of executions.

```
+ 1200 * 1:00:00 * * * 3 # Modify to start running at noon
```

The following example runs an online backup at 10-minute intervals every Saturday and on the first and fifteenth day of each month.

```
+ 0000 2400 600 Sat * * * # Every Sat at 10-minute intervals
+ 0000 2400 600 * * 1,15 * # Every 1st and 15th at 10-minute intervals
```
The following example runs an online backup on the first and fifteenth day of each month if those days fall on a Saturday.

```
+ 0000 2400 600 Sat * 1,15 * # Every 1st and 15th if on Sat
```

Either of the following examples runs a one-time online backup for immediate execution.

```
+ * * * * * * * # One-time ASAP
```

OR

```
+ * * * * * * 1 # One-time ASAP
```

The following example runs a one-time online backup for delayed execution on May 1 at 6:00 AM.

```
+ 0600 * * * May 1 * # Don't run until May 1 at 6AM
```

The following example runs an online backup across midnight on December 31.

```
+ 2330 2400 30 Fri Dec 31 * # Run across midnight
+ 0000 0030 30 Sat Jan 1 *
```
Examples of Invalid Job Specification Strings

The following example tries to specify both periodic and non-periodic schedules for an online backup.

```
+ 1200 * * * * * 1
+ 0000 2400 60 * * *
```

The following example does not specify an interval for the active period of a periodic online backup.

```
+ 0800 1600 * * * *
```

The following example specifies multiple days and times for a non-periodic online backup.

```
+ 0400 * * * May 1,15 *
```

The following example tries to run an online backup across midnight in a single line.

```
+ 2300 0030 30 Fri Dec 31
```
Changing the Service ID of the PATROL Console Server

The PATROL Console Server embeds its service ID in the names of the following:

- directory for management profiles
- access control database that stores the assignment of rights and permissions
- impersonation database that holds alias and impersonation tables
- PATROL Console Server startup configuration file
- secure key store database that contains impersonation aliases, user IDs, and encrypted passwords

In order to retain the above information when changing the service ID of the PATROL Console Server, you must change the names of the files and directories. You can use the admin_copy utility to perform this task.

For more information about the service ID, see “Choosing a Service ID for the PATROL Console Server” on page 64.

To Change the Service ID of the PATROL Console Server

1. Stop the PATROL Console Server.

2. Use the admin_copy utility to copy all of the named configuration files from the old service ID to the new service ID.

   For the -out argument, use the full path name to your PATROL7 directory (%PATROL_ROOT% for Windows and $PATROL_ROOT for Unix). Include the -profiles argument to copy management profiles as well as the security configuration information. The following example assumes that the PATROL Console Server is installed on a machine named mycserver.

   ![Example](example.png)

3. On Windows, remove the default PATROL Console Server:

   ![Example](example.png)
4 On Windows, you must also change the Windows service installation to use the new id:

   C> cserver -install -id newRTServiceID

5 Restart the PATROL Console Server with the new service ID.

   On Windows, you can start the PATROL Console Server from the Windows Services Control Panel.

   On Unix, you must start the PATROL Console Server as:

   % start_cserver.sh -id newRTServiceId

   For more information about command line arguments for starting the PATROL Console Server, see “Command Line Arguments for the PATROL Console Server” on page 67.
Moving the PATROL Console Server

You can move the PATROL Console Server to a different computer by backing it up and then restoring it to the new computer. You might also have to resolve any changes to the service ID of the PATROL Console Server. You can use the admin_copy utility to perform these tasks.

To Move the PATROL Console Server

1. Stop the PATROL Console Server on the original computer.
2. On the new computer, install the same version of the PATROL Console Server as on the original computer, including any patches.
3. Stop the PATROL Console Server on the new computer.
4. Run the admin_copy utility on the original computer to copy all of the named configuration files to the new computer using the new service ID.

   For the -out argument, use the full path name to the PATROL7 directory on the new computer (%PATROL_ROOT% for Windows and $PATROL_ROOT$ for Unix). Include the -profiles argument to copy management profiles as well as the security configuration information.

   - C> admin_copy.cmd -src_c originalComputerRTserviceID -dst_c newComputerRTserviceID -out %PATROL_ROOT%onNewComputer -profiles (Windows)

   - %admin_copy.sh -src_c originalComputerRTserviceID -dst_c newComputerRTserviceID -out $PATROL_ROOT$onNewComputer -profiles (Unix)

   For more information about the admin_copy utility, see “admin_copy” on page 177.

5. Start the new PATROL Console Server, using the appropriate service ID.

   **NOTE**

   If you keep the service ID of the PATROL Console Server the same across the move, do not restart the original PATROL Console Server. You cannot have two PATROL Console Servers with the same service ID running at the same time.
Setting Environment Variables for the PATROL Console Server

The PATROL Console Server uses a variety of environment variables. For a list of environment variables, see “Environment Variables Used by PATROL Console Server” on page 294.

You can change the values that the PATROL Console Server uses for environment variables by using the `patrol.conf` file, which is located in the following directory:

- `%BMC_ROOT%\common\patrol.d` (Windows)
- `/etc/patrol.d` (Unix)

Any environment variables that you set in the `patrol.conf` file override the environment specified on the command line or system and are used by the PATROL Console Server as well as any programs spawned by the PATROL Console Server. The variables are read and set before the PATROL Console Server performs any other functions.

Following is an example of the `[CSERVER]` section of the `patrol.conf` file. In the following example, the RTSERVERS environment variable is set to `tcp:localhost:2059`.

```
[CSERVER]

set_environment
   RTSERVERS=tcp:localhost:2059
   PATROL_ROOT_CONFIG=/opt/share/Patrol7/config
end

defaultAccount = patrol:FB0A195D60626300
```

You can specify multiple environment variables in the `patrol.conf` file. To remove an environment variable from the current process context, specify an empty string (""") for that environment variable.

You can also edit the `patrol.conf` file by using the confManager utility. For more information about confManager, see “confManager” on page 194.
Administering Management Profiles from the PATROL Console Server

By storing management profiles on the PATROL Console Server, users can access their management profiles from any instance of PATROL Central Operator. All the data related to a user session is collected and saved on the PATROL Console Server in one or more data files associated with a common name (the management profile).

Each management profile has an owner and can be opened either as read-only (from one or more PATROL consoles at the same time) or as read/write (from only one PATROL console). By default, a management profile can be open for read/write by only the owner of a management profile, PATROL administrators, or users with the FORCE_PROFILE_CLOSE_PRIV right.

PATROL Console Server supports multiple open management profiles at the same time from a single multi-threaded process. Therefore, if a management profile is open as read-only, other users can also open it as read-only. In addition, if a management profile is already open for read/write, other users can open it as read-only.

To force closed a management profile that is already open in read/write, you must be the owner of the management profile or have read/write access to the management profile. You must also have both of the following rights:

- FORCE CLOSING OF THE MANAGEMENT PROFILE THAT IS IN USE
- CREATE AND DESTROY MANAGEMENT PROFILE

The user who had the management profile open in read/write is notified that the management profile was forced closed. They may then reopen it as read-only.

You can manually copy management profiles from one PATROL Console Server to another as long as both PATROL Console Servers are stopped while the management profiles are copied. Management profiles are platform independent; for example, a management profile created on a Windows server will also work on Linux or Solaris servers. To copy management profiles from a running PATROL Console Server, use the online backup capability. However, copying profiles into a running PATROL Console Server is not supported.

When a given user logs out or the connection to the PATROL Console Server is dropped, the management profile is saved and closed. All management profiles opened as read/write are also saved, by default, every five minutes.
For information about using management profiles, including a list of what is stored in a management profile, see the following documents:

- PATROL Central Operator – Microsoft Windows Edition Getting Started
- PATROL Central Operator – Microsoft Windows Edition Help
- PATROL Central Operator – Web Edition Getting Started
- PATROL Central Operator – Web Edition Help

### Creating New Default Managed System Queries, Event Filters, and Chart Schemes

Management profiles contain both system and user-defined managed system queries, event filters, and chart schemes.

Normally, a new management profile contains only system-defined items by default. However, you can configure the PATROL Console Server to automatically add selected user-defined items to all new management profiles.

#### To Create a User-defined Default Managed System Query, Event Filter, or Chart Scheme

1. Create a user-defined managed system query, event filter, or chart scheme with the properties that you want in PATROL Central Operator.

   For more information, see the PATROL Central Operator – Microsoft Windows Edition online Help or the PATROL Central Operator – Web Edition online Help.


3. Stop the PATROL Console Server.

4. Copy the exported .mof file from the PATROL Central Operator computer to the following directory on the PATROL Console Server:

   - %PATROL_ROOT%\lib\mof\cserver (Windows)
   - $PATROL_ROOT/lib/mof/cserver (Unix)
5 Open the `acfg_cserver_ConsoleServerID.mof` configuration file in a text editor.

This file is located in the following directory:

- `%PATROL_ROOT%\config\cserver\` (Windows)
- `$PATROL_ROOT/config/cserver/` (Unix)

6 In the `acfg_cserver_ConsoleServerID.mof` file, modify the attribute of `newLayoutPreloadList` by adding the name of the exported `.mof` file.

7 Restart the PATROL Console Server.

The next time a new management profile is created, it will include the default system query, event filter, or chart scheme.

**NOTE**

If you use a security level greater than basic security, all of the files on the PATROL Console Server are signed. You must sign any new files and re-sign any edited files. For more information, see the *PATROL Security User Guide*.

---

**Viewing All Events for a PATROL Agent**

Users can view all events from a given PATROL Agent, including events for objects that are not part of the management profile.

In order to filter for all events for a given PATROL Agent, specify the wildcard character (*) as the event origin in the event filter. You must also have the ALLOW THE USER TO QUERY ALL EVENTS WITHOUT LIMITING TO LOADED KNOWLEDGE MODULES right. By default, only members of the patadm group have this right.

If you leave the event origin blank in an event filter in PATROL Central Operator, the filter still returns events for only the applications in the current management profile.
Running Commands on the PATROL Console Server From PATROL Central Operator

The PATROL Console Server provides a set of built-in commands that you can run from PATROL Central Operator.

You can run these built-in commands from any system output window in PATROL Central Operator.

Table 24 lists the built-in commands that you can run from PATROL Central Operator.

Table 24  PATROL Console Server Built-in Commands (Part 1 of 2)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%CS ?</td>
<td>This command lists the available commands in the given session. It is identical to the HELP command.</td>
</tr>
<tr>
<td>%CS HELP</td>
<td>This command lists the available commands in the given session. It is identical to the ? command.</td>
</tr>
<tr>
<td>%CS LIST COMPUTER</td>
<td>This command lists all the managed systems in the management profile.</td>
</tr>
<tr>
<td>%CS LIST METHODS</td>
<td>This command lists all action methods in the management profile.</td>
</tr>
<tr>
<td>%CS LIST QUERIES</td>
<td>This command lists all named managed system queries in the management profile.</td>
</tr>
<tr>
<td>%CS LIST LEMFILTERS</td>
<td>This command lists all named event filters in the management profile.</td>
</tr>
<tr>
<td>%CS LIST SESSIONS</td>
<td>This command lists all users currently logged on to the PATROL Console Server.</td>
</tr>
<tr>
<td>%CS ECHO CS_OS_ARCH</td>
<td>This command echoes the operating system architecture of the PATROL Console Server.</td>
</tr>
<tr>
<td>%CS ECHO CS_VER</td>
<td>This command echoes the version string of the PATROL Console Server.</td>
</tr>
<tr>
<td>%CS ECHO CS_REVISION</td>
<td>This command echoes the revision string of the PATROL Console Server.</td>
</tr>
<tr>
<td>%CS ECHO CS_BACKEND_REV</td>
<td>This command echoes the backend string of the PATROL Console Server.</td>
</tr>
</tbody>
</table>

This argument was introduced in version 7.2.30 of the PATROL Console Server.
### Table 24  PATROL Console Server Built-in Commands (Part 2 of 2)

<table>
<thead>
<tr>
<th>Command</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>%CS ECHO CONSOLE_OS_ARCH</td>
<td>This command echoes the operating system architecture of the PATROL console.</td>
</tr>
<tr>
<td>%CS ECHO CONSOLE_LOCALE</td>
<td>This command echoes the selected locale of the PATROL console.</td>
</tr>
<tr>
<td>%CS ECHO CONSOLE_VER</td>
<td>This command echoes the name and version string of the PATROL console.</td>
</tr>
<tr>
<td>%CS COMMIT</td>
<td>This command immediately saves the current management profile. The management profile must be open in read/write mode.</td>
</tr>
<tr>
<td>%CS SET OWNER</td>
<td>This command changes the owner of the management profile.</td>
</tr>
<tr>
<td>%CS RUN methodName</td>
<td>This command runs the action method that is specified by <code>methodName</code>.</td>
</tr>
<tr>
<td>%CS TRACE</td>
<td>This command returns the current trace level for the management profile.</td>
</tr>
<tr>
<td>%CS TRACE START level</td>
<td>This commands starts and stops trace for the management profile. If <code>level</code> &gt; 0, the command starts trace for the management profile, if it is not already started. If it is already started, it changes the trace level. If <code>level</code> = 0, the command stops trace for the management profile. The management profile must be open in read/write mode.</td>
</tr>
</tbody>
</table>

---

Running Commands on the PATROL Console Server From PATROL Central Operator
PATROL Console Server Replication

This chapter provides information about establishing failover for the PATROL Console Server by using either of the following methods:

- manual failover of the PATROL Console Server on any platform
- automatic failover for the PATROL Console Server on Windows by using it in a Windows 2000 or Windows 2003 cluster environment

This chapter also provides information on using the admin_copy utility for replication of the PATROL Console Server.

**NOTE**
The instructions in this chapter relating to clusters are specific to Windows 2000 and Windows 2003 clusters.

This chapter contains the following topics:

- Manual Failover of the PATROL Console Server .................................................. 147
- Cluster Management Software and Failover ......................................................... 148
  - Failover in a Cluster Environment ................................................................. 148
  - Failover Behavior of the PATROL Console Server in a Cluster ..................... 149
- Setting Up the PATROL Console Server and RTserver in a Windows Cluster Environment ................................................................. 150
- Installing the PATROL Console Server and RTserver on Each Computer in the Cluster ................................................................. 151
- Pointing the PATROL Console Server on Each Computer to Directories on the Shared Drive ................................................................. 152
- Moving RTserver Configuration Directory to the Shared Drive ....................... 155
- Setting Up the PATROL Console Server and RTserver in the Cluster ............... 156
Example Scenarios for Cluster Failover ........................................... 160
  New PATROL Console Server in a Cluster .................................. 160
  New PATROL Console Server and RTserver in a Cluster ............... 162
  New PATROL Console Server in a Cluster (Command Line Example)  164
  New PATROL Console Server and RTserver in a Cluster
  (Command Line Example) .................................................. 168
Replicating the PATROL Console Server .................................... 171
  Refreshing a Backup Copy of the PATROL Console Server .......... 171
  Creating a Copy For Other Instances of the PATROL Console Server 173
Manual Failover of the PATROL Console Server

You can establish manual failover to handle the relocation of computer resources. For example, if the first PATROL Console Server fails, then a second PATROL Console Server can be brought online that contains the same information as the first PATROL Console Server, including the management profiles and the service ID.

The completeness and degree of synchronization after manual failover depend upon how up-to-date the backup system is compared to the original system. For this reason, BMC Software recommends that you back up your PATROL Console Server data on a regular basis. For information about backing up PATROL Console Server data, see “Backing Up and Restoring the PATROL Console Server” on page 120.

This task describes how to manually fail over the PATROL Console Server.

Establishing manual failover of the PATROL Console Server essentially involves restoring the files of the original PATROL Console Server onto a new PATROL Console Server, and then starting the new PATROL Console Server with the service ID of the original PATROL Console Server.

**NOTE**

Note that this is a manual process to be performed after the original PATROL Console Server has failed. This process applies to all platforms.

**Before You Begin**

You must know the service ID of the original PATROL Console Server. For more information, see “Choosing a Service ID for the PATROL Console Server” on page 64.

You must have a backup of the PATROL Console Server files or access to the files on the original PATROL Console Server. For more information, see “Backing Up the PATROL Console Server” on page 123.

**To Manually Fail Over the PATROL Console Server**

1. If the backup PATROL Console Server is running, stop it.

2. Copy the necessary files to the backup PATROL Console Server.

   If the source computer is available, stop the PATROL Console Server and copy the files from it. Otherwise, use the most recent backups.
3 Restart the backup PATROL Console Server with the -id command line argument and the service ID of the original PATROL Console Server so that the new PATROL Console Server starts with the identity of the original one.

- On Unix, append the -id command line argument at the command line when starting the PATROL Console Server.

- On Windows, add the -id command line argument as a startup parameter of the PATROL Console Server service.

For more information about service IDs, see “Choosing a Service ID for the PATROL Console Server” on page 64. For more information about using command line arguments, see “Command Line Arguments for the PATROL Console Server” on page 67.

Cluster Management Software and Failover

A cluster is a collection of two or more computers that work together and share common disk storage. A cluster is controlled by cluster management software that operates within the cluster. In Windows, a cluster is managed through the Windows Cluster Management Software.

For more information about clusters, including details on how to configure your cluster, see your cluster documentation.

Failover in a Cluster Environment

This section provides a brief introduction to how a Windows cluster handles failover of an application from one computer to another in the cluster.

An application in a cluster environment runs as part of a resource group that contains related resources, such as a shared drive or another application. For example, if an application requires access to a shared drive or works in conjunction with another application, the group for the application also contains those resources. You configure the group in the cluster management software, and failover occurs at this group level.
Failover ensures that an application remains available for use, even if it stops running on one computer. When failover occurs, execution of the application and any other resources in the group moves from one computer in the cluster to another. Execution of an application might need to be moved to a different computer for many reasons, such as:

- application failure on the first computer
- hardware or software failure on the first computer
- load balancing
- scheduled maintenance

For more information about failover in a cluster, see your cluster documentation.

**Failover Behavior of the PATROL Console Server in a Cluster**

The following sequence of events describes how the PATROL Console Server runs in a Windows cluster environment and how failover occurs.

1. **Request:** A user requests that the group for the PATROL Console Server be brought online. A request can occur at anytime, including at startup of the cluster.

2. **Startup:** The cluster management software starts the PATROL Console Server on its primary computer, as well as any other resources in the same group.

3. **Run:** The PATROL Console Server runs on the primary computer, reading and writing data, such as management profiles and impersonation data, to a shared drive.

4. **Service Interrupted:** The PATROL Console Server application on the primary computer stops. For example, it might be unable to access resources that it requires to run, there might be a hardware failure on the primary computer, or the system administrator might want to force failover.

5. **Shutdown:** The cluster management software shuts down the PATROL Console Server on the primary computer, as well as other resources in the same group.

6. **Failover:** The cluster management software starts the PATROL Console Server on a secondary computer, along with other resources in the same group.

7. **Resume:** The PATROL Console Server runs on the secondary computer, reading and writing data to the same shared drive used by the primary computer.
Setting Up the PATROL Console Server and RTserver in a Windows Cluster Environment

You can use cluster failover for both the PATROL Console Server and the RTserver, or for just the PATROL Console Server.

You can use either of the following methods:

- Use cluster failover for the PATROL Console Server but not for the RTserver

  This method is simplest if you want to use an RTserver outside of the cluster. You can still achieve failover for the RTserver by using an RTserver cloud.

- Use cluster failover for both the PATROL Console Server and the RTserver

  This method is recommended because it ensures that the RTserver and the PATROL Console Server always run on the same physical computer. This method also provides failover for the RTserver if you do not have an RTserver cloud.

This section describes the following tasks for setting up the PATROL Console Server and RTserver, or just the PATROL Console Server for failover in a Windows cluster environment.

1. Install the PATROL Console Server (and RTserver, if appropriate) on each computer in the cluster (see page 151).

2. Point the PATROL Console Server on each computer to the same directories on a shared drive (see page 152).

3. If you are including the RTserver in cluster failover, move the RTserver configuration directory on each computer to the same shared drive (see page 155).

4. Set up the PATROL Console Server (and RTserver, if appropriate) in the cluster management software (see page 156).
Installing the PATROL Console Server and RTserver on Each Computer in the Cluster

You install the PATROL Console Server on each computer in the cluster. If you include the RTserver in cluster failover, you also install it to each computer at the same time.

Follow the regular installation instructions for each computer in the cluster, with the following additional guidelines.

- Use the same local installation directory for all installations.

  For example, if you install to the C:\Program Files\BMC Software on one computer in the cluster, use the same directory path for all computers in the cluster.

  Although the PATROL Console Server and RTserver do not need to use the same directory names on each local computer, having the same names is recommended to avoid confusion.

- Use the same user name and password for the PATROL Console Server account on each computer.

  You can use a domain account or separate local accounts with the same user name and password. Having the same user name and password enables you to log on as the PATROL Console Server account no matter where the PATROL Console Server is running.

- Choose the option to not start the PATROL Console Server after the installation. If you also install the RTserver, also choose the option to not start the RTserver after the installation.

  If you do start the PATROL Console Server or RTserver, stop it before proceeding with the next task.

- On Windows, change the PATROL Console Server service that is created by the installation to manual startup instead of automatic startup. If you also installed the RTserver, also change the SmartSockets RTserver service to manual startup.
Because the startup of the service in a cluster environment is controlled by the cluster management software, the service should not be started automatically by the operating system.

**NOTE**
You should keep all of the installations as similar as possible, to minimize any differences when running the PATROL Console Server on different computers. If you patch or upgrade the PATROL Console Server, patch or upgrade all installations in the cluster. If you change the PATROL Console Server account, change it on all installations in the cluster.

### Pointing the PATROL Console Server on Each Computer to Directories on the Shared Drive

You must point each PATROL Console Server installation to directories for configuration and data files on a shared drive. This shared drive will be a resource of the group for the PATROL Console Server in the cluster management software (see “Configuring the Resource Group” on page 157).

This section contains the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATROL Console Server Directories for the Shared Drive and their Environment Variables</td>
<td>page 153</td>
</tr>
<tr>
<td>Setting Environment Variables for the PATROL Console Server By Using the patrol.conf File</td>
<td>page 154</td>
</tr>
<tr>
<td>Verifying that Each PATROL Console Server Uses the Shared Directories</td>
<td>page 154</td>
</tr>
</tbody>
</table>
PATROL Console Server Directories for the Shared Drive and their Environment Variables

You use environment variables to make the PATROL Console Server on each computer use the same directories on a shared drive. Table 25 lists the environment variables and the corresponding directories.

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATROL_ROOT_CONFIG</td>
<td>This environment variable redirects the %PATROL_ROOT%\config directory. The PATROL Console Server uses configuration files in the cserver subdirectory of this directory.</td>
</tr>
<tr>
<td>PATROL_ROOT_TMP</td>
<td>This environment variable redirects the %PATROL_ROOT%\tmp directory. The PATROL Console Server creates temporary files and lock files in the cserver subdirectory of this directory.</td>
</tr>
<tr>
<td>PATROL_ROOT_ETC</td>
<td>This environment variable redirects the %PATROL_ROOT%\etc directory, where security sensitive data is stored. This directory holds SKS (Secure Key Store) databases of encrypted passwords used by the impersonation table.</td>
</tr>
<tr>
<td>PATROL_ROOT_LOG</td>
<td>This environment variable redirects the %PATROL_ROOT%\log directory. The PATROL Console Server creates management profiles, runtime logs, and other internal databases in the cserver subdirectory of this directory.</td>
</tr>
<tr>
<td>PATROL_ROOT_LAYOUT</td>
<td>This environment variable redirects the %PATROL_ROOT_LOG%\cserver\layout directory, where management profiles are stored. If you redirect the PATROL_ROOT_LOG directory you usually do not need to set this environment variable, unless you want to store management profiles in a totally different location.</td>
</tr>
</tbody>
</table>

You set these environment variables by using the operating system of each computer, or by using the patrol.conf file for each installation of the PATROL Console Server.

**NOTE**

If there are other applications on the same computer that need different values for these environment variables, set them for the PATROL Console Server by using the patrol.conf file, instead of the operating system.
If the target directories do not exist, the PATROL Console Server creates them when it starts. However, it is good practice to create the target directories first to ensure that the PATROL Console Server account has full access to the directories.

For more information about the files and directories used by the PATROL Console Server, see PATROL Console Server and RTserver Getting Started.

**Setting Environment Variables for the PATROL Console Server By Using the patrol.conf File**

1. Open the `patrol.conf` file, which is located in the `%BMC_ROOT%\common\patrol.d` directory.

2. Locate the `[CSERVER]` section in the file.

3. Locate or create the lines for setting environment variables.

```plaintext
[CSERVER]
set environment
  PATROL_ROOT_CONFIG=SharedDrive:\Patrol7\config
  PATROL_ROOT_ETC=SharedDrive:\Patrol7\etc
  PATROL_ROOT_TMP=SharedDrive:\Patrol7\tmp
  PATROL_ROOT_LOG=SharedDrive:\Patrol7\log
end
defaultAccount=patrol:82EA557C2461C11C517AA3A
```

4. Save and close the `patrol.conf` file.

**Verifying that Each PATROL Console Server Uses the Shared Directories**

1. If the PATROL Console Server is running, stop it.

2. Backup the `config`, `etc`, `log` and `tmp` subdirectories under of the `PATROL_ROOT` directory on the local file system.

3. Remove the `config`, `etc`, `log` and `tmp` subdirectories from the local file system.

4. Start the PATROL Console Server.

5. Verify that the `config`, `etc`, `log` and `tmp` subdirectories are not recreated on the local file system.

6. Verify that the directories identified by the environment variables exist on the shared drive.
7 Stop the PATROL Console Server.

8 If the files in the `config`, `etc`, `log` and `tmp` subdirectories must be on the local file system for another application, restore them.

**Moving RTserver Configuration Directory to the Shared Drive**

If you want cluster failover for the RTserver, you move the RTserver configuration directory for each RTserver installation to the same shared drive on the cluster.

You move this directory by using the RTHOME environment variable in the operating system of each computer.

1 Create a directory on the same shared drive that the PATROL Console Server uses.

2 Copy the `standard` directory from the `%BMC_ROOT%\common\smartsockets` directory on one of the computers to the newly created directory on the shared drive.

3 In the operating system of each computer, set the RTHOME system environment variable to the newly created directory on the shared drive (the parent directory of the `standard` directory).

   For example, if the shared drive contains the directory `j:\bmc\common\smartsockets\standard`, set the RTHOME environment variable to `j:\bmc\common\smartsockets`.  

   **NOTE**

   If you want the RTserver to be part of an RTserver cloud, configure the RTserver by using the `rtserver.cm` file in the `standard` directory on the shared drive. For more information about RTserver clouds, see *PATROL Console Server and RTserver Getting Started*.  

Setting Up the PATROL Console Server and RTserver in the Cluster

You use the cluster management software to configure the group and resources for the PATROL Console Server.

This section provides information specific to configuring the PATROL Console Server and RTServer for a Windows cluster. For general instructions on how to set up a group and resources, see your cluster documentation.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Choosing a Network Name and IP Address</td>
<td>page 156</td>
</tr>
<tr>
<td>Choosing a Service ID for the PATROL Console Server</td>
<td>page 157</td>
</tr>
<tr>
<td>Configuring the Resource Group</td>
<td>page 157</td>
</tr>
<tr>
<td>Configuring the Resource for the PATROL Console Server</td>
<td>page 158</td>
</tr>
<tr>
<td>Configuring the Resource for the RTserver</td>
<td>page 159</td>
</tr>
<tr>
<td>Verifying Cluster Failover</td>
<td>page 159</td>
</tr>
</tbody>
</table>

**NOTE**

Both the PATROL Console Server and the RTserver are *cluster unaware*. They run in a cluster environment as generic resources and do not communicate back to the cluster.

Choosing a Network Name and IP Address

If you include the RTserver in the cluster failover with the PATROL Console Server, you *must* specify a network name and IP address (also called a virtual host name and IP address) for the group. The network name for an RTserver is part of its logical connection name (such as tcp:myhost:2059) that the PATROL Console Server, PATROL Agent, PATROL Central Operator, and other RTclients use to locate the RTserver. Specifying a network name and IP address ensures that the logical connection name for the RTserver remains the same when failover occurs.

Specifying a network name and IP address for the resource group is optional if the PATROL Console Server is the only application in the group.
Choosing a Service ID for the PATROL Console Server

The service ID of the PATROL Console Server is how the PATROL Console Server is identified to the RTserver and other PATROL components, such as PATROL Central Operator. The service ID is also incorporated in several of the files and directories used by the PATROL Console Server.

You specify a service ID by using the command line option or startup parameter when configuring the PATROL Console Server resource in the cluster.

NOTE

Unlike when running the PATROL Console Server outside of a cluster, you do not have the option of using a default service ID when running in the cluster.

For more information about the service ID, see “Choosing a Service ID for the PATROL Console Server” on page 64.

Configuring the Resource Group

Use the following guidelines to set up the resource group in the cluster management software. You can create a new group or use an existing one.

- The group must include the shared drive to which you moved the PATROL Console Server directories.

  For more information, see “Pointing the PATROL Console Server on Each Computer to Directories on the Shared Drive” on page 152.

- The group can have a network name and IP address, but might not require one.

  For more information, see “Choosing a Network Name and IP Address” on page 156.

- The group must include the resource for the PATROL Console Server.

  For more information, see “Configuring the Resource for the PATROL Console Server” on page 158.

- The group can include the resource for the RTserver.

  For more information, see “Configuring the Resource for the RTserver” on page 159.
Configuring the Resource for the PATROL Console Server

Use the following guidelines to set up the PATROL Console Server as a generic resource in the cluster management software.

- The resource must belong to the group for the PATROL Console Server.
  
  For more information, see “Configuring the Resource Group” on page 157.

- Create a resource of type `generic service`, bind the resource to the `cserver` service, and set the service ID by using the `-id serviceID` command line option in the startup parameters.
  
  For more information, see “Choosing a Service ID for the PATROL Console Server” on page 157.

- The resource must have a dependency on the shared drive to which you moved the PATROL Console Server directories.
  
  This dependency ensures that the shared drives are available before the PATROL Console Server is started. For more information, see “Pointing the PATROL Console Server on Each Computer to Directories on the Shared Drive” on page 152.

- If the group has a network name or IP address, the resource for the PATROL Console Server must *not* use it.

- If you want cluster failover for the RTserver, the PATROL Console Server resource must also have a dependency on the RTserver resource.
  
  This dependency ensures that the RTserver is started before the PATROL Console Server started.
Configuring the Resource for the RTserver

Use the following guidelines to set up the RTserver as a generic resource in the cluster management software.

- Put the resource in the same the group as the PATROL Console Server.
  
  For more information, see “Configuring the Resource Group” on page 157.

- Create a resource of type generic service and bind the resource to the rtserver service.

- The PATROL Console Server resource must have a dependency on the RTserver resource, so that the RTserver is brought online before the PATROL Console Server.

- The resource must have a dependency on the shared drive to which you moved the RTserver directory.

Verifying Cluster Failover

1 Use the cluster management software to bring the group containing the PATROL Console Server (and optionally the RTserver) online.

2 Use PATROL Central Operator and PATROL Central Administration to connect to the PATROL Console Server and create or modify each of the following:

- management profiles, including adding PATROL Agents and loading KMs
- impersonation entries and aliases
- permissions for objects (ACL entries)

3 Use the cluster management software to move the group to a different computer.

4 Use PATROL Central Operator and PATROL Central Administration to verify that you can connect to the PATROL Console Server and that all of the modifications persist.
Example Scenarios for Cluster Failover

This section contains example scenarios for using the PATROL Console Server in a cluster environment.

<table>
<thead>
<tr>
<th>Example Scenario</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>New PATROL Console Server in a Cluster</td>
<td>page 160</td>
</tr>
<tr>
<td>New PATROL Console Server and RTserver in a Cluster</td>
<td>page 162</td>
</tr>
<tr>
<td>New PATROL Console Server in a Cluster (Command Line Example)</td>
<td>page 164</td>
</tr>
<tr>
<td>New PATROL Console Server and RTserver in a Cluster (Command Line Example)</td>
<td>page 168</td>
</tr>
</tbody>
</table>

New PATROL Console Server in a Cluster

This section contains an example scenario and solution for using the PATROL Console Server in a cluster environment.

**Scenario**

You want to run the PATROL Console Server in a Windows cluster environment. You do not have an existing PATROL Console Server or you do not need to preserve any information from your original PATROL Console Server.

You have two Windows computers in the cluster. Both computers use the same shared drive (J:).

You want to create a resource group for just the PATROL Console Server. You are not using cluster failover for the RTserver. You want to use an RTserver outside of the cluster.

You want to use a domain account for the PATROL Console Server account. This domain account is valid for both computers and has access to the shared J: drive.

**Solution**

1. On the first computer in the cluster, install the PATROL Console Server.
   - Specify the domain account as the PATROL Console Server account, and choose to *not* start the PATROL Console Server automatically.
   - Set the RTSERVERS environment variable to use the RTserver outside of the cluster.
2 Change startup properties of the newly created PATROL Console Server service to manual startup, instead of automatic.

3 At the root of the shared J: drive, create a new directory named Patrol7.

4 Under the new Patrol7 directory, create the subdirectories config, etc, log, and tmp.

The shared J: drive now contains the directories in the image below:

```
$Patrol7$
  $config$
  $etc$
  $log$
  $tmp$
```

5 Set up the environment variables for the PATROL Console Server.

   A Open the patrol.conf file in the %BMC_ROOT%\common\patrol.d directory in a text editor.

   B Locate the [CSERVER] section of the patrol.conf file and add the lines in bold below:

   ```
   [CSERVER]
   set environment
   PATROL_ROOT_CONFIG=J:\Patrol7\config
   PATROL_ROOT_ETC=J:\Patrol7\etc
   PATROL_ROOT_LOG=J:\Patrol7\log
   PATROL_ROOT_TMP=J:\Patrol7\tmp
   end
   defaultAccount=patrol:82EA557C2461C11C517AA3A
   ```

   C Save and close the patrol.conf file.

6 Repeat steps 1 through 5 on the second computer in the cluster.

   For steps 3 and 4, simply verify that the directories exist, instead of creating them.

7 Start the Cluster Administration utility.

8 In the Cluster Administration utility, create a new group named PATROL Console Server.

9 Add the shared drive J: to the new group.
10 Create a new resource of type *generic service* to the group, and bind this resource to the *cserver* service. Specify the *-ID serviceID* option as the startup parameter for the resource.

11 Add the shared drive as a dependency of the PATROL Console Server resource.

12 Bring the group online and verify that you can move the group from one computer to another.

**New PATROL Console Server and RTserver in a Cluster**

This section contains an example scenario and solution for using the PATROL Console Server in a cluster environment.

**Scenario**

You want to run the PATROL Console Server and RTserver in a Windows cluster environment. You do not have an existing PATROL Console Server or you do not need to preserve any information from your original PATROL Console Server.

You have two Windows computers in the cluster. Both computers use the same shared drive (J:).

You want to create a cluster group for both the PATROL Console Server and the RTserver.

You want to use a domain account for the PATROL Console Server account. This domain account is valid for both computers and has access to the shared J: drive.

**Solution**

1 On the first computer in the cluster, install the PATROL Console Server and RTserver.

Specify the domain account as the PATROL Console Server account, and choose to *not* start the PATROL Console Server or RTserver automatically.

2 Change the startup properties of the following newly created services to manual startup, instead of automatic.

- PATROL Console Server service
- SmartSockets RTserver service
3 Create the following directories on the shared J: drive:

- BMC\Patrol7\config
- BMC\Patrol7\etc
- BMC\Patrol7\log
- BMC\Patrol7\tmp
- BMC\rtserver_smartsockets\

4 Copy the standard subdirectory directory in the %BMC_ROOT%\common\smartsockets\ directory on the local computer to the new rtserver_smartsockets\ directory on the shared drive.

The shared J: drive now contains the directories in the image below:

5 Create the RTHOME environment variable and set it to J:\rtserver_smartsockets\

6 Set up the environment variables for the PATROL Console Server.

A Open the patrol.conf file in the %BMC_ROOT%\common\patrol.d directory in a text editor.

B Locate the [CSERVER] section of the patrol.conf file and add the lines in bold below:

```
[CSERVER]
set environment
PATROL_ROOT_CONFIG=J:\BMC\Patrol7\config
PATROL_ROOT_ETC=J:\BMC\Patrol7\etc
PATROL_ROOT_LOG=J:\BMC\Patrol7\log
PATROL_ROOT_TMP=J:\BMC\Patrol7\tmp
end
defaultAccount=patrol:82EA557C2461C11C517AA3A
```

C Save and close the patrol.conf file.

7 Repeat steps 1 through 6 on the second computer.

For steps 3 and 4, simply verify that the directories exist, instead of creating them.
8 Start the Cluster Administration utility.

9 In the Cluster Administration utility, create a new group named *PATROL Console Server*.

10 Ensure that the new group has a network name and IP address.

11 Add the shared drive J: to the new group.

12 Create a new resource of type *generic service* to the group, and bind this resource to the *cserver* service. Specify the `-ID serviceID` option as the startup parameter for the resource.

13 Create a new resource of type *generic service* to the group, and bind this resource to the *rtserver* service.

14 Set up the dependencies between the resources:

   - Add the shared drive J: as a dependency of both the PATROL Console Server resource and the RTserver resource.
   - Add the RTserver resource as a dependency of the PATROL Console Server resource.

15 Bring the group online and verify that you can move the group from one computer to another.

**New PATROL Console Server in a Cluster (Command Line Example)**

This section contains an example scenario and solution for using the PATROL Console Server in a cluster environment.

**Scenario**

You want to run the PATROL Console Server in a Windows cluster environment. You do not have an existing PATROL Console Server or you do not need to preserve any information from your original PATROL Console Server.

You have two Windows computers in the cluster *CS_Test*. Both computers use the same shared drive (J:), which is defined as the resource *CS Disk* in the cluster.
You want to use the resource group `CS_Group` for the PATROL Console Server. You are not using cluster failover for the RTserver. You want to use an RTserver outside of the cluster.

You want to use a domain account for the PATROL Console Server account. This domain account is valid for both computers and has access to the shared drive.

**NOTE**

This scenario is the same as “New PATROL Console Server in a Cluster” on page 160 except that you use the cluster administration command line interface and the resource group is already defined.

**Solution**

1. Complete step 1 on page 160 through step 6 on page 161 of the “New PATROL Console Server in a Cluster” example scenario to install the PATROL Console Server on the computers and point the installations to directories on the shared drive.

2. Use the cluster administration command line interface to check the current configuration for the resource group and shared disk resource for the PATROL Console Server service.

   Lines in bold below indicate the commands to enter.

   ```
   C:\>cluster.exe /CLUSTER:CS_Test RES "CS Disk"
   Listing status for resource 'CS Disk':
   Resource    Group    Node            Status
   -------------------- ---- --------------- ----
   CS Disk      CS_Group KMQADATACL1C    Offline
   ```

   ```
   C:\>cluster.exe /CLUSTER:CS_Test GROUP CS_Group
   Listing status for resource group 'CS_Group':
   Group    Node            Status
   -------------------- ---- --------------- ----
   CS_Group KMQADATACL1C    Offline
   ```
Example Scenarios for Cluster Failover

3 Use the cluster administration command line interface to create a resource for the PATROL Console Server service. All of the following can be accomplished in one command:

- Create a new **Generic Service** resource for cluster which belongs to the Resource Group, give a name and description which is indicative its use in the cluster. (/CREATE, /GROUP, /TYPE, and /Prop Description options).

- Add the shared drive that has the PATROL Console Server directories as a dependency (/ADDDEP option).

- Bind the resource to the default installed service name for PATROL Console Server (/PRIV ServiceName option).

- Specify the startup parameters for the service so that when it starts it uses the specified service ID (/PRIV StartupParameters option).

- Set the resource so that the cluster will continue to run if the PATROL Console Server fails and cannot be restarted (/Prop RestartAction option).

```
C:\>cluster.exe /CLUSTER:CS_Test RES "PATROL Console Server" /CREATE
/ GROUP:CS_Group / TYPE:"Generic Service" / ADDDEP:"CS Disk" / PROP
Description="PATROL Console Server Resource" / PRIV ServiceName=cserver / PRIV
StartupParameters="-id FAILOVER_TEST" / Prop RestartAction=1

Creating resource 'PATROL Console Server'...

<table>
<thead>
<tr>
<th>Resource</th>
<th>Group</th>
<th>Node</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATROL Console Server</td>
<td>CS_Group</td>
<td>KMQADATACL1C</td>
<td>Offline</td>
</tr>
</tbody>
</table>

Making resource 'PATROL Console Server' depend on resource 'CS Disk'...
```
4 Use the cluster administration command line interface to verify the options.

\texttt{C:\> cluster.exe /CLUSTER:CS_Test RES "PATROL Console Server" /prop}

Listing properties for 'PATROL Console Server':

\begin{tabular}{l l l}
\hline
T Resource & Name & Value \\
\hline
PATROL Console Server Name & & PATROL Console Server \\
S PATROL Console Server Type & & Generic Service \\
S PATROL Console Server Description & & PATROL Console Server \\
Failover Setup & & \\
D PATROL Console Server RestartAction & & 1 (0x1) \\
D PATROL Console Server RestartThreshold & & 3 (0x3) \\
D PATROL Console Server RestartPeriod & & 900000 (0xdbba0) \\
D PATROL Console Server RetryPeriodOnFailure & & 4294967295 (0xffffffff) \\
D PATROL Console Server PendingTimeout & & 180000 (0x2bf20) \\
\hline
\end{tabular}

\texttt{C:\> cluster.exe /CLUSTER:CS_Test RESOURCE "PATROL Console Server" /priv}

Listing private properties for 'PATROL Console Server':

\begin{tabular}{l l l}
\hline
T Resource & Name & Value \\
\hline
S PATROL Console Server ServiceName & & cserver \\
S PATROL Console Server StartupParameters & & -id FAILOVER_TEST \\
D PATROL Console Server UseNetworkName & & 0 (0x0) \\
\hline
\end{tabular}

5 Bring the PATROL Console Server resource online.

\texttt{C:\> cluster.exe /CLUSTER:CS_Test RES "PATROL Console Server" /ON}

Bringing resource 'PATROL Console Server' online...
New PATROL Console Server and RTserver in a Cluster (Command Line Example)

This section contains an example scenario and solution for using the PATROL Console Server in a cluster environment.

Scenario

You want to run the PATROL Console Server and RTserver in a Windows cluster environment. You do not have an existing PATROL Console Server or you do not need to preserve any information from your original PATROL Console Server.

You have two Windows computers in the cluster CS_Test. Both computers use the same shared drive (J:), which is defined as the resource CS Disk in the cluster.

You want to use the resource group CS_Group for both the PATROL Console Server and the RTserver.

You want to use a domain account for the PATROL Console Server account. This domain account is valid for both computers and has access to the shared J: drive.

NOTE

This scenario is the same as “New PATROL Console Server and RTserver in a Cluster” on page 162 except that you use the cluster administration command line interface and the resource group is already defined.

Solution

1 Complete step 1 on page 162 through step 7 on page 163 of the “New PATROL Console Server and RTserver in a Cluster” example scenario to install the PATROL Console Server and RTserver on the computers and point the installations to directories on the shared drive.

2 Complete step 2 on page 165 through step 4 on page 167 of the “New PATROL Console Server in a Cluster (Command Line Example)” example scenario to set up the PATROL Console Server in the cluster, but not bring it online.
3 Use the cluster administration command line interface to verify the current configuration for the virtual IP address resource.

Lines in bold below indicate the commands to enter.

```
C:\>cluster.exe /CLUSTER:CS_TEST RESOURCE "CS IP"
Listing status for resource 'CS IP':

<table>
<thead>
<tr>
<th>Resource</th>
<th>Group</th>
<th>Node</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS IP</td>
<td>CS_Group</td>
<td>KMQADATACL1C</td>
<td>Online</td>
</tr>
</tbody>
</table>
```

```
C:\>cluster.exe /CLUSTER:CS_TEST RES "CS IP" /PRIV
Listing private properties for 'CS IP':

<table>
<thead>
<tr>
<th>T</th>
<th>Resource</th>
<th>Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>S</td>
<td>CS IP</td>
<td>Network</td>
<td>Public Network</td>
</tr>
<tr>
<td>S</td>
<td>CS IP</td>
<td>Address</td>
<td>192.168.20.195</td>
</tr>
<tr>
<td>S</td>
<td>CS IP</td>
<td>Subnet Mask</td>
<td>255.255.255.0</td>
</tr>
<tr>
<td>D</td>
<td>CS IP</td>
<td>Enable NetBIOS</td>
<td>1 (0x1)</td>
</tr>
</tbody>
</table>
```

**NOTE**

Use the virtual IP address in the RTSERVERS environment variable used by RTclients such as PATROL Central Operator, PATROL Console Server, and PATROL Agent.

4 Create a resource for the RTserver service. All of the following can be accomplished in one command:

- Create a new *Generic Service* resource for cluster which belongs to the resource group, giving it a name and description which is indicative its use in the cluster. (/CREATE, /GROUP, /TYPE, and /Prop Description options).

- Add the shared drive that has the shared directories as a dependency (/ADDDEP option).

- Add the virtual IP address as a dependency (/ADDDEP option).

- Bind the resource to the default installed service name for RTserver (/PRIV ServiceName option).

- Set the resource so that the cluster will continue to run if the PATROL Console Server fails and cannot be restarted (/Prop RestartAction option).
5 Verify the options for the RTserver resource.

```
C:\> cluster.exe /CLUSTER:CS_Test RESOURCE "Smartsockets RTServer" /CREATE
/GROUP:CS_Group /TYPE:"Generic Service" /ADDDEP:"CS Disk" /ADDDEP:"CS IP" /PROP
RestartAction=1 /PRIV ServiceName=rtserver

Creating resource 'Smartsockets RTServer'...

Resource Group Node Status
-------------------- -------------------- --------------- ------
Smartsockets RTServer CS_Group KMQADATACL1C Offline

Making resource 'Smartsockets RTServer' depend on resource 'CS Disk'...
Making resource 'Smartsockets RTServer' depend on resource 'CS IP'...
```

6 Setup the dependency between the PATROL Console Server and RTserver resources.

```
C:\> cluster.exe /CLUSTER:CS_Test RES "Smartsockets RTServer" /prop

Listing properties for 'Smartsockets RTServer':

T Resource Name Value
---- -------------------- ------------------------- ------------------------------
SR Smartsockets RTServer Name Smartsockets RTServer
S Smartsockets RTServer Type Generic Service
S Smartsockets RTServer Description Smartsockets RTServer Resource
D Smartsockets RTServer RestartAction 1 (0x1)
D Smartsockets RTServer RestartThreshold 3 (0x3)

C:\> cluster.exe /CLUSTER:CS_Test RESOURCE "Smartsockets RTServer" /priv

Listing private properties for 'Smartsockets RTServer':

T Resource Name Value
---- -------------------- -----------------------
S Smartsockets RTServer ServiceName rtserver
D Smartsockets RTServer UseNetworkName 1 (0x1)
```
7 Bring the RTserver resource online.

```
C:\> cluster.exe /CLUSTER:CS_Test RES "Smartsockets RTServer" /ON
Bringing resource 'Smartsockets RTServer' online...
```

8 Bring the PATROL Console Server resource online.

```
C:\> cluster.exe /CLUSTER:CS_Test RES "PATROL Console Server" /ON
Bringing resource 'PATROL Console Server' online...
```

## Replicating the PATROL Console Server

You can copy configuration data from one PATROL Console Server to another PATROL Console Server using the admin_copy utility. For more information about the admin_copy utility, see “admin_copy” on page 177.

This section contains the following tasks:

<table>
<thead>
<tr>
<th>Task</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refreshing a Backup Copy of the PATROL Console Server</td>
<td>page 171</td>
</tr>
<tr>
<td>Creating a Copy For Other Instances of the PATROL Console Server</td>
<td>page 173</td>
</tr>
</tbody>
</table>

### Refreshing a Backup Copy of the PATROL Console Server

You can refresh the backup of a PATROL Console Server with a subset of the files that are changed through PATROL Central, such as ACLs, rights, aliases, impersonations, and management profiles.

**Before You Begin**

Create a full backup of the source PATROL Console Server. For more information about creating a full backup, see “Backing Up and Restoring the PATROL Console Server” on page 120.
Replicating the PATROL Console Server

You must know the service ID of the PATROL Console Server you will use as both the source and destination for the copy operation.

You must know the name of the output directory.

To Refresh a Backup Copy of the PATROL Console Server

1. Stop the source PATROL Console Server.

2. Run the following command:

   ```
   C> admin_copy.cmd -src_c sourceCserverId -dst_c sourceCserverId -profiles -out outputDirectory (Windows)
   ```

   ```
   % admin_copy.sh -src_c sourceCserverId -dst_c sourceCserverId -profiles -out outputDirectory (Unix)
   ```

   **NOTE**
   In this task, the service ID is the same for both the source and destination PATROL Console Server.

3. Start the source PATROL Console Server.

Example of Creating a Backup Copy of the PATROL Console Server

For Windows, the drive that contains the PATROL_ROOT directory on the csbackup computer can be mapped to the csprod computer. For example, F:\Program Files\BMC Software\Patrol7.

For Unix, the /opt/bmc/Patrol7 directory on the csbackup computer can be mounted on the csprod computer as /data/csbackup/opt/bmc/Patrol7.

For Windows:

```
C> admin_copy.cmd -src_c csprod -dst_c csprod -profiles -out "F:\Program Files\BMC Software\Patrol7"
```

For Unix:

```
% admin_copy.sh -src_c csprod -dst_c csprod -profiles -out /data/csbackup/opt/bmc/Patrol7
```
Creating a Copy For Other Instances of the PATROL Console Server

If you have several concurrent instances of PATROL Console Server in your environment, you can use the admin_copy utility to use the same configuration data and management profiles for each instance.

**Before You Begin**

You must have an instance of the PATROL Console Server on the destination computer for the copy operation.

You must know the service ID of the PATROL Console Server you will use as the source for the copy operation.

You must know the service ID of the PATROL Console Server you will use as the destination for the copy operation.

You must know the name of the output directory.

**To Create a Copy For Other Instances of the PATROL Console Server**

1. Stop the source and the destination PATROL Console Servers.

2. Run the following command:

   - C> admin_copy.cmd -src_c sourceCserverId -dst_c destinationCserverId -profiles -out outputDirectory (Windows)
   - % admin_copy.sh -src_c sourceCserverId -dst_c destinationCserverId -profiles -out outputDirectory (Unix)

   **NOTE**
   
   In this task, the service ID is different for the source and destination PATROL Console Servers.

3. Start the source and destination PATROL Console Servers.
Example of Creating a Copy For Other Instances of the PATROL Console Server

For Windows, the drive that contains the PATROL_ROOT directory on the cstest computer can be mapped to the csprod computer. For example, F:\Program Files\BMC Software\Patrol7.

For Unix, the /opt/bmc/Patrol7 directory on the cstest computer can be mounted on the csprod computer as /data/cstest/op/bmc/Patrol7.

| For Windows:                                      |
| C> admin_copy.cmd -src_c csprod -dst_c cstest -profiles -out "F:\Program Files\BMC Software\Patrol7" |

| For Unix:                                         |
| % admin_copy.sh -src_c csprod -dst_c cstest -profiles -out /data/cstest/op/bmc/Patrol7 |
Chapter 5 Command Line Utilities With PATROL Console Server

This chapter provides information about several command line utilities that are available with the PATROL Console Server.

This chapter contains the following topics:

About the Command Line Utilities ................................................................. 176
General Command Line Utilities ................................................................. 177
  admin_copy ................................................................. 177
  admincli ................................................................. 180
  confManager .............................................................. 194
  sec_encrypt_p3x .......................................................... 198
  stopPatrolService .......................................................... 199
  hg_import ................................................................. 202
  hg_export ................................................................. 204
Command Line Utilities for Management Profiles ................................. 206
  lt_counters ............................................................... 206
  lt_db_util ............................................................... 216
  lt_import ................................................................. 218
  lt_export ................................................................. 222
About the Command Line Utilities

This section contains reference information about the following command line utilities:

<table>
<thead>
<tr>
<th>Utility</th>
<th>Brief Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>General Command Line Utilities</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>admin_copy</td>
<td>copies configuration data from one console server to another console server</td>
<td>page 177</td>
</tr>
<tr>
<td>admincli</td>
<td>configures administrative data such as impersonations and rights</td>
<td>page 180</td>
</tr>
<tr>
<td>confManager</td>
<td>edits patrol.conf files</td>
<td>page 194</td>
</tr>
<tr>
<td>sec_encrypt_p3x</td>
<td>encrypts passwords</td>
<td>page 198</td>
</tr>
<tr>
<td>stopPatrolService</td>
<td>stops PATROL servers</td>
<td>page 199</td>
</tr>
<tr>
<td>hg_import</td>
<td>imports managed system groups</td>
<td>page 202</td>
</tr>
<tr>
<td>hg_export</td>
<td>exports managed system groups</td>
<td>page 204</td>
</tr>
<tr>
<td><strong>Command Line Utilities for Management Profiles</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>lt_counters</td>
<td>lists management profile performance counters</td>
<td>page 206</td>
</tr>
<tr>
<td>lt_import</td>
<td>imports .mof files into management profiles</td>
<td>page 218</td>
</tr>
<tr>
<td>lt_export</td>
<td>exports management profiles to .mof files</td>
<td>page 222</td>
</tr>
</tbody>
</table>
General Command Line Utilities

The PATROL Console Server includes the following general command line utilities:

<table>
<thead>
<tr>
<th>Utility</th>
<th>Brief Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>admin_copy</td>
<td>copies configuration data from one console server to another console server</td>
<td>page 177</td>
</tr>
<tr>
<td>admincli</td>
<td>configures administrative data such as impersonations and rights</td>
<td>page 180</td>
</tr>
<tr>
<td>confManager</td>
<td>edits <code>patrol.conf</code> files</td>
<td>page 194</td>
</tr>
<tr>
<td>sec_encrypt_p3x</td>
<td>encrypts passwords</td>
<td>page 198</td>
</tr>
<tr>
<td>stopPatrolService</td>
<td>stops PATROL servers</td>
<td>page 199</td>
</tr>
<tr>
<td>hg_import</td>
<td>imports managed system group definitions</td>
<td>page 202</td>
</tr>
<tr>
<td>hg_export</td>
<td>exports managed system group definitions</td>
<td>page 204</td>
</tr>
</tbody>
</table>

**admin_copy**

Use the `admin_copy` utility to make a copy of the administrative configuration files for the PATROL Console Server. The utility copies the files from the installation directory identified by the PATROL_ROOT environment variable to a user-specified output directory.

The files copied by `admin_copy` are:

- impersonation table
- alias table
- managed system group definitions
- access control database containing object rights and permissions
- `{optional}` management profiles

Since the configuration files are identified by the server’s service ID, the utility requires an input `source` service ID and a `destination` service ID. The source service ID is used to locate the files to be copied with the current PATROL_ROOT directory. The destination service ID is used to name the results created in the user-specified output directory.

Once the utility has run, you can copy the contents of the output directory to the destination server installation. As an alterative, you can specify the PATROL_ROOT directory of the destination server as the output directory so that the files are copied directly from one server installation to another.
**NOTE**

Do not copy or replace the administrative configuration files while the server is running.

Before you run the admin_copy utility, you must stop the source server.

Before you can copy the results of the admin_copy operation into the destination PATROL_ROOT directory, you must stop the destination server. If the output directory argument specifies the destination server’s PATROL_ROOT directory, you must stop the destination server before running admin_copy.

**Location of admin_copy**

The admin_copy utility is located in the following directories:

- `%BMC_ROOT%\common\bmc` (Windows)
- `$BMC_ROOT/common/bmc` (Unix)

On Windows, the file name is `admin_copy.cmd`.

On Unix, the file name is `admin_copy.sh`.

**Syntax for admin_copy**

```
admin_copy -src_c sourceCserverId -dst_c destinationCserverId -out outputDirectory [-d] [-profiles]
admin_copy -h
admin_copy -v
```
Command Line Arguments for admin_copy

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>-src_c sourceCserverId</td>
<td>specifies the service ID of the PATROL Console Server used as the source for the copy operation. Configuration files for this PATROL Console Server are copied from the PATROL_ROOT directory to the directory identified by the -out argument.</td>
</tr>
<tr>
<td>-dst_c destinationCserverId</td>
<td>specifies the service ID of the PATROL Console Server used as the destination for the copy operation. The configuration files copied to the directory identified by the -out argument are named based on this destination service ID.</td>
</tr>
<tr>
<td>-out outputDirectory</td>
<td>specifies the name of the output directory. If this directory does not exist, it will be created. This directory may be the name of a temporary directory or the name of the PATROL_ROOT directory from which the destination PATROL Console Server runs. Paths containing spaces should be enclosed in double quotes.</td>
</tr>
<tr>
<td>-profiles</td>
<td>&lt;optional&gt; enables copying of management profiles from one PATROL Console Server installation to another</td>
</tr>
<tr>
<td>-d</td>
<td>&lt;optional&gt; enables the output of debug messages</td>
</tr>
<tr>
<td>-h</td>
<td>displays the list of arguments</td>
</tr>
<tr>
<td>-v</td>
<td>displays the version of the tool</td>
</tr>
</tbody>
</table>

Notes for admin_copy

You must stop the source server before running this utility to ensure that the files copied are complete and consistent. The source server is the server identified by the -src_c argument.

You must stop the destination server before the results in the output directory can be copied into the PATROL_ROOT directory of the destination server. The destination server is the server identified by the -dst_c argument.

For more information on the directory structure and files used by PATROL Console Server, see Appendix A, “Environment Variables, Directories, and Files.”

For more information on the service IDs used by PATROL Console Server, see “Choosing a Service ID for the PATROL Console Server” on page 64.
Examples of Running admin_copy

The following example assumes that the admin_copy utility is executed on a machine named server1 and drive E: is mapped to the machine named server2 where the destination Console Server is installed. You should stop the PATROL Console Server on both server1 and server2 before performing this operation.

```
C:\> admin_copy -src_c server1 -dst_c server2 -out "E:\Program Files\BMC Software\Patrol7"
done
```

In addition to this example, the admin_copy utility can be used to perform a variety of administrative tasks:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Changing the Service ID of the PATROL Console Server</td>
<td>page 136</td>
</tr>
<tr>
<td>Moving the PATROL Console Server</td>
<td>page 138</td>
</tr>
<tr>
<td>Replicating the PATROL Console Server</td>
<td>page 171</td>
</tr>
</tbody>
</table>

admincli

**NOTE**

In previous releases of PATROL Central, the term “privileges” was used to refer to controlling access to particular functions or operations, and the term “rights” was used to refer to controlling access to particular objects. In the 7.5.00 release of PATROL Central, this terminology was updated to be more in line with other BMC products. Now, the term “rights” is used to refer to controlling access to functions or operations, and the term “permissions” is used to refer to controlling access to particular objects. However, to maintain backwards compatibility with existing releases, the command line arguments for admincli were not modified.

Use the admincli command line utility to administer tables in the PATROL Console Server.

The tables supported by the current release are:
The basic command line syntax consists of identifying a target PATROL service, optionally supplying user and password information for authentication with the PATROL service, identifying the type of table, and specifying the operations to be performed on that table.

The valid operations for admincli options include the following:

<table>
<thead>
<tr>
<th>admincli options</th>
<th>Valid operations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Control List (ACL) table</td>
<td>list, add, modify, remove</td>
</tr>
<tr>
<td>Alias table</td>
<td>list, add, modify, remove</td>
</tr>
<tr>
<td>Configuration</td>
<td>modify default account</td>
</tr>
<tr>
<td>Impersonation table</td>
<td>list, add, modify, remove, move</td>
</tr>
<tr>
<td>User Sessions</td>
<td>list</td>
</tr>
</tbody>
</table>
### Location of admincli

The admincli utility is located in the following directories:

- `%BMC_ROOT%\common\bmc\bin\target` (Windows)
- `$BMC_ROOT/common/bmc/bin/target` (Unix)

### Syntax for admincli

<table>
<thead>
<tr>
<th>command</th>
<th>full Service Name</th>
<th>console Server ID</th>
<th>distribution Server ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>admincli -s fullServiceName [-user username -pwd encryptedPassword] -table tableType tableOptions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>admincli -c consoleServerID [-user username -pwd encryptedPassword] -table tableType tableOptions</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>admincli -d distributionServerID [-user username -pwd encryptedPassword] -table tableType tableOptions</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

`admincli -h`

`admincli -v`
General Command Line Arguments for `admincli`

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>-c consoleServerID</td>
<td>specifies the service ID of a PATROL Console Server</td>
</tr>
<tr>
<td>-s fullServiceName</td>
<td>specifies the full service name of the PATROL Console Server</td>
</tr>
<tr>
<td>-user username</td>
<td>&lt;optional&gt; specifies the user name to use when authenticating with the server&lt;br&gt;&lt;br&gt;If this option is omitted, and the server requires authentication, you are prompted to enter the information from the command line. Both -user and -pwd can be omitted, but if one is specified the other must be specified.</td>
</tr>
<tr>
<td>-pwd encryptedPassword</td>
<td>&lt;optional&gt; specifies the encrypted password to use when authenticating with the server&lt;br&gt;&lt;br&gt;You can use the sec_encrypt_p3x utility to encrypt the password (see “sec_encrypt_p3x” on page 198). If this option is omitted, and the server requires authentication, you are prompted to enter the clear text password from the command line. Both -user and -pwd can be omitted, but if one is specified the other must be specified.</td>
</tr>
<tr>
<td>-table tableType tableOptions</td>
<td>specifies the type of table for the current operation&lt;br&gt;&lt;br&gt;Values for tableType include:&lt;br&gt;&lt;ul&gt;&lt;li&gt;acl&lt;/li&gt;&lt;li&gt;alias&lt;/li&gt;&lt;li&gt;config&lt;/li&gt;&lt;li&gt;hg&lt;/li&gt;&lt;li&gt;imp&lt;/li&gt;&lt;li&gt;privilege&lt;/li&gt;&lt;li&gt;session&lt;/li&gt;&lt;/ul&gt;&lt;br&gt;For values for tableOptions, see “Options for Alias Table Administration” on page 186 and “Options for Impersonation Table Administration” on page 187.</td>
</tr>
<tr>
<td>-h</td>
<td>displays the list of arguments</td>
</tr>
<tr>
<td>-v</td>
<td>displays the version of the tool</td>
</tr>
</tbody>
</table>
Options for Administering Access Permissions

The following options apply for administering access permissions, which were previously known as rights.

Access permissions are defined in terms of an access-control-list (ACL) object and one or more access-control-entries (ACE).

An ACL is composed of:

- unique name
- boolean flag for “ACL inheritable by child objects”
- 0 to N access-control-entries

The unique name for an ACL is based on the namespace path for the object to which the ACL applies. To see a list of existing ACL names, type `-table acl -op ls`.

In the following table, the argument `aclName` refers to the namespace path. ACL names (namespace paths) containing spaces must be quoted.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-table acl -op ls</code></td>
<td>prints a list of ACLs</td>
</tr>
<tr>
<td></td>
<td>Each ACL is identified by the namespace path for the object to which the ACL applies.</td>
</tr>
<tr>
<td><code>-table acl -op ls -name aclName</code></td>
<td>prints a list of ACEs for the specified namespace path</td>
</tr>
<tr>
<td><code>-table acl -op add -name aclName [-[no]inherit]</code></td>
<td>adds a new ACL</td>
</tr>
<tr>
<td></td>
<td>If the <code>-[no]inherit</code> option is missing, the default value is <code>-inherit</code>. This option corresponds to the PATROL Central Administrator checkbox “ACL inheritable by child objects.”</td>
</tr>
<tr>
<td><code>-table acl -op remove -name aclName</code></td>
<td>destroys the ACL and all of the ACEs defined for the specified ACL</td>
</tr>
<tr>
<td><code>-table acl -op set -name aclName [-no]inherit</code></td>
<td>sets the inheritable flag for a specified ACL</td>
</tr>
</tbody>
</table>
Options for Changing the Default Account

The following option applies to changing the default account.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
</table>
| -table acl -op set_ace -name aclName 
  ace_type allow | creates a new ACE for the specified ACL for each user or group specified 
  rights mask 
  -u_list user1[,user2[...]] 
  -g_list group1[,group2[...]] | The -ace_type argument defines the type of entry, either allow or deny. The -rights argument defines the object permissions to be assigned to the ACE. The mask value is an integer bit mask defining the required permissions: 1=read, 2=write, 3=read/write, and so on. |
| -table acl -op del_ace -name aclName 
  ace_type allow | removes an ACE entry from the specified ACL for each user or group specified 
  rights mask 
  -u_list user1[,user2[...]] 
  -g_list group1[,group2[...]] | |

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
</table>
| -table config -op set 
  -def_user username 
  -def_pass encryptedPassword | changes the cserver's default account in patrol.conf; on Windows, it also changes the Windows Service account The account specified by -def_user must already exist. The value specified by -def_pass must be encrypted by using the sec_encrypt_p3x utility. See “sec_encrypt_p3x” on page 198. The account used to log on to the console server from admincli requires certain rights. This is the account specified by the -user argument, or the account entered from a prompt if -user was not specified. This account must have the SET DEFAULT ACCOUNT FOR THE SERVICE right. The default, out-of-the-box configuration grants this right to members of the patadm and patscdadm user groups. |
Options for Assigning Rights

The following options apply when assigning rights, which were previously known as privileges.

In the following table, the argument privilege refers to the internal object name for a right (previously known as privilege), which is the value specified in one of the PATROL_ROOT/lib/mof/privs/*.mof files. Typically, this name is different from the right description displayed in the GUI. To get the internal object name for a right, along with its description, use the first form of the command to list all rights by name.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>-table privilege -op ls</td>
<td>prints the list of rights by name</td>
</tr>
<tr>
<td>-table privilege -op ls -name privilege</td>
<td>prints the users and/or groups that have been assigned the specified right</td>
</tr>
<tr>
<td>-table privilege -op add -name privilege</td>
<td>assigns the specified right to the specified users and/or groups</td>
</tr>
<tr>
<td></td>
<td>At least one user or group must be specified.</td>
</tr>
<tr>
<td>-table privilege -op remove -name privilege</td>
<td>removes the right from the specified users and/or groups</td>
</tr>
<tr>
<td></td>
<td>At least one user or group must be specified.</td>
</tr>
<tr>
<td>-table privilege -op set -name privilege</td>
<td>replaces all current assignments for the specified right with the specified list of users/groups</td>
</tr>
<tr>
<td></td>
<td>At least one user or group must be specified.</td>
</tr>
</tbody>
</table>

Options for Alias Table Administration

The following options apply to operations on the alias table only.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>-table alias -op ls</td>
<td>lists all of the entries in the alias table</td>
</tr>
<tr>
<td>-table alias -op add -name alias -login user [-pass password] [-epass encrypted password]</td>
<td>adds an entry to the alias table</td>
</tr>
<tr>
<td></td>
<td>You can specify -pass or -epass but you cannot use both arguments together.</td>
</tr>
<tr>
<td></td>
<td>For -pass, you must use a clear-text password. For -epass, use the sec_encrypt_p3x utility, without the -patrol_conf option, to encrypt the password. See “sec_encrypt_p3x” on page 198.</td>
</tr>
</tbody>
</table>
Options for Impersonation Table Administration

The following options apply to operations on the impersonation table.

The combination of `userPattern`, `servicePattern`, and `mlmServiceNamePattern` uniquely identify each entry in the table. For operations against the impersonation table, all of these pattern arguments are optional. If not specified, the wildcard character (*) is used for `userPattern` and `mlmServiceNamePattern`, and `PATROL_AGENT_*` is the default for `servicePattern`.

Different types of services are uniquely identified by a service prefix that must be specified with each value of `serviceNamePattern` and `mlmServiceNamePrefix`. For `serviceNamePattern`, the most commonly used prefixes are `PATROL_AGENT_` for managed systems and `PATROL_HOSTGROUP_` for managed system groups. For example, a value of `PATROL_AGENT_*` for `serviceNamePattern` maps to all PATROL Agents, and a value of `PATROL_HOSTGROUP_UNIX` maps to the managed system group UNIX.

For more information about service names, service prefixes, and service IDs, see “Configuring Service Names and Service IDs” on page 268.

**NOTE**

You cannot use wildcard characters with the command line utility to match (or act upon) multiple entries. Wildcards are read as part of the field definition. For example, the arguments `-u_pat pat* -m_pat * -s_pat PATROL_AGENT_*` create or identify a single entry.
**NOTE**

The option `-group groupName` identifies the name of a user group to be used for group-based impersonation.

The values for `userPattern` and `groupName` are treated differently at runtime. The value for `userPattern` is treated as a wildcard pattern when matching the current PATROL Console user against entries in the table.

For example, the PATROL Console user id `patrolOper` would match a row in the impersonation table with `userPattern=patrol*`. However, the values for `groupName` are treated literally. You can use either `-u_pat` or `-group`. If both are omitted, the default value will be `-u_pat *`.

If `-s_pat` is not specified, the default value is `-s_pat PATROL_AGENT_*`.

If `-m_pat` is not specified, the default value is `-m_pat *`.

If `-id uuid` is specified and the target server does not support the PATROL 7.2 impersonation backend, `admincli` exits with an error that the command is not supported on the older console server.

The `-id` and `-group` options are only available with PATROL 7.3 or later.

With PATROL 7.3 or later installations, PATROL 7.2.x syntax must be preserved for backwards compatibility.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-table imp -op ls</code></td>
<td>lists all of the entries in the impersonation table</td>
</tr>
<tr>
<td><code>-table imp -op add</code></td>
<td>adds an entry to the impersonation table</td>
</tr>
<tr>
<td><code>-table imp -op add</code></td>
<td>This form of the command will work with PATROL Console Server versions 7.2.x or later.</td>
</tr>
<tr>
<td><code>-table imp -op add</code></td>
<td>adds an entry to the impersonation table</td>
</tr>
<tr>
<td><code>-table imp -op add</code></td>
<td>The <code>-id uuid</code> argument specifies the universal unique identifier (UUID) for the new entry. If it is not specified, a new UUID is assigned to the entry. UUIDs for existing entries are included in the output from <code>-op ls</code>.</td>
</tr>
<tr>
<td><code>-table imp -op add</code></td>
<td>This form of the command only works with PATROL Console Server version 7.3 or later.</td>
</tr>
<tr>
<td>Argument</td>
<td>Function</td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
</tr>
</tbody>
</table>
| `table imp -op replace`  
[-u\_pat userPattern]  
[-s\_pat servicePattern]  
[-m\_pat mlmServiceNamePattern]  
-alias alias | replaces only the alias definition of the impersonation entry identified by the unique combination of -u\_pat,  
-s\_pat, and -m\_pat  
This form of the command only works with PATROL Console Server versions 7.2.x or later. |
| `table imp -op replace`  
-id uuid  
[-u\_pat userPattern | -group groupName]  
[-s\_pat servicePattern]  
[-m\_pat mlmServiceNamePattern]  
-alias alias | replaces one or more attributes of the impersonation entry identified by uuid  
The -id `uuid` argument specifies the universal unique identifier (UUID) for the entry that is modified.  
UUIDs for existing entries are included in the output from -op ls.  
This form of the command only works with PATROL Console Server version 7.3 or later. |
| `table imp -op remove`  
[-u\_pat userPattern | -group groupName]  
[-s\_pat servicePattern]  
[-m\_pat mlmServiceNamePattern] | removes the impersonation entry identified by a combination of -u\_pat | -group, -s\_pat, and -m\_pat  
When used without the -group argument, this form of the command works with PATROL Console Server version 7.2.x or later. |
| `table imp -op remove -id uuid` | removes the impersonation entry identified by uuid  
The -id `uuid` argument specifies the universal unique identifier (UUID) for the entry that is modified.  
UUIDs for existing entries are included in the output from -op ls.  
For the remove operation, the use of -id is mutually exclusive with the use of -u\_pat, -s\_pat, and -m\_pat options.  
This form of the command only works with PATROL Console Server version 7.3 or later. |
Options for Listing User Sessions

The following option applies to listing user sessions.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>-table session -op ls</td>
<td>lists the current log on sessions</td>
</tr>
</tbody>
</table>

Options for Managed System Group Administration

The following options apply to operations on the managed system group table only.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>-table hg -op ls</td>
<td>prints the list of managed system groups by name</td>
</tr>
<tr>
<td>-table hg -op add -name groupname [-desc description] [-label label]</td>
<td>adds an empty managed system group A description and a label can be specified.</td>
</tr>
<tr>
<td>-table hg -op remove -name groupname</td>
<td>removes the specified managed system group</td>
</tr>
<tr>
<td>-table hg -op replace -name groupname [-desc description] [-label label]</td>
<td>modifies the description or the label of the specified managed system group</td>
</tr>
<tr>
<td>-table hg -op ls_mem -name groupname</td>
<td>prints the list of managed systems in the specified managed system group</td>
</tr>
</tbody>
</table>
Options for Manually Requesting an Online Backup

Only users with the ALLOW THE USER TO INITIATE ONLINE BACKUPS right can manually request an online backup. By default, only the patadm group has this right. However, you can grant this right to individual users to meet your requirements.

The following option applies to manually requesting an online backup.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>-table hg -op add_mem -name</td>
<td>adds the specified managed system to the managed system group</td>
</tr>
<tr>
<td>groupname member membername</td>
<td></td>
</tr>
<tr>
<td>-table hg -op del_mem -name</td>
<td>removes the specified managed system from the managed system group</td>
</tr>
<tr>
<td>groupname member membername</td>
<td></td>
</tr>
</tbody>
</table>

-table config -op backup
[-inc]
[-path backup directory root]

requests an online backup from the PATROL Console Server

The -inc argument requests an incremental backup. This argument is optional. If it is not specified, a full backup is requested.

The -path argument specifies the top-level backup directory. This argument is optional. If it is not specified, the PATROL Console Server uses its pre-configured primary backup area as the top-level backup directory.

If you specify a path on Windows, make sure you type backslashes twice or use a forward slash. For example, use one of the following:

- e:\\test
- e:/test
Notes for admincli

The admincli utility communicates to the servers using the RTserver cloud. Therefore, when running the utility on a different computer from the server to be stopped, ensure that the RTSERVERS environment variable is defined and points to the correct cloud configuration.

Only one of the following may be specified: -s, -c, or -d.

The PATROL Console Server can be specified using the full service name or by using options that serve as shortcuts for certain well-known service types. When using the shortcut options, only the service id portion of the full service name must be specified.

For more information regarding service names and service IDs, see “Components That Use RTservers” on page 232. For example, both of the methods below will administer tables in the console server whose service ID is ABC:

- `$ admincli -s PATROL_CSERVER_ABC otherOptions`
- `$ admincli -c ABC otherOptions`

Both -user and -pwd must be specified, or both must be omitted.

To create encrypted passwords for alias entries, use the sec_encrypt_p3x utility without the -patrol_conf option.

To successfully complete most of the operations on ACLs and impersonation backends, users of this utility must be members of the patscadm group. Users who submit backup requests must have backup rights. Additionally, users must have the appropriate read/write/create/destroy permissions on the managed system groups object, depending on the operations on the managed system group.
Examples of Running admincli

The following example lists the contents of the alias table on the PATROL Console Server identified as ABC.

```bash
$ admincli -user patrol -pwd 82EA557C243D6605B86BE664C330628378661C11C517AA3A \ -c ABC -table alias -op ls
Connected to global name space...
Aliases table:
1: Alias:dbaAccount, Mapped User Name = user1
2: Alias:patrolAccount, Mapped User Name = patrol
```

The following example adds a single impersonation table entry on the PATROL Console Server identified as ABC. In this example, the utility prompts for a user name and password since none was specified on the command line. Also note the use of quotes to prevent the expansion of wild card characters in the pattern arguments.

```bash
$ admincli -s PATROL_CSERVER_ABC -table imp -op add -u_pat 'pat*' -s_pat 'PATROL_AGENT_*' -m_pat 'PATROL_CSERVER_*' -alias patrolAccount
Connected to global name space...

Enter a user name for the service 'PATROL_CSERVER_ABC': patrol
userId: patrol
Password required for the user 'patrol' on service 'PATROL_CSERVER_ABC'. Password:
```
confManager

Use the confManager utility to configure sections of the patrol.conf file. The patrol.conf file contains security-sensitive configuration settings for various PATROL applications. This utility enables scripted changes to the file for basic add, change, or delete operations.

Each form of the command shares the same basic characteristics: an operation (-replace, -insert, -delete), an optional target file specification (-c), and an input specification (one of either -f or -section). By default, the utility operates on the patrol.conf file in the current working directory. If the -c option is specified, then utility operates on the specified file.

For -replace operations, the entire section in the target file is replaced with the input values. If no input values are specified, then the section's contents are removed, leaving an empty section in the target file. If the specified section does not exist in the target file, then it is added.

For -insert operations, the input values are added to the target file. If the input section already exists in the patrol.conf file, then the input values are merged into the existing section in patrol.conf. Any variable definitions from the input replace existing variables in the target file.

For -delete operations, the input values identify items to be removed from the target file. If an individual variable is specified in the input, then only that variable is removed. If no variables are specified in the input, then all of the variables in the section are removed.

The input can be specified in one of two forms: an input file containing one or more sections (-f), or a single section defined on the command line (-s). Only one of these options can be specified at a time. If -s is used, then the optional options -var and -value can be used to identify a single variable name/value pair within the section.

Location of confManager

The confManager utility is located in the following directories:

- %BMC_ROOT%\common\bmc\bin\target (Windows)
- $BMC_ROOT/common/bmc/bin/target (Unix)
Syntax for confManager

```bash
confManager -replace [-c configFileName] [-f inputFileName] [-section sectionName] [-var varName] [-value valueString]
confManager -insert [-c configFileName] [-f inputFileName] [-section sectionName] [-var varName] [-value valueString]
confManager -delete [-c configFileName] [-f inputFileName] [-section sectionName] [-var varName] [-value valueString]
confManager -h
confManager -v
```

Command Line Arguments for confManager

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>-replace</td>
<td>replaces an entire section or sections</td>
</tr>
<tr>
<td>-insert</td>
<td>inserts one or more variables into an existing section</td>
</tr>
<tr>
<td></td>
<td>If a variable already exists, its value is replaced.</td>
</tr>
<tr>
<td>-delete</td>
<td>deletes one or more sections or a variable within a section</td>
</tr>
<tr>
<td>-c configFileName</td>
<td>specifies the name of the target configuration file to be modified</td>
</tr>
<tr>
<td></td>
<td>If this option is not specified, the default is the file named patrol.conf from the current working directory.</td>
</tr>
<tr>
<td>-f inputFileName</td>
<td>specifies an input file containing one or more sections defined in the same format as patrol.conf</td>
</tr>
<tr>
<td></td>
<td>May not be used with -section.</td>
</tr>
<tr>
<td>-section sectionName</td>
<td>specifies the name of a single section to operate on</td>
</tr>
<tr>
<td></td>
<td>May not be used with -f.</td>
</tr>
<tr>
<td>-var varName</td>
<td>&lt;optional&gt; defines the name of the variable to be inserted/replaced/deleted</td>
</tr>
<tr>
<td></td>
<td>This argument is only valid if -section is specified (ignored for -f).</td>
</tr>
</tbody>
</table>
Notes for confManager

You can specify only one operation at a time (-replace, -insert or -delete).

You can specify -f or -section (but not both) to identify the input sections/values. If you specify -section, then you can specify only one pair of -var/-value options.

You cannot use this utility to add, change, or delete environment variable definitions, PATROL Agent definitions, or PATROL Classic Console definitions.

To create encrypted passwords for defaultAccount entries, use the sec_encrypt_p3x utility with the -patrol_conf option (see “sec_encrypt_p3x” on page 198).

Examples of Running confManager

The following example replaces all of the section named section1 in the file /etc/patrol.d/patrol.conf with:

```
confManager -c /etc/patrol.d/patrol.conf -replace -section section1
```

The following example inserts (or updates) the defaultAccount variable in the CSERVER section.

```
confManager -c /etc/patrol.d/patrol.conf -insert -section CSERVER -var defaultAccount -value 82EA557C243D6605B86BE664C330628378661C11C517AA3A
```
The following example appends/merges all of the sections defined in the file newsection.conf into the patrol.conf file in the current working directory.

```
$ confManager -insert -f newsection.conf
Warning missing patrol.conf file, default to ./patrol.conf
```

The following example deletes all the variables in the section named `section1` from the patrol.conf file in the current working directory.

```
$ confManager -delete -section section1
Warning missing patrol.conf file, default to ./patrol.conf
```

The following example deletes only the variable named `var1` from the section named `section1` from the patrol.conf file in the current working directory.

```
$ confManager -delete -section section1 -var var1
Warning missing patrol.conf file, default to ./patrol.conf
```
**sec_encrypt_p3x**

Use the sec_encrypt_p3x utility to encrypt passwords with PATROL 3.x encryption. There are two forms of the command: one for encrypted passwords that belong in the `patrol.conf` configuration file and one for encrypted passwords used by various command line utilities. For both forms, the encrypted password is echoed to standard output.

**Location of sec_encrypt_p3x**

`sec_encrypt_p3x` is located in the following directories:

- `%BMC_ROOT%\common\bmc\bin\target\` (Windows)
- `$BMC_ROOT/common/bmc/bin/target/` (Unix)

**Syntax for sec_encrypt_p3x**

```
sec_encrypt_p3x [-patrol_conf] plainTextPassword
sec_encrypt_p3x -h
sec_encrypt_p3x -v
```

**Command Line Arguments for sec_encrypt_p3x**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>-patrol_conf</td>
<td>if specified, the encrypted output is suitable for use in the <code>patrol.conf</code> configuration file; otherwise, the output is suitable for use in various utilities that accept command line arguments for encrypted passwords</td>
</tr>
<tr>
<td></td>
<td>This option is only used to manually enter the encrypted password into the <code>patrol.conf</code> file. For example, use the sec_encrypt_p3x utility to encrypt the password without the -patrol_conf argument in the command prompt, and then copy and paste results into the <code>patrol.conf</code> file.</td>
</tr>
<tr>
<td></td>
<td>However, if a PATROL Console Server utility requires an encrypted password, then use the sec_encrypt_p3x utility without the -patrol_conf argument.</td>
</tr>
<tr>
<td>plainTextPassword</td>
<td>plain text password to be encrypted</td>
</tr>
<tr>
<td>-h</td>
<td>displays the list of arguments</td>
</tr>
<tr>
<td>-v</td>
<td>displays the version of the tool</td>
</tr>
</tbody>
</table>
**Examples of Running sec_encrypt_p3x**

The following example generates an encrypted password that can be used by the CLI tools.

```
$ sec_encrypt_p3x patrol
FB0A195D60626300
```

The following example generates an encrypted password that can be used only in the `patrol.conf` file. (This will prevent a user from reading the `patrol.conf` file and using the encrypted password to invoke command line utilities to communicate with a given server.)

```
$ sec_encrypt_p3x.exe -patrol_conf patrol
82EA557C243D6605B86BE664C330628378661C11C517AA3A
```

**stopPatrolService**

Use the stopPatrolService command line utility to stop the PATROL Console Server or other PATROL services on local or remote computers. (This utility is included with multiple products from BMC Software.)

The user must have logon permissions to the given server. If the server includes the SHUTDOWN right, the user must also have this right.

If the server requires authentication and the user does not specify a user name and password, then the utility prompts the user to enter the user name and password.

**Location of stopPatrolService**

The stopPatrolService utility is located in the following directories:

- `%BMC_ROOT%/common\bmc\bin\target` (Windows)
- `$BMC_ROOT/common/bmc/bin/target` (Unix)
Syntax for stopPatrolService

```
stopPatrolService [-user username [-pwd encryptedPassword]] fullServiceName

stopPatrolService -c consoleServerID [-user username [-pwd encryptedPassword]]

stopPatrolService -d distributionServerID [-user username [-pwd encryptedPassword]]

stopPatrolService -h

stopPatrolService -v
```

Command Line Arguments for stopPatrolService

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>-user username</td>
<td>&lt;optional&gt; specifies the user name to use when authenticating with the PATROL Console Server</td>
</tr>
<tr>
<td></td>
<td>If this option is omitted, and the server requires authentication, you are prompted to enter the information from the command line.</td>
</tr>
<tr>
<td>-pwd encryptedPassword</td>
<td>&lt;optional&gt; specifies the encrypted password to use when authenticating with the PATROL Console Server</td>
</tr>
<tr>
<td></td>
<td>You can use the sec_encrypt_p3x utility to encrypt the password (see “sec_encrypt_p3x” on page 198). If this option is omitted, and the server requires authentication, you are prompted to enter the clear text password from the command line.</td>
</tr>
<tr>
<td>-c consoleServerID</td>
<td>specifies the service ID of a PATROL Console Server</td>
</tr>
<tr>
<td>-h</td>
<td>displays the list of arguments</td>
</tr>
<tr>
<td>-v</td>
<td>displays the version of the tool</td>
</tr>
</tbody>
</table>
Notes for stopPatrolService

The stopPatrolService utility communicates to the servers using the RTserver cloud. Therefore, when running the utility on a different computer from the server to be stopped, ensure that the RTSERVERS environment variable is defined and points to the correct cloud configuration.

The PATROL Console Server can be specified using the full service name or by using options that serve as shortcuts for certain well-known service types. When using the shortcut options, only the service id portion of the full service name must be specified. For more information regarding service names and service IDs, see “Components That Use RTservers” on page 232.

For example, both of the methods below will stop the PATROL Console Server whose service ID is ABC:

- `$ stopPatrolService PATROL_CSERVER_ABC`
- `$ stopPatrolService -c ABC`

Examples of Running stopPatrolService

The following example stops the PATROL Console Server ABC, prompting you for user name and password information.

```
$ stopPatrolService -c ABC
Connected to global name space...
Shutting down /services/PATROL_CSERVER_ABC...
Enter a user name for the service 'PATROL_CSERVER_ABC': user1
userId: user1
Password required for the user 'user1' on service 'PATROL_CSERVER_ABC'.
Password: ******
done.
```
**hg_import**

Use the hg_import utility to import .mof files that contain managed system groups into the PATROL Console Server or any other server.

**Location of hg_import**

The hg_import utility is located in the following directories:

- `%BMC_ROOT%\Patrol7\bin\Windows-x86` (Windows)
- `$BMC_ROOT/Patrol7/bin/target` (Unix)

On Windows, the file name is `hg_import.exe`.

On Unix, the file name is `hg_import`.

**Syntax for hg_import**

```
hg_import -c cserverId | -s fullserviceId -data importMOFfile [importMOFfile ...] [-timeout seconds] [-user username] [-pwd password]
```

```
hg_import -h
hg_import -v
```

**Command Line Arguments for hg_import**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-c cserverId</code></td>
<td>specifies the service ID of the PATROL Console Server to which the managed system group is imported</td>
</tr>
<tr>
<td></td>
<td>If you are importing the managed system group into a PATROL Console Server, the <code>cserverId</code> specifies the host name of the PATROL Console Server.</td>
</tr>
<tr>
<td><code>-s fullserviceId</code></td>
<td>specifies the service ID of the arbitrary server to which the managed system group is imported</td>
</tr>
<tr>
<td></td>
<td>If you are importing the managed system group into an arbitrary server, the <code>fullserviceId</code> specifies the full service name of the server.</td>
</tr>
<tr>
<td><code>-data importMOFfile [importMOFfile ...]</code></td>
<td>defines the name of one or more .mof files that contain the managed system group that you are importing</td>
</tr>
</tbody>
</table>
Notes for hg_import

When managed system groups or members of managed system groups are imported into the PATROL Console Server, the PATROL Central Administration console views are not updated. You must restart the PATROL Central Administration console after running hg_import to ensure that the PATROL Central Administration console and the PATROL Console Server are synchronized.

Examples of Running hg_import

The following example loads the managed system group files managedsysgroupA.mof and managedsysgroupB.mof into the PATROL Console Server ABC.

```
hg_import -c ABC -data managedsysgroupA.mof, managedsysgroupB.mof
```
**hg_export**

Use the `hg_export` utility to export managed system groups into an external text file. The utility can export all managed system groups, a single managed system group, or a specific managed system that is a member of a managed system group.

**Location of hg_export**

The `hg_export` utility is located in the following directories:

- `%BMC_ROOT%\Patrol7\bin\Windows-x86` (Windows)
- `$BMC_ROOT/Patrol7/bin/target` (Unix)

On Windows, the file name is `hg_export.exe`.

On Unix, the file name is `hg_export`.

**Syntax for hg_export**

```
hg_export -c cserverId | -s fullserviceId [ManagedSystemGroup | ManagedSystemGroup/ManagedSystemGroupMember] [-data export MOF file] [-timeout seconds] [-user username] [-pwd password]
```

```
hg_export -h
hg_export -v
```

**Command Line Arguments for hg_export**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-c cserverId</code></td>
<td>specifies the service ID of the PATROL Console Server to which the managed system group is exported. If you are exporting the managed system group into a PATROL Console Server, the <code>cserverId</code> specifies the host name of the PATROL Console Server.</td>
</tr>
<tr>
<td><code>-s fullserviceId</code></td>
<td>specifies the service ID of the arbitrary server to which the managed system group is exported. If you are exporting the managed system group into an arbitrary server, the <code>fullserviceId</code> specifies the full service name of the server.</td>
</tr>
</tbody>
</table>
Notes for `hg_export`

The `hg_export` utility is a read-only operation.

Examples of Running `hg_export`

The following example exports the managed system group file `MyManagedSystemGroupA.mof` ...

```
hg_export -c ABC -data MyManagedSystemGroupA.mof MyManagedSystemGroup
```
Command Line Utilities for Management Profiles

The PATROL Console Server includes the following command line utilities for administering management profiles.

<table>
<thead>
<tr>
<th>Utility</th>
<th>Brief Description</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>lt_counters</td>
<td>lists management profile performance counters</td>
<td>page 206</td>
</tr>
<tr>
<td>lt_db_util</td>
<td>works with management profile .mk4 files</td>
<td>page 216</td>
</tr>
<tr>
<td>lt_import</td>
<td>imports .mof files into management profiles</td>
<td>page 218</td>
</tr>
<tr>
<td>lt_export</td>
<td>exports management profiles to .mof files</td>
<td>page 222</td>
</tr>
</tbody>
</table>

**lt_counters**

Use the lt_counters command line utility to monitor performance counters for management profiles. This utility obtains the performance counters from the specified open management profile or from the aggregated counters and displays them.

There are two columns of data in the output. The first column shows the current value of the counter. The second column shows the delta from the last value read.

By default, the data is polled every 5 seconds when monitoring individual profiles, or every 60 seconds when monitoring all profiles. The default interval can be modified by specifying the -interval argument. By default, the lt_counters utility runs continuously until the process is stopped. This behavior can be changed by specifying the number of intervals to collect before exiting; the number of intervals is specified by the -samples argument.

**Location of lt_counters**

The lt_counters utility is located in the following directories:

- %PATROL_ROOT%\bin\target (Windows)
- $PATROL_ROOT/bin/target (Unix)
Syntax for lt_counters

```
lt_counters [-cserverID cserverID] [-layout managementProfileName] 
[-user username -pwd encryptedPassword] [-timeout seconds] 
[-interval seconds] [-samples numberOfSamples] [-network_metrics] [-easy_parse]
```

```
lt_counters -h
lt_counters -v
```

Command Line Arguments for lt_counters

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>-cserverID cserverID</td>
<td>&lt;optional&gt; specifies the service ID of a PATROL Console Server If this argument is not specified, lt_counters uses the current hostname as the default cserver ID.</td>
</tr>
</tbody>
</table>
| -layout managementProfileName | specifies the management profile from which the counters should be read You must specify the runtime instance name of the management profile, which can be found in the cserver-serviceID.log file. The runtime instance name is the management profile name appended with (number). For example, for the management profile named myprofile, the runtime instance name would be similar to myprofile(2). Since the runtime instance name contains parenthesis, you must use single quotes to specify the name. For the above example, you would specify the following: 
  -layout 'myprofile(2)' If this argument is not specified, the aggregated performance counters are read (for all databases). |
<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>-user</td>
<td><code>&lt;optional&gt;</code> specifies the user name to use when authenticating with the console server</td>
</tr>
<tr>
<td></td>
<td>If this option is omitted, and the server requires authentication, you are prompted to enter the information from the command line. Both -user and -pwd can be omitted, but if one is specified the other must be specified as well.</td>
</tr>
<tr>
<td>-pwd</td>
<td><code>&lt;optional&gt;</code> specifies the encrypted password to use when authenticating with the console server</td>
</tr>
<tr>
<td></td>
<td>You can use the sec_encrypt_p3x utility without the -patrol_conf option to encrypt the password. See &quot;sec_encrypt_p3x&quot; on page 198.</td>
</tr>
<tr>
<td></td>
<td>If this option is omitted, and the server requires authentication, you are prompted to enter the clear text password from the command line. Both -user and -pwd can be omitted, but if one is specified the other must be specified as well.</td>
</tr>
<tr>
<td>-timeout seconds</td>
<td><code>&lt;optional&gt;</code> specifies the communication timeout in seconds</td>
</tr>
<tr>
<td></td>
<td>If not specified, the default value is 30 seconds.</td>
</tr>
<tr>
<td>-interval seconds</td>
<td><code>&lt;optional&gt;</code> specifies the interval duration in seconds between each collection of data</td>
</tr>
<tr>
<td></td>
<td>If not specified, the default is 5 seconds when monitoring individual profiles and 60 seconds when monitoring all profiles.</td>
</tr>
<tr>
<td>-samples numberOfSamples</td>
<td><code>&lt;optional&gt;</code> specifies the number of samples or intervals to collect before exiting</td>
</tr>
<tr>
<td></td>
<td>If not specified, lt_counters continues collecting data until it is stopped.</td>
</tr>
<tr>
<td>-network_metrics</td>
<td><code>&lt;optional&gt;</code> specifies that network I/O metrics should be included in the output</td>
</tr>
<tr>
<td></td>
<td>The network I/O metrics are independent of any single profile; they represent all of the network traffic in and out of the console server (including network messages sent back to lt_counters).</td>
</tr>
</tbody>
</table>
### Notes for lt_counters

The aggregated counters display the change from the last request and the total since the startup of PATROL Console Server. This includes all the information included in already closed profiles. The purpose of the counters is maintain information regarding the amount and type of data served by the PATROL Console Server.

You cannot reset the performance counters.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>-easy_parse</td>
<td><code>&lt;optional&gt;</code> specifies that the output should be formatted in a way that is easy to parse from PSL.</td>
</tr>
<tr>
<td></td>
<td>If this option is specified, each interval or sample is formatted as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>  EASY_PARSE_BEGIN</td>
</tr>
<tr>
<td></td>
<td>  &lt;1st field description&gt;&lt;tab&gt;&lt;value&gt;&lt;newline&gt;</td>
</tr>
<tr>
<td></td>
<td>  &lt;2nd field description&gt;&lt;tab&gt;&lt;value&gt;&lt;newline&gt;</td>
</tr>
<tr>
<td></td>
<td>  ...</td>
</tr>
<tr>
<td></td>
<td>  &lt;last field description&gt;&lt;tab&gt;&lt;value&gt;&lt;newline&gt;</td>
</tr>
<tr>
<td></td>
<td>  EASY_PARSE_END</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The -easy_parse option is mutually exclusive of the -csv option.</td>
</tr>
<tr>
<td>-h</td>
<td>displays the list of arguments</td>
</tr>
<tr>
<td>-v</td>
<td>displays the version of the tool</td>
</tr>
<tr>
<td>-csv</td>
<td><code>&lt;optional&gt;</code> writes the output in Comma Separated Value (CSV) format</td>
</tr>
<tr>
<td></td>
<td>This output can then be imported into an application such as Microsoft Excel.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The -csv option is mutually exclusive of the -easy_parse option.</td>
</tr>
</tbody>
</table>
## Example of Running `lt_counters`

The following example displays a report generated by the `lt_counters` utility.

```sh
% lt_counters -samples 3 -interval 10 -network_metrics
INFORM:9/18/2004 12:24:57 PM:::cserverId is not specified, using 'aixserver01'
INFORM:9/18/2004 12:24:57 PM:::Connected to global name space...

Enter a user name for the service 'PATROL_CSERVER_AIXSERVER01' [patrol] :
Password required for the user 'patrol' on service 'PATROL_CSERVER_AIXSERVER01'.
Password:
INFORM:9/18/2004 12:25:02 PM:::Retrieving performance data
timestamp(Sat Sep 18 12:24:43 2004)                         1095528283
  numSessions                                                 7
  numObjects                                                  3028712
  numComputers                                                3389
  numCharts                                                   22
  numContainers                                               21
  numAgentQueries                                             22
  numLEMFilters                                               35
  numSessionsReadonly                                         0
  numSessionsReadWrite                                        7
  numAgentsUnderService                                       691
  inOverload                                                  0
  numSessionsMax                                              7
  numObjectsMax                                               3553834
  numComputersMax                                             3689

  # Events Received                                           52413913
  # Events Received - RT Dispatch                            13761695
  # Events Received - Create                                  61954
  # Events Received - Destroy                                 0
  # Events Received - Knowledge Module Create                 0
  # Events Received - Set                                     9432274
  # Events Received - State Change                            9432274
  # Events Received - Suspend                                 0
  # Events Received - SOH : Total                             38623975
  # Events Received - SOH : Job                               38502048
  # Events Received - SOH : Log                               0
  # Events Received - SOH : Task                              0
  # Events Received - SOH : Response                          0
  # Events Received - SOH : GDF                               0
  # Events Received - SOH : KMP Finish                        121927
  # Events Received - SOH : Run Command Request               0
  # Events Received - SOH : Check Right                        0

  # Managements Queries Issued                                 5156
  # Managements Query Replies Received                        50177
  # Managements Query Rows Received                           2936388

  # LEM Queries Issued                                       3494
  # LEM Query Replies Received                                31867
  # LEM Query Rows Received                                  636810
  # LEM Events Received                                      5089
```
<table>
<thead>
<tr>
<th>Command Line Utilities for Management Profiles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Chapter 5</td>
</tr>
<tr>
<td># Issued - Subscription</td>
</tr>
<tr>
<td># Issued - Execute Request</td>
</tr>
<tr>
<td># Issued - GetChildren Request</td>
</tr>
<tr>
<td># Issued - GetThis Request</td>
</tr>
<tr>
<td># Issued - History Query</td>
</tr>
<tr>
<td># Issued - Real Time Data Subscription</td>
</tr>
<tr>
<td># Issued - Connection Attempt to Management Systems</td>
</tr>
<tr>
<td># Events Send</td>
</tr>
<tr>
<td># Events Send - State Change</td>
</tr>
<tr>
<td># Events Send - Rule State Change</td>
</tr>
<tr>
<td># Events Send - Create</td>
</tr>
<tr>
<td># Events Send - Destroy</td>
</tr>
<tr>
<td># Events Send - Activity</td>
</tr>
<tr>
<td># Events Send - Log</td>
</tr>
<tr>
<td># Events Send - Management Query Add Row</td>
</tr>
<tr>
<td># Events Send - LEM Window Add Row</td>
</tr>
<tr>
<td># Events Send - Task Output</td>
</tr>
<tr>
<td># Events Send - Data Point</td>
</tr>
<tr>
<td># Events Send - Help Info</td>
</tr>
<tr>
<td># Req Received</td>
</tr>
<tr>
<td># Req Received - GetAttribute</td>
</tr>
<tr>
<td># Req Received - GetAttributes</td>
</tr>
<tr>
<td># Req Received - SetAttribute</td>
</tr>
<tr>
<td># Req Received - SetAttributes</td>
</tr>
<tr>
<td># Req Received - Execute</td>
</tr>
<tr>
<td># Req Received - GetChildren</td>
</tr>
<tr>
<td># Req Received - Create</td>
</tr>
<tr>
<td># Req Received - Destroy</td>
</tr>
<tr>
<td># Req Received - Open Resource</td>
</tr>
</tbody>
</table>

RT Cloud 'Clients - 1' messagesReceived 14125
RT Cloud 'Clients - 1' totalBytesReceived 5860504
RT Cloud 'Clients - 1' minMsgSizeReceived 40
RT Cloud 'Clients - 1' maxMsgSizeReceived 2240
RT Cloud 'Clients - 1' incomingBufferCurrentSize 0
RT Cloud 'Clients - 1' incomingBufferMaximunSize 163840
RT Cloud 'Clients - 1' messagesSent 4255985
RT Cloud 'Clients - 1' totalBytesSent 3231539480
RT Cloud 'Clients - 1' minMsgSizeSent 16
RT Cloud 'Clients - 1' maxMsgSizeSent 293840
RT Cloud 'Clients - 1' outgoingBufferCurrentSize 0
RT Cloud 'Clients - 1' outgoingBufferMaximunSize 2736
RT Cloud 'Clients - 1' rateBytesRecv 0.034824
RT Cloud 'Clients - 1' rateMsgsRecv 0.077388
RT Cloud 'Clients - 1' rateMsgsSent 0.077388
RT Cloud 'Clients - 1' elapsedSeconds 129.219423

RT Cloud 'Agents - 1' messagesReceived 28987221
RT Cloud 'Agents - 1' totalBytesReceived 34714132304
RT Cloud 'Agents - 1' minMsgSizeReceived 128
RT Cloud 'Agents - 1' maxMsgSizeReceived 36984
RT Cloud 'Agents - 1' incomingBufferCurrentSize 0
RT Cloud 'Agents - 1' incomingBufferMaximunSize 183424
RT Cloud 'Agents - 1' messagesSent 1701001
RT Cloud 'Agents - 1' totalBytesSent 960622728
RT Cloud 'Agents - 1' minMsgSizeSent 16
RT Cloud 'Agents - 1' maxMsgSizeSent 2736
<table>
<thead>
<tr>
<th>Metric</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>RT Cloud 'Agents - 1' outgoingBuffer Current Size</td>
<td>0</td>
</tr>
<tr>
<td>RT Cloud 'Agents - 1' outgoingBuffer Max numSize</td>
<td>65552</td>
</tr>
<tr>
<td>RT Cloud 'Agents - 1' rateBytesRecv</td>
<td>280.693822</td>
</tr>
<tr>
<td>RT Cloud 'Agents - 1' rateMsgsRecv</td>
<td>242.239125</td>
</tr>
<tr>
<td>RT Cloud 'Agents - 1' rateBytesSent</td>
<td>5.480503</td>
</tr>
<tr>
<td>RT Cloud 'Agents - 1' rateMsgsSent</td>
<td>242.239125</td>
</tr>
<tr>
<td>RT Cloud 'Agents - 1' elapsedSeconds</td>
<td>129.219423</td>
</tr>
</tbody>
</table>

InFORM: 9/18/2004 12:25:12 PM: Retrieving performance data

timestamp (Sat Sep 18 12:24:43 2004) 1095528283 0 (0/sec)
numSessions 7 0 (0/sec)
numObjects 3028712 0 (0/sec)
numComputers 3389 0 (0/sec)
numCharts 22 0 (0/sec)
numContainers 21 0 (0/sec)
nmAgentQuerries 77 0 (0/sec)
nmLEMFilters 35 0 (0/sec)
nmSessionsReadOnly 0 0 (0/sec)
nmSessionsReadWrite 7 0 (0/sec)
nmAgentsUnderService 691 0 (0/sec)
inOverload 0 0 (0/sec)
numSessionsMax 7 0 (0/sec)
numObjectsMax 3553834 0 (0/sec)
numComputersMax 3689 0 (0/sec)

# Events Received 52413913 0 (0/sec)
# Events Received - RT Dispatch 13761695 0 (0/sec)
# Events Received - Create 61954 0 (0/sec)
# Events Received - Destroy 0 0 (0/sec)
# Events Received - Knowledge Module Create 0 0 (0/sec)
# Events Received - Set 9432274 0 (0/sec)
# Events Received - State Change 9432274 0 (0/sec)
# Events Received - Suspend 0 0 (0/sec)
# Events Received - SOH : Total 38623975 0 (0/sec)
# Events Received - SOH : Job 38502048 0 (0/sec)
# Events Received - SOH : Log 0 0 (0/sec)
# Events Received - SOH : Task 0 0 (0/sec)
# Events Received - SOH : Response 0 0 (0/sec)
# Events Received - SOH : GDF 0 0 (0/sec)
# Events Received - SOH : KMP Finish 121927 0 (0/sec)
# Events Received - SOH : Run Command Request 0 0 (0/sec)
# Events Received - SOH : Check Right 0 0 (0/sec)

# Managements Queries Issued 5156 0 (0/sec)
# Managements Query Replies Received 50177 0 (0/sec)
# Managements Query Rows Received 2936388 0 (0/sec)

# LEM Queries Issued 3494 0 (0/sec)
# LEM Query Replies Received 31867 0 (0/sec)
# LEM Query Rows Received 636810 0 (0/sec)
# LEM Events Received 5089 0 (0/sec)

# Issued - Subscription 379354 0 (0/sec)
# Issued - Execute Request 67733141 0 (0/sec)
# Issued - GetChildren Request 474518 0 (0/sec)
# Issued - Get this Request 3876941 0 (0/sec)
# Issued - History Query 317 0 (0/sec)
# Issued - Real Time Data Subscription 47 0 (0/sec)
# Issued - Connection Attempt to Management Systems 130951 0 (0/sec)
# Events Send  12867760  0 (0/sec)
# Events Send - State Change  8617215  0 (0/sec)
# Events Send - Rule State Change  431966  0 (0/sec)
# Events Send - Create  162779  0 (0/sec)
# Events Send - Destroy  3047487  0 (0/sec)
# Events Send - Activity  1800  0 (0/sec)
# Events Send - Log  0  0 (0/sec)
# Events Send - Management Query Add Row  47925  0 (0/sec)
# Events Send - LEM Window Add Row  36956  0 (0/sec)
# Events Send - Task Output  0  0 (0/sec)
# Events Send - Data Point  23273  0 (0/sec)
# Events Send - Help Info  0  0 (0/sec)

# Req Received  163670  0 (0/sec)
# Req Received - GetAttribute  0  0 (0/sec)
# Req Received - GetAttributes  824  0 (0/sec)
# Req Received - SetAttribute  0  0 (0/sec)
# Req Received - SetAttributes  1403  0 (0/sec)
# Req Received - Execute  159343  0 (0/sec)
# Req Received - GetChildren  0  0 (0/sec)
# Req Received - Create  900  0 (0/sec)
# Req Received - Destroy  1200  0 (0/sec)
# Req Received - Open Resource  0  0 (0/sec)

RT Cloud 'Clients - 1' messagesReceived 14127  2 (0.2/sec)
RT Cloud 'Clients - 1' totalBytesReceived 5861176  672 (67.2/sec)
RT Cloud 'Clients - 1' minMsgSizeReceived 40  0 (0/sec)
RT Cloud 'Clients - 1' maxMsgSizeReceived 2240  0 (0/sec)
RT Cloud 'Clients - 1' incomingBufferSizeCurrentSize 0  0 (0/sec)
RT Cloud 'Clients - 1' incomingBufferSizeMaximumSize 16384  0 (0/sec)
RT Cloud 'Clients - 1' messagesSent 4256301  316 (31.6/sec)
RT Cloud 'Clients - 1' totalBytesSent 3231746368  206888 (20688.8/sec)
RT Cloud 'Clients - 1' minMsgSizeSent 16  0 (0/sec)
RT Cloud 'Clients - 1' maxMsgSizeSent 2736  0 (0/sec)
RT Cloud 'Clients - 1' outgoingBufferSizeCurrentSize 0  0 (0/sec)
RT Cloud 'Clients - 1' outgoingBufferSizeMaximumSize 65552  0 (0/sec)
RT Cloud 'Clients - 1' rateBytesRecv 0.065374 58 (5.8/sec)
RT Cloud 'Clients - 1' rateMsgsRecv 0.199234 47 (4.7/sec)
RT Cloud 'Clients - 1' rateBytesSent 4.738823 -1 (-0.1/sec)
RT Cloud 'Clients - 1' rateMsgsSent 0.199234 47 (4.7/sec)
RT Cloud 'Clients - 1' elapsedSeconds 10.038424 -119 (-11.9/sec)

RT Cloud 'Agents - 1' messagesReceived 28990130  2909 (290.9/sec)
RT Cloud 'Agents - 1' totalBytesReceived 34717609352  3477048 (347705/sec)
RT Cloud 'Agents - 1' minMsgSizeReceived 128  0 (0/sec)
RT Cloud 'Agents - 1' maxMsgSizeReceived 3694  0 (0/sec)
RT Cloud 'Agents - 1' incomingBufferSizeCurrentSize 0  0 (0/sec)
RT Cloud 'Agents - 1' incomingBufferSizeMaximumSize 183424  0 (0/sec)
RT Cloud 'Agents - 1' messagesSent 1701094  93 (9.3/sec)
RT Cloud 'Agents - 1' totalBytesSent 960671440  48712 (4871.2/sec)
RT Cloud 'Agents - 1' minMsgSizeSent 16  0 (0/sec)
RT Cloud 'Agents - 1' maxMsgSizeSent 2736  0 (0/sec)
RT Cloud 'Agents - 1' outgoingBufferSizeCurrentSize 0  0 (0/sec)
RT Cloud 'Agents - 1' outgoingBufferSizeMaximumSize 65552  0 (0/sec)
RT Cloud 'Agents - 1' rateBytesRecv 338.25 55 58 (5.8/sec)
RT Cloud 'Agents - 1' rateMsgsRecv 289.786524 47 (4.7/sec)
RT Cloud 'Agents - 1' rateBytesSent 4.738823 -1 (-0.1/sec)
RT Cloud 'Agents - 1' rateMsgsSent 289.786524 47 (4.7/sec)
RT Cloud 'Agents - 1' elapsedSeconds 10.038424 -119 (-11.9/sec)
timestamp(Sat Sep 18 12:25:13 2004) 1095528313 30  (2.72727/sec)
umSessions 7  0  (0/sec)
umObjects   3028797  85  (7.72727/sec)
umComputers 3389  0  (0/sec)
umCharts    22  0  (0/sec)
umContainers 21  0  (0/sec)
umAgentQueries 77  0  (0/sec)
umLEMFilters 35  0  (0/sec)
umSessionsReadonly 0  0  (0/sec)
umSessionsReadWrite 7  0  (0/sec)
umAgentsUnderService 691  0  (0/sec)
inOverload 0  0  (0/sec)
umSessionsMax 7  0  (0/sec)
umObjectsMax 3553834  0  (0/sec)
umComputersMax 3689  0  (0/sec)

# Events Received      52421736  7823  (711.182/sec)
# Events Received - RT Dispatch 13762526  831  (75.5455/sec)
# Events Received - Create 61954  0  (0/sec)
# Events Received - Destroy 0  0  (0/sec)
# Events Received - Knowledge Module Create 0  0  (0/sec)
# Events Received - Set 9433105  831  (75.5455/sec)
# Events Received - State Change 9433105  831  (75.5455/sec)
# Events Received - Suspend 0  0  (0/sec)
# Events Received - SOH : Total 38630967  6992  (635.636/sec)
# Events Received - SOH : Job 38509040  6992  (635.636/sec)
# Events Received - SOH : Log 0  0  (0/sec)
# Events Received - SOH : Task 0  0  (0/sec)
# Events Received - SOH : Response 0  0  (0/sec)
# Events Received - SOH : GDF 0  0  (0/sec)
# Events Received - SOH : KMP Finish 121927  0  (0/sec)
# Events Received - SOH : Run Command Request 0  0  (0/sec)
# Events Received - SOH : Check Right 0  0  (0/sec)

# Managements Queries Issued 5156  0  (0/sec)
# Managements Query Replies Received 50177  0  (0/sec)
# Managements Query Rows Received 2936388  0  (0/sec)

# LEM Queries Issued 3494  0  (0/sec)
# LEM Query Replies Received 31867  0  (0/sec)
# LEM Query Rows Received 636810  0  (0/sec)
# LEM Events Received 5089  0  (0/sec)

# Issued - Subscription 379354  0  (0/sec)
# Issued - Execute Request 6773393  252  (22.9091/sec)
# Issued - Getchildren Request 47458  0  (0/sec)
# Issued - Getthis Request 3877144  203  (18.4545/sec)
# Issued - History Query 317  0  (0/sec)
# Issued - Real Time Data Subscription 47  0  (0/sec)
# Issued - Connection Attempt to Management Systems 130951  0  (0/sec)
# Events Send 12868660 90 (81.8182/sec)
# Events Send - State Change 8618115 900 (81.8182/sec)
# Events Send - Rule State Change 431966 0 (0/sec)
# Events Send - Create 162779 0 (0/sec)
# Events Send - Destroy 3047487 0 (0/sec)
# Events Send - Activity 1800 0 (0/sec)
# Events Send - Log 0 0 (0/sec)
# Events Send - Management Query Add Row 47925 0 (0/sec)
# Events Send - LEM Window Add Row 36956 0 (0/sec)
# Events Send - Task Output 0 0 (0/sec)
# Events Send - Data Point 23273 0 (0/sec)
# Events Send - Help Info 0 0 (0/sec)
# Req Received 163670 0 (0/sec)
# Req Received - Get Attribute 0 0 (0/sec)
# Req Received - Get Attributes 824 0 (0/sec)
# Req Received - Set Attribute 0 0 (0/sec)
# Req Received - Set Attributes 1403 0 (0/sec)
# Req Received - Execute 159343 0 (0/sec)
# Req Received - Get Children 0 0 (0/sec)
# Req Received - Create 900 0 (0/sec)
# Req Received - Destroy 1200 0 (0/sec)
# Req Received - Open Resource 0 0 (0/sec)

RT Cloud 'Clients - 1' messagesReceived 14129 2 (0.181181/sec)
RT Cloud 'Clients - 1' totalBytesReceived 5861848 672 (61.0909/sec)
RT Cloud 'Clients - 1' minMsgSizeReceived 40 0 (0/sec)
RT Cloud 'Clients - 1' maxMsgSizeReceived 2240 0 (0/sec)
RT Cloud 'Clients - 1' incomingBufferSizeCurrentSize 0 0 (0/sec)
RT Cloud 'Clients - 1' incomingBufferSizeMaxSize 163840 0 (0/sec)
RT Cloud 'Clients - 1' messagesSent 4256547 246 (22.3636/sec)
RT Cloud 'Clients - 1' totalBytesSent 3231908504 162136 (14739.6/sec)
RT Cloud 'Clients - 1' minMsgSizeSent 16 0 (0/sec)
RT Cloud 'Clients - 1' maxMsgSizeSent 293840 0 (0/sec)
RT Cloud 'Clients - 1' outgoingBufferSizeCurrentSize 0 0 (0/sec)
RT Cloud 'Clients - 1' outgoingBufferSizeMaxSize 65552 0 (0/sec)
RT Cloud 'Clients - 1' rateBytesRecv 0.064208 -81 (-0.736364/sec)
RT Cloud 'Clients - 1' rateMsgsRecv 0.195682 -69 (-6.27273/sec)
RT Cloud 'Clients - 1' rateBytesSent 0.064208 -81 (-7.36364/sec)
RT Cloud 'Clients - 1' rateMsgsSent 0.195682 -69 (-6.27273/sec)
RT Cloud 'Clients - 1' elapsedSeconds 10.220689 0 (0/sec)
Use the `lt_db_util` command line utility to work with management profile `.mk4` files. You *must* stop the PATROL Console Server before running this utility. The changes you make are implemented once you restart the PATROL Console Server.

The specified management profile must already exist and the user must have read and write access for the management profile.

The value specified for `dbName` must be the full path to the management profile. Management profiles are stored in the following directories:

- `%PATROL_ROOT%\log\layout\PATROL_CSERVER_serviceID` (Windows)
- `$/PATROL_ROOT/log/layout/PATROL_CSERVER_serviceID` (Unix)

For more information about the various directories used by the PATROL Console Server, see “PATROL Console Server Directories and Files” on page 296.

### Location of `lt_db_util`

The `lt_db_util` utility is located in the following directories:

- `%PATROL_ROOT%\bin\target` (Windows)
- `$/PATROL_ROOT/bin/target` (Unix)

### Syntax for `lt_db_util`

```
lt_db_util -set_owner userid dbName
lt_db_util -info dbName
lt_db_util -h
lt_db_util -v
```
Command Line Arguments for lt_db_util

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>-set_owner userid dbName</td>
<td>changes the database owner to the specified user ID</td>
</tr>
<tr>
<td>dbName</td>
<td>must specify the full path to the management profile database file. For information about where management profiles are stored, see “PATROL Console Server Directories and Files” on page 296.</td>
</tr>
<tr>
<td>-info dbName</td>
<td>prints summary information about the management profile</td>
</tr>
<tr>
<td>dbName</td>
<td>must specify the full path to the management profile database file. For information about where management profiles are stored, see “PATROL Console Server Directories and Files” on page 296.</td>
</tr>
<tr>
<td>-h</td>
<td>displays the list of arguments and exits</td>
</tr>
<tr>
<td>-v</td>
<td>displays the version number and exit</td>
</tr>
</tbody>
</table>

Notes for lt_db_util

This utility should not be run for profiles that may be in use by the Console Server.

You must stop the PATROL Console Server before running this utility. The changes you make are implemented once you restart the PATROL Console Server.

Examples of Running lt_db_util

The following example sets the database owner to the specified user ID, patdev1.

```
% lt_db_util -set_owner patdev1 local.mk4
INFORM: 8/20/2004 6:46:50 AM:::Layout Database Checker Utility
INFORM: 8/20/2004 6:46:50 AM:::- Starting minimal checks
INFORM: 8/20/2004 6:46:50 AM:::- Retrieving the header view
INFORM: 8/20/2004 6:46:50 AM:::- All checks passed
INFORM: 8/20/2004 6:46:50 AM:::- Starting updates
INFORM: 8/20/2004 6:46:50 AM:::- Setting owner to patdev1
INFORM: 8/20/2004 6:46:50 AM:::Closing database
```
The following example prints summary information about the management profile, local.mk4

```bash
% lt_db_util -info local.mk4
INFORM: 8/20/2004 6:43:57 AM::: Layout Database Checker Utility
INFORM: 8/20/2004 6:43:57 AM::: - Starting minimal checks
INFORM: 8/20/2004 6:43:57 AM::: - Retrieving the header view
INFORM: 8/20/2004 6:43:57 AM::: - All checks passed
INFORM: 8/20/2004 6:43:57 AM::: - Dumping basic information
INFORM: 8/20/2004 6:43:57 AM::: - The layout backed version = 7.5.0
INFORM: 8/20/2004 6:43:57 AM::: - The layout backed version = 7.5.0
INFORM: 8/20/2004 6:43:57 AM::: - The layout backed version = 7.5.0
INFORM: 8/20/2004 6:43:57 AM::: - Database owner = patrol
INFORM: 8/20/2004 6:43:57 AM::: - Maximum OID used = 13
INFORM: 8/20/2004 6:43:57 AM::: - Flags = 0x0
INFORM: 8/20/2004 6:43:57 AM::: - # Computers = 1
INFORM: 8/20/2004 6:43:57 AM::: - # Objects = 501
INFORM: 8/20/2004 6:43:57 AM::: Closing database
```

**lt_import**

Use the `lt_import` command line utility to import one or more .mof files into a new or existing management profile in the PATROL Console Server management profile database. You can use this utility to create new management profiles and to update an existing management profile.

If the specified management profile name does not exist and the user does have proper rights and permissions, a new management profile is created using all system defaults and then the data from the files is imported into the newly created management profile.

If the specified management profile does exist, and the user does have the permission to open/read/write, the profile is opened in read/write mode and the data from the files is merged. If the import data describes new objects, they are added into the management profile. If the data describes an existing object, the object is updated with the attributes from the file. Any attribute of the object that is also specified in the file is replaced during the import operation. Any attribute of the object that is not specified in the file is left as it was before the import operation.
Location of lt_import

The lt_import utility is located in the following directories:

- %PATROL_ROOT%\bin\target (Windows)
- $PATROL_ROOT/bin/target (Unix)

Syntax for lt_import

```
lt_import [-c cserverID] -layout managementProfileName -data importFileName [...] [-user username [-pwd encryptedPassword]] [-timeout timeoutInSeconds]
```

```
lt_import -h
lt_import -v
```

Command Line Arguments for lt_import

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>-cserverID cserverID</td>
<td>&lt;optional&gt; specifies the service ID of a PATROL Console Server</td>
</tr>
<tr>
<td></td>
<td>If this argument is not specified, lt_import uses the current hostname as the default cserver ID.</td>
</tr>
<tr>
<td>-layout managementProfileName</td>
<td>specifies the name of the management profile to create and/or update</td>
</tr>
<tr>
<td>-data importFileName [...]</td>
<td>specifies the name or names of the import data .mof file(s) to be imported into the management profile in the given order</td>
</tr>
<tr>
<td></td>
<td>If the data file is not in the current directory, specify using a relative or full path name.</td>
</tr>
<tr>
<td>-user username</td>
<td>&lt;optional&gt; specifies the user name to use when authenticating with the console server</td>
</tr>
<tr>
<td></td>
<td>If this option is omitted, and the server requires authentication, you are prompted to enter the information from the command line. Both -user and -pwd can be omitted, but if one is specified the other must be specified as well.</td>
</tr>
</tbody>
</table>
Notes for lt_import

If using this utility to import .mof files into a new management profile, the user must have the appropriate rights to create a new management profile. If using this utility to merge .mof files into the existing management profile, the user must have the appropriate rights to open the management profile for read/write.

Examples of Running lt_import

The following example loads a single import file into a new management profile.

```
$ lt_import -cserverId ABC -layout abc -data e:/temp/abc.mof -user patrol -pwd 9B0A1956D0626300
INFORM:3/6/03 11:31:36 AM:::Layout Database Import Utility
INFORM:3/6/03 11:31:36 AM:::Connected to global name space...
INFORM:3/6/03 11:31:36 AM:::Opened layout abc
INFORM:3/6/03 11:31:36 AM:::Initialized layout abc
INFORM:3/6/03 11:31:36 AM:::Processing file: 'e:/temp/abc.mof'
INFORM:3/6/03 11:31:37 AM:::Closed layout abc
```
The following example loads multiple import files into a new management profile.

```
$ lt_import -cserverId ABC -layout abc -data imp1.mof imp2.mof -user patrol -pwd FB0A195D60626300
INFORM:3/6/03 11:32:05 AM:::Layout Database Import Utility
INFORM:3/6/03 11:32:06 AM:::Connected to global name space...
INFORM:3/6/03 11:32:06 AM:::Opened layout abc
INFORM:3/6/03 11:32:06 AM:::Initialized layout abc
INFORM:3/6/03 11:32:06 AM:::Processing file: 'imp1.mof'
INFORM:3/6/03 11:32:06 AM:::Processing file: 'imp2.mof'
INFORM:3/6/03 11:32:06 AM:::Closed layout abc
```

The following example loads a single import file, prompting the user for user name and password information.

```
$ lt_import -cserverId ABC -layout abc -data e:/temp/abc.mof
INFORM:3/6/03 11:32:37 AM:::Layout Database Import Utility
INFORM:3/6/03 11:32:37 AM:::Connected to global name space...
Enter a user name for the service 'PATROL_CSERVER_ABC': patrol
userId: patrol
Password required for the user 'patrol' on service 'PATROL_CSERVER_ABC'.
Password: *******
INFORM:3/6/03 11:32:46 AM:::Opened layout abc
INFORM:3/6/03 11:32:46 AM:::Initialized layout abc
INFORM:3/6/03 11:32:46 AM:::Processing file: 'e:/temp/abc.mof'
INFORM:3/6/03 11:32:46 AM:::Closed layout abc
```
**lt_export**

Use the `lt_export` command line utility to export a management profile to a `.mof` file.

The specified management profile must already exist and the user must have read permissions for the management profile.

You can also use the `lt_export` command line utility to export individual user-defined objects (such as custom views, charts, and folders) and their descendant objects.

To specify the object to export, use the `-object` option, followed by the full path name to the object. If the full path name includes spaces, enclose the name in quotes.

The resulting `.mof` file contains only the specified object and references to descendant objects, such as parameters in a chart. It does not contain other information, such as the PATROL Agents and KMs in the management profile. For example, if you import a `.mof` file with an exported custom view, your management profile does not contain the PATROL Agents and KMs for the parameters in the custom view, and those parameters are not displayed.

The following example exports the custom view *My Custom View* under the PATROL Main Map.

```
$lt_export -cserverId ABC -layout abc -data e:/temp/abc.mof -object "PATROL_Main_Map/My Custom View"
```

**Location of lt_export**

The `lt_export` utility is located in the following directories:

- `%PATROL_ROOT%\bin\target` (Windows)
- `$PATROL_ROOT/bin/target` (Unix)

**Syntax for lt_export**

```
lt_export [-cserverId cserverID] -layout managementProfileName -data exportFileName [-object objectPathAndName] [-user username -pwd encryptedPassword] [-timeout timeoutInSec ]
```

```
lt_export -h
lt_export -v
```
### Command Line Arguments for `lt_export`

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-cserverId cserverID</code></td>
<td><code>&lt;optional&gt;</code> specifies the service ID of a PATROL Console Server&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>If this argument is not specified, <code>lt_export</code> uses the current hostname as the default cserver ID.</td>
</tr>
<tr>
<td><code>-layout managementProfileName</code></td>
<td>specifies the name of the management profile to export</td>
</tr>
<tr>
<td><code>-data exportFileName</code></td>
<td>specifies the name of the export data <code>.mof</code> file&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>If the data file is not in the current directory, specify using a relative or full path name. This file will be overwritten with the information received from the PATROL Console Server.</td>
</tr>
<tr>
<td><code>-object objectPathAndName</code></td>
<td>specifies a user-defined folder, chart, or custom view to export&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>The <code>objectPathAndName</code> must include the full path of the object from the PATROL Main Map, and if it includes spaces it must be enclosed in quotes. For example, “PATROL_Main_Map/My Folder/My Chart”.</td>
</tr>
<tr>
<td></td>
<td>This argument was added in PATROL Console Server 7.2.36.02.</td>
</tr>
<tr>
<td><code>-user username</code></td>
<td><code>&lt;optional&gt;</code> specifies the user name to use when authenticating with the console server&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>If this option is omitted and the server requires authentication, you are prompted to enter the information from the command line. Both <code>-user</code> and <code>-pwd</code> can be omitted, but if one is specified the other must be specified as well.</td>
</tr>
<tr>
<td><code>-pwd encryptedPassword</code></td>
<td><code>&lt;optional&gt;</code> specifies the encrypted password to use when authenticating with the console Server&lt;br&gt;</td>
</tr>
<tr>
<td></td>
<td>You use the <code>sec_encrypt_p3x</code> utility to encrypt the password. See “<code>sec_encrypt_p3x</code>” on page 198.</td>
</tr>
<tr>
<td></td>
<td>If this option is omitted, and the server requires authentication, the user is prompted to enter the clear text password from the command line. Both <code>-user</code> and <code>-pwd</code> can be omitted, but if one is specified the other must be specified as well.</td>
</tr>
</tbody>
</table>
Notes for lt_export

Users of this utility must have the LOGON right for the target console server. Additionally, the user must have the read permissions for the specified profile on the console server specified.

Examples of Running lt_export

The following example exports a management profile.

```bash
$ lt_export -cserverId ABC -layout abc -data e:/temp/abc.mof -user patrol -pwd FB0A195D60626300
INFORM:3/6/03 11:31:36 AM:::Layout Database Export Utility
INFORM:3/6/03 11:31:36 AM:::Connected to global name space...
INFORM:3/6/03 11:31:36 AM:::Opened layout abc
INFORM:3/6/03 11:31:36 AM:::Initialized layout abc
INFORM:3/6/03 11:31:37 AM:::Closed layout abc
```
The following example exports a management profile, prompting the user for user name and password information.

```bash
$ lt_export -cserverId ABC -layout abc -data e:/temp/abc.mof
INFORM:3/6/03 11:32:37 AM:::Layout Database Export Utility
INFORM:3/6/03 11:32:37 AM:::Connected to global name space...
Enter a user name for the service 'PATROL_CSERVER_ABC': patrol
userId: patrol
Password required for the user 'patrol' on service 'PATROL_CSERVER_ABC'.
Password: ******
INFORM:3/6/03 11:32:46 AM:::Opened layout abc
INFORM:3/6/03 11:32:46 AM:::Initialized layout abc
INFORM:3/6/03 11:32:46 AM:::Closed layout abc
```

The following example exports the chart *My Chart* in the folder *My Folder*.

```bash
lt_export -cserverId ABC -layout abc -data e:/temp/abc.mof
-obj ect "PATROL_Main_Map/My Folder/My Chart"
```
This chapter provides information about configuring and using RTservers and configuring the components that use RTservers.

This chapter contains the following topics:

- Planning RTserver Cloud Architectures for PATROL Central Infrastructure
- Starting and Stopping the RTserver
- Components That Use RTservers
- Logical Connection Names and RTserver Communications
- The rtserver.cm File
- Configuring RTservers to Listen for Connections
- Configuring an RTserver Cloud
- Configuring RTclients to Connect to RTservers
- Configuring the Number of Client Connections
- Specifying RTserver Logging Options

- Recommended Order for Starting and Stopping the RTserver
- Starting and Stopping the RTserver on Windows
- Starting and Stopping the RTserver on Unix
- Logical Connection Names for Listening for Connections
- Logical Connection Names for Forming RTserver Clouds
- Using Multiple Logical Connection Names for Listening for Connections
- Using Multiple Logical Connection Names When Forming a Cloud
- Example: Forming an RTserver Cloud

- Editing an rtserver.cm File
- Example of an rtserver.cm File
- Configuring RTservers for Standalone Mode
- Logical Connection Names for Forming RTserver Clouds
- Using Multiple Logical Connection Names When Forming a Cloud
- Example: Forming an RTserver Cloud
- Configuring RTservers for Standalone Mode
- Logical Connection Names for Forming RTserver Clouds
- Using Multiple Logical Connection Names When Forming a Cloud
- Example: Forming an RTserver Cloud
Configuring Advanced RTserver Options ........................................ 249
Running Multiple RTservers on the Same Computer .......................... 252
  Configuring RTserver Instances .................................................. 253
Starting and Stopping Multiple RTservers on the Same Unix Computer . 255
Starting and Stopping Multiple RTservers on the Same Windows Computer. 256
Removing the Default RTserver Instance ....................................... 257
Installing Instances of RTserver as Windows Services ........................ 257
  rtservice ................................................................. 258
Using Auto-location of RTservers .................................................. 260
  Configuring Auto-location of RTservers .................................... 260
  How Auto-location of RTservers Works .................................... 261
Setting Up RTclients for RTserver Failover ..................................... 262
  Example: Setting Up RTserver Failover for RTclients ...................... 262
Configuring RTservers and RTclients in a Firewall Environment ............ 263
  How to Configure the RTserver Cloud in a Firewall Environment ....... 263
  How to Configure RTclients in a Firewall Environment ................. 263
  How to Configure the Firewall ............................................. 264
  Example: Setting Up RTservers in a Firewall Environment ............... 264
Changing the Port Number of an RTserver ...................................... 267
  Example: Changing an RTserver Port Number ................................ 267
Configuring Service Names and Service IDs .................................... 268
  Service Name Format ...................................................... 268
  How Service IDs Are Used ............................................... 269
  Choosing a Service ID ..................................................... 269
Planning RTserver Cloud Architectures for PATROL Central Infrastructure

There are several factors you must consider when you plan the overall architecture and RTserver cloud implementation to support PATROL Central Infrastructure. These factors and the overall planning process are not discussed in this guide. For guidance on how to plan and implement the RTserver cloud architecture that best suits your enterprise, see the PATROL Central Infrastructure Best Practices Guide.

Starting and Stopping the RTserver

This section contains information about starting and stopping the RTserver.

Recommended Order for Starting and Stopping the RTserver

When possible, you should start the RTserver before starting other PATROL components that use the RTserver, such as the PATROL Console Server, PATROL Central Operator, and PATROL Agents.

Similarly, you should shutdown PATROL components that use the RTserver before stopping the RTserver.
Starting and Stopping the RTserver on Windows

By default, the RTserver is started automatically as a service when you install it. However, you can start or stop it manually. This section describes how to start and stop the RTserver and verify that it is running on Windows.

**To Start or Stop the RTserver as a Service on Windows**

Start or stop the *SmartSockets RTserver service*.

**To Manually Start the RTserver from the Command Line on Windows**

1. Open a command window.
2. Change to the `%RTHOME%\bin\platform\` directory.
3. Run the following program:

   `rtserver.exe`

**To Manually Stop the RTserver from the Command Line on Windows**

Press Ctrl+C.

**To Verify That the RTserver Is Running on Windows**

1. Press Ctrl+Alt+Delete.
2. Click the *Task Manager* button.
3. On the Processes tab, look for the `rtserver.exe` process.
Starting and Stopping the RTserver on Unix

By default, the RTserver is started automatically when you install it. However, you can start or stop it manually. This section describes how to start and stop the RTserver and verify that it is running on Unix.

To Manually Start the RTserver on Unix

1. Change to the $RTHOME directory.
2. Enter the following command:

   ./start_rtserver.sh

To Verify That the RTserver Is Running on Unix

1. Enter the following command:

   ps -ef | grep rtserver

2. In the display list, look for the rtserver process.

To Manually Stop the RTserver on Unix

1. Change to the $RTHOME directory.
2. Enter the following command:

   ./stop_rtserver.sh
Components That Use RTservers

A component that uses an RTserver is an RTclient. The following PATROL components are RTclients:

- PATROL Agent
- PATROL Console Server
- PATROL Central Operator – Microsoft Windows Edition
- PATROL Central Operator – Web Edition

Logical Connection Names and RTserver Communications

An RTserver is identified by one or more logical connection name. A logical connection name is a string that includes a protocol, host, and port number, separated by colons (:) such as protocol:host:port.

An RTserver uses logical connection names to

- listen for connections from RTclients
  
  For more information, see “Configuring RTservers to Listen for Connections” on page 236.

- form an RTserver cloud with other RTservers
  
  For more information, see “Configuring an RTserver Cloud” on page 239.

RTclients use logical connection names to connect to RTservers. For more information, see “Configuring RTclients to Connect to RTservers” on page 243.

Although a logical connection name always has the same format (protocol:host:port), each part can have different values, depending on where the logical connection name is used. For more information, see the following topics:

<table>
<thead>
<tr>
<th>Topic</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical Connection Names for Listening for Connections</td>
<td>page 236</td>
</tr>
<tr>
<td>Logical Connection Names for Forming RTserver Clouds</td>
<td>page 240</td>
</tr>
<tr>
<td>Logical Connection Names for Connecting to RTservers</td>
<td>page 244</td>
</tr>
</tbody>
</table>
The rtserver.cm File

You configure an RTserver to listen for connections and form a cloud by editing the options in its rtserver.cm file. You can also configure the maximum number of client connections and specify RTserver logging options. The RTserver reads this file at startup.

This text file is located in the following directory:

- \%RTHOME\%\standard\ (Windows)
- \$RTHOME/standard/ (Unix)

For instructions on which options to edit, see the following topics:

- “Configuring RTservers to Listen for Connections” on page 236
- “Configuring an RTserver Cloud” on page 239
- “Configuring the Number of Client Connections” on page 245
- “Specifying RTserver Logging Options” on page 246

---

NOTE

The rtserver.cm file contains several options that are not covered in this chapter. These options are for use when troubleshooting or contacting customer support.

---

Editing an rtserver.cm File

You can edit an rtserver.cm file using a text editor. If editing this file on a Windows computer, use Notepad as the text editor.

If you are running multiple RTservers on the same computer, the file name is rtserver.idName.cm, where idName is an instance-specific identifier for the RTserver. For more information about multiple RTservers on the same computer, see “Running Multiple RTservers on the Same Computer” on page 252.

You do not have to stop the RTserver before editing the file; however, you must restart the RTserver before any changes take effect.

---

NOTE

It is good practice to always back up the rtserver.cm file before making any changes to it.
Example of an rtserver.cm File

The following is an example of an rtserver.cm file.

```c
//
// File: rtserver.cm
// ------------------------------------------------------
// Description: rtserver configuration options
// ------------------------------------------------------
// File Version Info : $Revision: 1.19.4.9 $
// Last check-in date: $Date: 2004/07/02 15:52:04 $
// ------------------------------------------------------
// Notes:
//
// 1. Always make a backup of this before making changes.
// 2. If editing this file on a Windows server, please use Notepad as
//    the text editor.
//
setopt prompt "SERVER> "

/* ----------------------------- */
/* BMC RTserver-specific options */
/* ----------------------------- */
setopt client_connect_timeout 600.0
setopt client_read_timeout 1200.0
setopt client_keep_alive_timeout 1200.0
setopt client_max_buffer 200000000 /* ~200 MB */
setopt log_in_client UNKNOWN
setopt log_in_server UNKNOWN
setopt log_out_client UNKNOWN
setopt log_out_server UNKNOWN

setopt server_reconnect_interval 90
setopt server_read_timeout 300
setopt server_keep_alive_timeout 150
setopt server_connect_timeout 600

setopt udp_broadcast_timeout 5.0

setopt server_names UNKNOWN
setopt conn_names tcp:_node:2059, tcp:localhost:2059

setopt server_num_threads 4
setopt max_client_conns 500
```
// RTserver uses a trace file to log system activity. The configuration options related to the trace file are:

```
setopt trace_file
C: \PROGRA~1\BMCSOF~1\common\smartsockets\log\rts\three\log
setopt trace_file_size 10000000 /*10MB*/
setopt backup_name .bak
// setopt trace_level verbose
setopt trace_flags timestamp, prefix /* Overwritten Option */
setopt time_format FULL /* Overwritten Option */
```
Configuring RTservers to Listen for Connections

This section describes how to configure an RTserver to listen for connections.

**NOTE**

By default, an RTserver is configured to listen for connections on the port specified when the RTserver was installed.

The RTserver listens for connections based on the logical connection names listed in the conn_names options in its `rtserver.cm` file (see “The `rtserver.cm` File” on page 233).

Logical Connection Names for Listening for Connections

Table 26 lists possible formats for the logical connection names in the conn_names option.

<table>
<thead>
<tr>
<th>Format</th>
<th>Example</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcp:_node:port</td>
<td>tcp:_node:2059</td>
<td>This is the recommended format for most RTservers. This format is the same as specifying the host name of the computer.</td>
</tr>
<tr>
<td>tcp:_any:port</td>
<td>tcp:_any:1988</td>
<td>Not recommended. For more information, see “Configuring RTserver to Listen for Connections on Computers with Multiple IP Addresses” on page 238.</td>
</tr>
<tr>
<td>tcp:localhost:port</td>
<td>tcp:localhost:2059</td>
<td>Use this format to configure the RTserver to listen for connections on the localhost IP address.</td>
</tr>
<tr>
<td>tcp:IPAddress:port</td>
<td>tcp:111.111.111.111:2059</td>
<td>Use this format to configure the RTserver to listen for connections on the specified IP address.</td>
</tr>
</tbody>
</table>
In the following example, the first logical connection name tells the RTserver to listen for TCP connections on port 2059 of the host computer. The second logical connection name tells the RTserver to also listen for TCP connections on port 2059 of the localhost IP address.

<table>
<thead>
<tr>
<th>Format</th>
<th>Example</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcp:hostName:port</td>
<td>tcp:myhostname:1988</td>
<td>Use this format to configure the RTserver to listen for connections on the default IP address of the computer.</td>
</tr>
<tr>
<td>udp_broadcast:_node:port</td>
<td>udp_broadcast:_node:2059</td>
<td>Use this format for auto-location of RTservers. For more information, see “Using Auto-location of RTservers” on page 260.</td>
</tr>
<tr>
<td>udp_broadcast:localhost:port</td>
<td></td>
<td></td>
</tr>
<tr>
<td>udp_broadcast:IPaddress:port</td>
<td></td>
<td></td>
</tr>
<tr>
<td>udp_broadcast:hostName:port</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The `setopt conn_names tcp:_node:2059, tcp:localhost:2059` command configures the RTserver to listen for connections on the default IP address of the computer and localhost IP address.

**Using Multiple Logical Connection Names for Listening for Connections**

The `conn_names` option can be a comma-separated list of multiple logical connection names.

Multiple logical connection names in the `conn_names` option work simultaneously, so the order of the names is unimportant. The RTserver always listens for connections on all of the names in the list.
Configuring RTserver to Listen for Connections on Computers with Multiple IP Addresses

When configuring an RTserver to listen for connections on computers with multiple IP addresses, you must use the IP address or localhost in the conn_names option in the rtserver.cm file. Using _node or the host name is not dependable because they might not resolve to the same IP address consistently.

The default version of the rtserver.cm file uses _node in the conn_names option, which cannot be used with computers with multiple IP addresses.

If you have an RTserver on a computer with multiple IP addresses, you must configure it to listen for connections by editing the conn_names option to include the logical connection names for the IP addresses on which you want the RTserver to listen.

In the following example, the RTserver listens for connections on IP addresses 111.111.111.111 and 222.222.222.222, and the localhost address on port 2059. If the computer has other IP addresses that are not included in any logical connection names, the RTserver is unavailable via those IP addresses.

```bash
setopt conn_names
```
Configuring an RTserver Cloud

This section describes how to configure multiple RTservers to form an RTserver cloud. Using an RTserver cloud provides flexibility and load balancing.

RTclients that are connected to the same RTserver cloud can communicate with each other, even if they are connected to different RTservers in the cloud. In this context, any PATROL application that communicates with the RTserver cloud is considered an RTclient.

As a general rule, if you use multiple clouds, RTclients cannot communicate with RTclients that are connected to different clouds. An exception to this rule is the ability to connect to multiple clouds with PATROL Console Server. When configured to support multiple clouds, the PATROL Console Server acts as broker between PATROL Agents and PATROL Central consoles in different RTserver clouds.

For more information about using the PATROL Console Server with multiple RTserver clouds, see “Configuring Connections to Multiple RTserver Clouds” on page 93.

You form RTserver clouds by including logical connection names of other RTservers in the server_names option in the rtserver.cm file (see “The rtserver.cm File” on page 233).

---

**NOTE**

Setting up an RTserver cloud is separate from setting up failover. To set up failover, see “Setting Up RTclients for RTserver Failover” on page 262.

---

Configuring RTservers for Standalone Mode

By default, an RTserver is configured to operate in standalone mode. A standalone RTserver does not seek out other RTservers. However, a standalone RTserver can still be part of an RTserver cloud if other RTservers point to it.

An RTserver operates in standalone mode when the value of the server_names option in the rtserver.cm file is UNKNOWN.

```bash
setopt server_names UNKNOWN
```
Configuring RTservers to Form a Cloud

To have an RTserver form a cloud with other RTservers, edit the value of the server_names option in the rtserver.cm file to include the logical connection names of the other RTservers. When configuring RTserver clouds, the best practice is to use as few RTservers as possible. Most RTserver clouds only need two RTservers. The first RTserver is the primary RTserver for all RTclients, and the second RTserver is used as a backup.

In the following example, the RTserver points to two other RTservers, one on hostA with port 2112, and another on hostB with port 5150. The three RTservers form an RTserver cloud. If the RTservers on hostA and hostB also point to additional RTservers, those servers are also part of the RTserver cloud.

```
setopt server_names tcp:hostA:2112,tcp:hostB:5150
```

Logical Connection Names for Forming RTserver Clouds

Table 27 lists possible formats for logical connection names in the server_names option.

<table>
<thead>
<tr>
<th>Format</th>
<th>Example</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcp:IPAddress:port</td>
<td>tcp:111.111.111.111:2059</td>
<td>Use this format to specify the remote computer by IP address.</td>
</tr>
<tr>
<td>tcp:hostname:port</td>
<td>tcp:myhostname:1988</td>
<td>Use this format to specify the remote computer by host name.</td>
</tr>
</tbody>
</table>

**NOTE**

The server_names option can use only the TCP connection protocol. It cannot use auto-location (see “Using Auto-location of RTservers” on page 260).
Using Multiple Logical Connection Names When Forming a Cloud

The server_names option can be a comma-separated list of multiple logical connection names.

Multiple logical connection names in the server_names option work simultaneously, so the order of the logical connection names is unimportant. The RTserver has a simultaneous connection to all of the RTservers.

If the server_names option for an RTserver points to an RTserver that is not available, the original RTserver periodically attempts to connect to the unavailable RTserver in case it becomes available.

Example: Forming an RTserver Cloud

You have four RTservers on host1, host2, host3, and host4. All of the RTservers use the default port 2059.

You want the four RTservers to form an RTserver cloud, with the RTserver on host1 as the central RTserver and the other RTservers pointing to it.

1 Leave the RTserver on host1 as a standalone server.

The server_names option in the `rtserver.cm` file of host1 is set to UNKNOWN.

```
setopt server_names UNKNOWN
```

2 Edit the `rtserver.cm` files of host2, host3, and host4 so that they all point to the RTserver on host1.

The server_names option in the `rtserver.cm` files of these hosts is set to the logical connection name of the RTserver on host1.

```
setopt server_names tcp:host1:2059
```
Figure 3 shows the arrangement of the RTservers.

**Figure 3  Example of RTServers Forming a Cloud**

```
Host2
  setopt server_names tcp:host1:2059
  setopt conn_names tcp:_node:2059,
                 tcp:localhost:2059

Host3
  setopt server_names tcp:host1:2059
  setopt conn_names tcp:_node:2059,
                 tcp:localhost:2059

Host1
  setopt server_names UNKNOWN
  setopt conn_names tcp:_node:2059,
                 tcp:localhost:2059

Host4
  setopt server_names tcp:host1:2059
  setopt conn_names tcp:_node:2059,
                 tcp:localhost:2059
```
Configuring RTclients to Connect to RTservers

This section describes how to configure RTclients, such as PATROL Agents, PATROL Console Server, and PATROL Central Operator consoles, to connect to RTservers.

RTclients connect to RTservers by using a list of logical connection names. How you provide RTclients with that list depends on the RTclient. Some RTclients can use multiple methods for specifying the list of RTservers. Common methods include the following:

- an environment variable (the most common method)
- a file
- a command line option

For example, the PATROL Agent, PATROL Console Server, and PATROL Central Operator – Microsoft Windows Edition, obtain the list of RTserver names from the RTSERVERS environment variable, which is set during the installation of these components.

You can also use the `config_rtservers.psl` script to create or update the RTSERVERS environment variable on PATROL Agent computers. See “Enabling PATROL Agent Communications with the PATROL Console Server and RTserver” on page 61. However, for PATROL Central Operator – Web Edition the list of RTservers is set in the `startup.cfg` file.

For instructions on how to set the list of RTserver names for a specific RTclient, see the documentation for that component.

For instructions on how to configure the PATROL Console Server to communicate with a single RTserver cloud, see “Configuring Connection to a Single RTserver Cloud” on page 90.

For instructions on how to configure the PATROL Console Server to communicate with multiple RTserver clouds, see “Configuring Connections to Multiple RTserver Clouds” on page 93.

**NOTE**

Although the list of RTservers in the RTSERVERS environment variable is used by many RTclients, it is not used by the RTserver itself.
Logical Connection Names for Connecting to RTservers

Table 28 lists possible formats for logical connection names used by RTclients to connect to RTservers:

Table 28 Logical Connection Names for Connecting to RTservers

<table>
<thead>
<tr>
<th>Format</th>
<th>Example</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>tcp:localhost:port</td>
<td>tcp:localhost:2059</td>
<td>Use this format to connect to an RTserver by using the localhost IP address.</td>
</tr>
<tr>
<td>tcp:IPAddress:port</td>
<td>tcp:111.111.111.111:2059</td>
<td>Use this format to connect to an RTserver by using the specified IP address.</td>
</tr>
<tr>
<td>tcp:hostname:port</td>
<td>tcp:myhostname:1988</td>
<td>Use this format to connect to an RTserver by using the specified host name.</td>
</tr>
<tr>
<td>auto:*:port</td>
<td>auto:*:2059</td>
<td>Use this format for auto-location of RTservers. For more information, see “Using Auto-location of RTservers” on page 260.</td>
</tr>
</tbody>
</table>

Using Multiple Logical Connection Names to Connect to RTservers

An RTclient can use a list of multiple logical connection names. Each logical connection name must be separated by a comma.

**NOTE**

Do not include any white space (such as spaces or tabs) in the list.

As an RTclient can be connected to only one RTserver at a time, it uses the logical connection names in its list of RTservers in order. When the RTclient starts up, it attempts to connect to the first RTserver in the list. If this RTserver does not respond, then the RTclient attempts to connect to the next RTserver in the list, and so on, until it is able to establish a connection. If the connection is lost, the RTclient repeats the process, starting with the first RTserver in the list.

For more information about setting up RTclients for RTserver Failover, see “Setting Up RTclients for RTserver Failover” on page 262.
Configuring the Number of Client Connections

Depending on the monitored environment and your RTserver hardware, a single RTserver cloud is typically sufficient for anywhere between 500–700 PATROL Agents. In the simple case of a cloud with only two RTservers (one primary and one backup), you must configure each RTserver to accept the maximum number of RTclients in the cloud.

The number of client connections includes all RTclient connections in the cloud, including PATROL Central connections, PATROL Console Server, PATROL Agents and most of the PATROL Console Server command line utilities. Therefore, when you select an appropriate value for your installation, you need to account for all of these types of connections.

Use the max_client_conns option in the rtserver.cm file to define the maximum number of client connections allowed for an RTserver. The default value for this option is 500. The maximum value specified should never exceed 750 for a single RTserver.

If the number of client connections exceeds the value defined in the max_client_conns option, you may have issues connecting to clouds with PATROL Central Operators, or some PATROL Agents may have an unreachable connection state. In addition, if you configure the value of max_client_conns to be a value greater than 750, then you may have similar types of connection problems.

If you decide that your environment needs more than 500 clients for a given RTserver, then you must modify the max_client_conns option in the rtserver.cm file to use a larger value. Using the generic example of a cloud composed of a primary and backup RTserver, the value of max_client_conns for both RTservers should be the same. In the rtserver.cm file, ensure that the value of the max_client_conns option does not exceed 750.

For instructions about how to edit the rtserver.cm file, see “Editing an rtserver.cm File” on page 233.

For more information about RTserver and RTserver cloud scalability, see the PATROL Central Infrastructure Best Practices Guide.
Specifying RTserver Logging Options

The RTserver uses a trace file to log system activity. You can configure this log file to better suit your needs. You do this by editing options that are stored in the rtserver.cm file. For instructions about how to edit the rtserver.cm file, see “Editing an rtserver.cm File” on page 233.

Table 29 lists the RTserver logging options in the rtserver.cm file that you can configure.

Table 29  RTserver Logging Options (Part 1 of 3)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| trace_file  | defines the file to which trace output is written. You can specify any valid file name, including stdout and stderr. If this option is not specified, or if the value is “unknown”, then the trace output is written to stdout. The default value is:

- %BMC_ROOT%\common\smartsockets\log\rtserver.log (Windows)
- $BMC_ROOT/common/smartsockets/log/rtserver.log (Unix)

If you have multiple RTservers on the same computer, then the format of the file name should be rtserver.idName.log, where idName is the instance-specific identifier of the RTserver. This should be different for each RTserver instance on the same machine. |
| trace_file_size | defines the maximum size, in bytes, of the trace file. When a trace file reaches the maximum size, it is backed up according to the rules specified by the backup_name option. The content of the existing trace file is deleted and new output is written to the file. There is only one level of backup. When the trace file reaches the maximum size again, the backup file is overwritten, and new output is written to the file. The default value is 10000000, which is equivalent to 10 MB. To disable this option and allow the trace file to grow indefinitely, set the value to 0. |
backup_name

defines the extension that is given to the backup trace file that is created when the file reaches the size limit specified by the trace_file_size option.

The backup file is given the same name as the existing file, with the addition of the extension specified in this option.

For example, if the value of backup_name is `.bak` and the trace file name is `rtserver.log`, then the backup file is given the name `rtserver.log.bak`.

The default value is `.bak`.

There is only one level of backup. Every time the trace file reaches the maximum size set in the trace_file_size option, the existing backup file is overwritten.

In addition, a backup file is created each time the RTserver starts. In this case, the backup file is given the same name as the existing file, with the addition of a tilde “~” to the end of the file name. For example, if the trace file name is `rtserver.log`, then the backup file is given the name `rtserver.log~`.
Specifying RTserver Logging Options

On Unix, the `start_rtserver.log` file is created to capture unexpected startup problems. This file is located in the `$BMC_ROOT/common/smartsockets/log` directory. However, all other information is written to the file identified by the `trace_file` option.

### Table 29  RTserver Logging Options (Part 3 of 3)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>trace_level</code></td>
<td>defines the amount of information that is written to the trace file</td>
</tr>
<tr>
<td></td>
<td>The available values for this option, in order of increasing information, are:</td>
</tr>
<tr>
<td></td>
<td>- never—no information is written to the trace file</td>
</tr>
<tr>
<td></td>
<td>- error—only error messages are written to the trace file</td>
</tr>
<tr>
<td></td>
<td>- warning—error and warning messages are written to the trace file</td>
</tr>
<tr>
<td></td>
<td>- info—general informational output such as clients connecting, disconnecting, and subscribing are written to the trace file</td>
</tr>
<tr>
<td></td>
<td>- info_1—thread information for multi-threading is written to the trace file</td>
</tr>
<tr>
<td></td>
<td>- info_2</td>
</tr>
<tr>
<td></td>
<td>- verbose</td>
</tr>
<tr>
<td></td>
<td>- verbose_1</td>
</tr>
<tr>
<td></td>
<td>- verbose_2</td>
</tr>
<tr>
<td></td>
<td>- debug—the maximum amount of information is written to the trace file</td>
</tr>
</tbody>
</table>

The default value is `warning`. 

On Unix, the `start_rtserver.log` file is created to capture unexpected startup problems. This file is located in the `$BMC_ROOT/common/smartsockets/log` directory. However, all other information is written to the file identified by the `trace_file` option.
Configuring Advanced RTserver Options

The options defined in this section rarely need to be modified. In some cases where the RTserver has problems maintaining connectivity to other RTservers or accepting large volumes of query data, some of these options may need to be adjusted.

These options are stored in the rtserver.cm file. For instructions about how to edit the rtserver.cm file, see “Editing an rtserver.cm File” on page 233.

Table 31 lists the advanced RTserver options you may want to configure.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>server_reconnect_interval</td>
<td>specifies the number of seconds an RTserver waits between attempts to reconnect to another RTserver in its server_names option</td>
</tr>
<tr>
<td></td>
<td>An RTserver tries to reconnect immediately after losing contact with another RTserver process. If that attempt fails, the RTserver process waits the number of seconds specified in server_reconnect_interval before attempting to reconnect.</td>
</tr>
<tr>
<td></td>
<td>An RTserver also starts the reconnect cycle if it cannot open an initial connection to another RTserver process.</td>
</tr>
<tr>
<td></td>
<td>An RTserver makes no attempt to reconnect when server_reconnect_interval is set to 0.0.</td>
</tr>
<tr>
<td></td>
<td>The default value is 90.0.</td>
</tr>
</tbody>
</table>
server_read_timeout helps determine whether an RTserver process sends a KeepAlive.

A KeepAlive is a message sent by an RTserver process to another RTserver process. By responding to a KeepAlive, an RTserver process signals that it is available to exchange information. When an RTserver process has an open connection with another, it measures the amount of time that has passed since their last communication.

If this interval is greater than the number of seconds specified by server_read_timeout, the RTserver process sends a KeepAlive.

You can disable KeepAlive by setting server_read_timeout to 0.0.

The default value is 300.0, which is equivalent to 300 seconds.

server_keep_alive_timeout specifies the number of seconds an RTserver waits for a response after sending a KeepAlive to another RTserver process.

If the receiver of the KeepAlive does not respond within server_keep_alive_timeout seconds, the sender terminates the connection. The sender of the KeepAlive might try to reconnect depending on the value of server_reconnect_interval.

You can disable KeepAlive by setting server_keep_alive_timeout to 0.0.

The default value is 150.0.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>server_read_timeout</td>
<td>helps determine whether an RTserver process sends a KeepAlive.</td>
</tr>
<tr>
<td></td>
<td>A KeepAlive is a message sent by an RTserver process to another RTserver process. By responding to a KeepAlive, an RTserver process signals that it is available to exchange information. When an RTserver process has an open connection with another, it measures the amount of time that has passed since their last communication. If this interval is greater than the number of seconds specified by server_read_timeout, the RTserver process sends a KeepAlive. You can disable KeepAlive by setting server_read_timeout to 0.0. The default value is 300.0, which is equivalent to 300 seconds.</td>
</tr>
<tr>
<td>server_keep_alive_timeout</td>
<td>specifies the number of seconds an RTserver waits for a response after sending a KeepAlive to another RTserver process. If the receiver of the KeepAlive does not respond within server_keep_alive_timeout seconds, the sender terminates the connection. The sender of the KeepAlive might try to reconnect depending on the value of server_reconnect_interval. You can disable KeepAlive by setting server_keep_alive_timeout to 0.0. The default value is 150.0.</td>
</tr>
</tbody>
</table>
### Table 30  Advanced RTserver Options (Part 3 of 3)

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| client_max_buffer | specifies the maximum number of bytes of data that an RTserver process can buffer  
An RTserver process buffers data when it is trying to send data to a PATROL component, such as a PATROL Console Server or another RTserver process, that is too busy to accept the data. When the component is no longer busy, the RTserver process sends the data, which empties the buffer.  
The client_max_buffer option exists to keep the buffer from becoming too big, which could crash the RTserver process by consuming all of its memory and disk space.  
When the size of the buffer reaches client_max_buffer bytes, the RTserver process displays a buffer overflow message and purges some data to maintain the buffer size limit.  
If client_max_buffer is set too low, the buffer overflow message is often displayed because minor problems, such as slightly elevated network traffic, trigger the alarm. If set too low, the RTserver process can crash or cause poor performance.  
This option accepts an integer value.  
The default is 200000000, which is equivalent to 200 MB. |
Running Multiple RTservers on the Same Computer

To help minimize the amount of hardware required in large environments, you can run more than one RTserver concurrently on the same computer, as long as the configuration for each instance of RTserver is independent of each other.

For Windows, only two RTservers can run concurrently on the same computer.

When running more than one RTserver on the same computer, each instance of RTserver must be started with a unique identifier. The unique identifier is used to locate each instance-specific configuration file (rtserver.idName.cm) so that multiple instances do not conflict with each other. The following sections explain how to configure separate instances of an RTserver.

Table 31 lists the minimum options you should configure for each RTserver to run multiple servers on the same computer.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| conn_names     | defines the port that the RTserver uses to accept connections for clients and other RTservers  
|                | The value of the conn_names option should not collide with another RTserver on the same computer. It should be unique for each instance of RTserver on the same computer.  
|                | The default value is tcp:_node:2059, tcp:localhost:2059.  |
| server_names   | defines the list of 0 or more RTservers to which a particular instance of RTserver should connect to form a cloud  
|                | Depending on the local topology of each cloud, you may need to change the value of the server_names option for the RTserver role to avoid collision.  
|                | The default value is UNKNOWN.  |
Configuring RTserver Instances

All instance-specific settings should be defined in an instance-specific configuration file using the following naming convention:

- `%RTHOME%\common\smartsockets\standard\rtserver.idName.cm` (Windows)
- `$RTHOME/common/smartsockets/standard/rtserver.idName.cm` (Unix)

where *idName* is an instance-specific identifier that does not contain spaces. This identifier is used to start and stop each instance of RTserver. For more information about starting and stopping instances of RTservers, see “Starting and Stopping Multiple RTservers on the Same Unix Computer” on page 255 or “Starting and Stopping Multiple RTservers on the Same Windows Computer” on page 256.
To Configure an RTserver Instance

1. Copy `rtserver.cm` to `rtserver.idName.cm`.

2. In the `rtserver.idName.cm` file, change the value of the following options:
   - conn_names
   - server_names
   - trace_file

   For more information about the options to be edited, see Table 31 on page 252.

3. In the `rtserver.cm` file, comment out or delete each instance-specific option that you defined in `rtserver.idName.cm` to avoid collisions.

---

**NOTE**

In addition to the options listed in this task, you can change other configuration options on a per-instance basis. However, retain common settings in the `rtserver.cm` file to simplify maintenance.

For instructions about how to edit the configuration file, see “Editing an rtserver.cm File” on page 233.
Starting and Stopping Multiple RTservers on the Same Unix Computer

You can start and stop more than one RTserver on the same Unix computer.

Since each instance-specific option defined in `rtserver.idName.cm` should be commented out of or deleted from the `rtserver.cm` file, it is not possible to use the `rtserver.cm` configuration file by itself. You must use the `start_rtserver.sh` and `stop_rtserver.sh` scripts with the `-id` argument on computers that are configured to run more than one instance of RTserver.

**Before You Begin**

You must have created at least one instance-specific configuration file. For more information, see “Configuring RTserver Instances” on page 253.

**To Start an Instance of RTserver on Unix**

1. Change to the `$RTHOME` directory.
2. Enter the following command:

   `%start_rtserver.sh -id idName`

**To Stop an Instance of RTserver on Unix**

1. Change to the `$RTHOME` directory.
2. Enter the following command:

   `%stop_rtserver.sh -id idName`
Starting and Stopping Multiple RTservers on the Same Windows Computer

On a Windows 2000 and Windows Server 2003 computer, the RTserver is started and stopped through the Services MMC snap-in. Therefore, you must install the different instances of RTserver as Windows services. Once you have installed the different instances of RTserver as Windows services, you can stop or start them like any other Windows service.

For Windows, only two RTservers can run concurrently on the same computer.

Since each instance-specific option defined in rtserver.idName.cm should be commented out of or deleted from the rtserver.cm file, it is not possible to use the rtserver.cm configuration file by itself. You must remove the default Windows service installation for RTserver in order to run more than one instance of RTserver on the same Windows computer.

Before You Begin

You must remove the default RTserver instance. For instructions, see “To Remove the Default RTserver Instance” on page 257.

You must install a unique instance of RTserver as a Windows service. For instructions, see “To Install a Unique Instance of RTserver as a Windows Service” on page 257.

Use the rtservice utility to register different instances so that each instance of RTserver has a unique service name and description. For more information about the rtservice utility, see “rtservice” on page 258.

To Start or Stop an Instance of RTserver on Windows

Start or stop the SmartSockets RTserver idName service.
Removing the Default RTserver Instance

The RTserver instance installed by default has no specific instance id. You must remove this default instance of RTserver from the Windows service registry in order to avoid conflicts with the ID-specific instances of RTserver on the same computer.

**To Remove the Default RTserver Instance**

Open a CMD window and enter the following command:

```
C> rtserver -uninstall
```

Installing Instances of RTserver as Windows Services

Since the RTserver is started and stopped through the Services MMC snap-in on Windows 2000 and Windows Server 2003 computers, you must install different instances of RTserver as Windows services.

**To Install a Unique Instance of RTserver as a Windows Service**

Open a CMD window and enter the following command:

```
C> rtserver -install -id idName
```

where *idName* is the unique identifier of the instance of RTserver.

To run multiple instances of RTserver, run the command several times with the different unique identifiers. Each time you run the command, the Windows registry is updated with another RTserver service.
rtservice

This section describes the rtservice utility.

**Description of rtservice**

The `rtservice` utility installs one or more RTservers as Windows services and is used to manage several instances of RTserver on the same Windows computer. The instances of RTserver are installed to the Windows Service Control Manager (SCM). This utility is also used to uninstall instances of RTserver from the Windows SCM.

**NOTE**

The `rtservice` install and uninstall activities are not related to installing and uninstalling the RTserver product itself.

Each instance of RTserver must be uniquely identified with the Windows SCM and must have its own unique RTserver configuration file. For more information, see “Configuring RTserver Instances” on page 253.

The `-id` command line argument provides the unique identifier for each instance of RTserver. The `idName` value of the argument is used in three ways:

- The Windows service name is constructed as `rtserver_idName`. This is the name that is used by Windows SCM to uniquely identify each Windows service. The name can be viewed on the Windows Properties page.

- The Windows Service Display Name is constructed as `SmartSockets RTserver idName`. This is the name that is displayed in the Windows SCM.

- The service is registered with an extra command line argument that tells the RTserver to use the `rtserver.idName.cm` configuration file.

**Location of rtservice**

The `rtservice` utility is located in the following directory:

```
%BMC_ROOT%\common\smartsockets\bin\target
```
## Syntax for rtservice

<table>
<thead>
<tr>
<th>Argument</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>rtservice -install -id idName</td>
<td>installs the RTserver using the specified ID as part of the Windows Service Name and as part of the RTserver command line options</td>
</tr>
<tr>
<td>rtservice -uninstall -id idName</td>
<td>uninstalls the Windows service for the specified RTserver</td>
</tr>
<tr>
<td>rtservice -id idName</td>
<td>specifies a unique identifier that is part of both the Windows Service Name and the command line options that are passed to the RTserver when it starts</td>
</tr>
<tr>
<td>rtservice -h</td>
<td>prints a usage statement and exits the rtservice utility</td>
</tr>
<tr>
<td>rtservice -v</td>
<td>prints the version number of the rtservice utility and exits the utility</td>
</tr>
</tbody>
</table>

## Command Line Arguments for rtservice

The following command registers an instance of the RTserver that will read its configuration from the rtserver.primary.cm file. This instance of RTserver is listed as SmartSockets RTserver -primary in the Windows Service Control Panel.

C> rtservice -install -id primary
Using Auto-location of RTservers

RTclients can use auto-location to discover all active RTservers that use a given port and are on the same subnet as the RTclient computer.

**NOTE**
Auto-location of RTservers *cannot* be used by the RTserver itself to form RTserver clouds.

**WARNING**
Use auto-location with caution if you have more than one RTserver cloud. If an RTclient that normally connects to an RTserver in one cloud discovers and connects to an RTserver outside of that cloud, that RTclient will no longer be able to communicate with other RTclients that are still connected to the original cloud.

Configuring Auto-location of RTservers

To use auto-location, you must configure both the RTserver and RTclient as follows:

- For all RTservers that you want to respond to auto-location requests, use the logical connection name `udp_broadcast_node:port` in the `conn_names` option. (See “Configuring RTservers to Listen for Connections” on page 236.) This logical connection name tells the RTserver to listen for UDP auto-location requests on the specified port.

- For all RTclients for which you want to use auto-location, use the logical connection name `auto:*:port` in the list of RTservers. (See “Configuring RTclients to Connect to RTservers” on page 243). This logical connection name tells the RTclient to discover, by means of a UDP broadcast, all RTservers that use the given port and are on the local subnet.
How Auto-location of RTservers Works

The following process describes how auto-location works:

1. The RTclient reaches the auto:*:port logical connection name, according to the order-of-evaluation rules. (See “Using Multiple Logical Connection Names to Connect to RTservers” on page 244.)

2. The RTclient sends out a UDP broadcast to discover all active RTservers on the local subnet with the specified port number.

3. Any RTservers on the local subnet with the udp_broadcast:_node:port logical connection name and a corresponding port number respond to the RTclient.

4. The RTclient expands the auto:*:port logical connection name to use the TCP protocol with the names of the discovered RTservers and the specified port number.

5. Once the auto-locator string is expanded, the RTclient continues to attempt to connect to an RTserver, using the same order-of-evaluation rules.

**NOTE**

Because the udp_broadcast:_node:port logical connection name is used only when RTclients are discovering RTservers, the RTserver still needs at least one additional logical connection name with the TCP protocol for listening for regular connections.
Setting Up RTclients for RTserver Failover

This section describes how to set up RTclients for RTserver failover. For RTclients to have RTserver failover, each RTclient must point to two or more RTservers.

If the first RTserver fails, the RTclient then uses the second RTserver. For more information, see “Using Multiple Logical Connection Names to Connect to RTservers” on page 244.

Example: Setting Up RTserver Failover for RTclients

You have four RTservers: host1, host2, host3, and host4. These RTservers form a cloud, as in “Example: Forming an RTserver Cloud” on page 241.


3 On the PATROL Agent computers, set the RTSERVERS environment variable to tcp:host3:2059,tcp:host4:2059.

If the RTserver on either host2 or host3 fails, the corresponding RTclients will switch to RTserver4. For more information, see “Using Multiple Logical Connection Names to Connect to RTservers” on page 244.
Configuring RTservers and RTclients in a Firewall Environment

If your PATROL environment includes firewall protection, you need to configure your RTserver cloud, RTclients (such as PATROL Agents and PATROL Console Server), and firewall to work together.

How to Configure the RTserver Cloud in a Firewall Environment

You install at least one RTserver on each side of the firewall. You configure the cloud by having an RTserver inside the firewall point to an RTserver outside the firewall. This configuration causes the connection to be initiated from inside the firewall.

For more information about configuring a cloud, see “Configuring an RTserver Cloud” on page 239.

How to Configure RTclients in a Firewall Environment

You configure RTclients to connect to RTservers on their own side of the firewall.

- You configure RTclients inside the firewall to connect to RTservers inside the firewall.
- You configure RTclients outside the firewall to connect to RTservers outside of the firewall.

For more information about configuring RTclients to connect to RTservers, see “Configuring RTclients to Connect to RTservers” on page 243.
How to Configure the Firewall

You configure the firewall to allow RTserver communications, much like you configure the firewall for other TCP/IP programs.

The firewall must allow outbound TCP connections from the RTserver inside the firewall to the RTserver outside the firewall on TCP port 2059 (or the port that they are configured to use). The connection between the RTservers is kept as long as both RTservers are running.

For specific instructions on how to configure the firewall, consult your firewall documentation.

Example: Setting Up RTservers in a Firewall Environment

You have several PATROL components and four RTservers inside a firewall. The four RTservers (host1, host2, host3, and host4) form a cloud, as in “Example: Forming an RTserver Cloud” on page 241.

You have a PATROL Agent outside the firewall that you want to use with the PATROL components inside the firewall.

1 Add an RTserver on host5 outside the firewall. Let it act standalone (the default).

```
setopt server_names UNKNOWN
setopt conn_names tcp:_node:2059;tcp:localhost:2059
```

2 Configure the PATROL Agent outside the firewall to connect to the RTserver outside the firewall, by setting the PATROL Agent’s RTSERVERS environment variable to tcp:host5:2059.

3 Change the RTserver on host1 to point to the RTserver on host5 outside the firewall.

```
setopt server_names tcp:host5:2059
setopt conn_names tcp:_node:2059;tcp:localhost:2059
```
Configure the firewall so that TCP traffic can pass outbound from host1, inside the firewall, to host5, outside the firewall, on port 2059.

Figure 4 illustrates the setup for the firewall example.

**Figure 4**   Example Firewall Setup on Port 2059

You can also use a different port to cross the firewall. Figure 5 illustrates the setup for the firewall example, if it used an alternate port number, 2000, to cross the firewall. Note that the default port 2059 can still be used for other RTservers in the cloud that do not communicate across the firewall.
Figure 5  Example Firewall Setup on Port 2000

Inside the Firewall

RTserver on Host2
setopt server_names tcp:host1:2059
setopt conn_names tcp:_node:2059,
tcp:localhost:2059

RTserver on Host1
setopt server_names tcp:host5:2000
setopt conn_names tcp:_node:2059,
tcp:localhost:2059

RTserver on Host3
setopt server_names tcp:host1:2059
setopt conn_names tcp:_node:2059,
tcp:localhost:2059

RTserver on Host4
setopt server_names tcp:host1:2059
setopt conn_names tcp:_node:2059,
tcp:localhost:2059

Outside the Firewall

PATROL Agent
RTSERVERS=tcp:host5:2000

RTserver on Host5
setopt server_names UNKNOWN
setopt conn_names tcp:_node:2000,
tcp:localhost:2000
Changing the Port Number of an RTserver

Changing the port number of an RTserver involves changing the following:

- the port number on which the RTserver listens for connections (see “Configuring RTservers to Listen for Connections” on page 236)
- the port number that other RTservers use when forming a cloud (see “Configuring an RTserver Cloud” on page 239)
- the port number to which RTclients connect (see “Configuring RTclients to Connect to RTservers” on page 243)

Example: Changing an RTserver Port Number

You have two RTservers, on hostA and hostB, that form a cloud. Both use the default port of 2059.

The rtserver.cm file for hostA contains the following values for the server_names and conn_names options:

```
setopt server_names UNKNOWN
setopt conn_names tcp:_node:2059, tcp:localhost:2059
```

The rtserver.cm file for hostB contains the following values for the server_names and conn_names options:

```
setopt server_names tcp:hostA:2059
setopt conn_names tcp:_node:2059, tcp:localhost:2059
```

The list of RTservers for the RTclients is tcp:hostA:2059, tcp:hostB:2059.

You want to change the RTserver on hostA to use port 1988, but you want the RTserver on hostB to continue using port 2059.
Configuring Service Names and Service IDs

1 Edit the `rtserver.cm` file of the RTserver on hostA and restart the RTserver.

Because you want the RTserver on hostA to listen for connections on a new port, you change its conn_names option.

```bash
setopt server_names UNKNOWN
```

2 Edit the `rtserver.cm` file of the RTserver on hostB and restart the RTserver.

Because the RTserver on hostB must connect to hostA on a new port, you change the server_names option:

```bash
setopt server_names tcp:hostA:1988
setopt conn_names tcp:_node:2059,tcp:localhost:2059
```

3 Edit the list of RTservers for the RTclients to be `tcp:hostA:1988,tcp:hostB:2059` and restart them (unless they do not need to be restarted for the change to take effect).

Configuring Service Names and Service IDs

RTclients that act as PATROL servers have service names. For example, the PATROL Agent and PATROL Console Server are PATROL servers that have service names.

Each service name uniquely identifies the RTclient to the RTserver cloud. Each PATROL server in the same RTserver cloud must have a unique service name.

Service Name Format

A service name is the combination of a service prefix, followed by an underscore, followed by a service ID. A service prefix is a hard-coded prefix based on the type of PATROL server. A service ID is a user-configurable service identifier that varies from one PATROL server to another.

For example, in the service name `PATROL_CS蠲E_SERVER_MYHOST`, the service prefix is `PATROL_CS蠲E_SERVER`, and the service ID is `MYHOST`. 
Table 32 lists the service prefixes and default service IDs for PATROL servers:

<table>
<thead>
<tr>
<th>Server Type</th>
<th>Service Prefix</th>
<th>Default Service ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATROL Console Server</td>
<td>PATROL_CSERVER</td>
<td>The default service ID is the host name of the computer.</td>
</tr>
<tr>
<td>PATROL Agent</td>
<td>PATROL_AGENT</td>
<td>The default service ID is the host name of the computer, followed by an underscore, followed by the PATROL Agent’s port number.</td>
</tr>
</tbody>
</table>

How Service IDs Are Used

In addition to using service IDs to identify PATROL servers to the RTserver cloud, PATROL servers can also use their service IDs for additional purposes.

For example, the PATROL Console Server uses its service ID to determine which configuration files to use. PATROL Central Operator uses the service IDs of PATROL Console Servers and PATROL Agents to identify them to users.

Choosing a Service ID

You might choose to use a user-specified service ID for a PATROL server for many reasons, such as:

- to make it easier to identify the purpose of the server
- to make failover of that component easier

For example, you might choose one of the following as the service ID:

- a simple numerical value
- the virtual host ID or virtual IP address
- the service ID of an existing PATROL server that you are replacing
- a logical name that is associated with the role of the PATROL component in your enterprise
Service IDs can contain any of the following:

- upper case letters ('A' through 'Z')
- numbers ('0' through '9')
- dash ('-')
- underscore ('_')
- period ('.')

Service IDs cannot contain any other special characters. Any lower case characters ('a' through 'z') are automatically converted to upper case characters.

You can change the service ID of a PATROL server when starting it by using the -id command line argument. See the documentation for your PATROL server for more information.
Troubleshooting PATROL Console Server and RTserver

This chapter contains information for troubleshooting the PATROL Console Server and RTserver. This chapter contains the following topics:

PATROL Console Server Problems ................................................................. 272
  PATROL Console Server Fails to Start on Windows ................................. 273
  PATROL Console Server Fails to Start on Unix ....................................... 274
  PATROL Console Server on an NFS File System Fails to Start on Unix ...... 274
  On Unix, PATROL Console Server Stops After You Log Out .................. 275
  You Get the Error Message UNINITIALIZED_CONNECTION ..................... 275
  You Are Unable to Log Onto the PATROL Console Server with a Domain Account ................................................................. 276
  Connections to PATROL Agents Version 3.5.00 Fail ............................... 278
  Messages About Authentication Problems in the Log File .......................... 278
  PATROL Console Server Crashes Under a Heavy Processing Load on Windows 2000 ................................................................. 279
  Cannot Find Console Server DLLs Using the dllVersion Utility ............... 280
  Cannot Locate Files for KM Help or Icons ............................................ 281
  PATROL Central Operators or Management Profiles Disconnect ................ 281
  Secure Channel Error Messages ............................................................... 282
  Size of Management Profiles are Too Large .......................................... 282
RTserver Problems ...................................................................................... 283
  Error Messages Referring to “subject name ABCDEFG is not unique” ....... 283
  RTclient Cannot Connect to an RTserver .................................................. 284
  Auto-location of RTservers Does Not Work ............................................. 286
  The SmartSockets RTserver Service May Not Restart on Windows .......... 286
  All Object Query Fails to Complete Correctly ........................................ 287
Gathering Diagnostic Information ................................................................. 288
  Where to Find Diagnostic Information .................................................... 288
  Installation Log ......................................................................................... 289
  PATROL Console Server Logs ................................................................. 290
  RTserver Logs ......................................................................................... 290
  Generating Debug Information ................................................................. 291
  Determining the PATROL Console Server Version ................................ 291
PATROL Console Server Problems

This section contains troubleshooting information for the following problems:

<table>
<thead>
<tr>
<th>Problem Type</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATROL Console Server Fails to Start on Windows</td>
<td>page 273</td>
</tr>
<tr>
<td>PATROL Console Server Fails to Start on Unix</td>
<td>page 274</td>
</tr>
<tr>
<td>PATROL Console Server on an NFS File System Fails to Start on Unix</td>
<td>page 274</td>
</tr>
<tr>
<td>On Unix, PATROL Console Server Stops After You Log Out</td>
<td>page 275</td>
</tr>
<tr>
<td>You Get the Error Message UNINITIALIZED_CONNECTION</td>
<td>page 275</td>
</tr>
<tr>
<td>You Are Unable to Log Onto the PATROL Console Server with a Domain Account</td>
<td>page 276</td>
</tr>
<tr>
<td>Connections to PATROL Agents Version 3.5.00 Fail</td>
<td>page 278</td>
</tr>
<tr>
<td>Messages About Authentication Problems in the Log File</td>
<td>page 278</td>
</tr>
<tr>
<td>PATROL Console Server Crashes Under a Heavy Processing Load on Windows 2000</td>
<td>page 279</td>
</tr>
<tr>
<td>Cannot Find Console Server DLLs Using the d11Version Utility</td>
<td>page 280</td>
</tr>
<tr>
<td>Cannot Locate Files for KM Help or Icons</td>
<td>page 281</td>
</tr>
<tr>
<td>PATROL Central Operators or Management Profiles Disconnect</td>
<td>page 281</td>
</tr>
<tr>
<td>Secure Channel Error Messages</td>
<td>page 282</td>
</tr>
<tr>
<td>Size of Management Profiles are Too Large</td>
<td>page 282</td>
</tr>
</tbody>
</table>
PATROL Console Server Fails to Start on Windows

On Windows, the PATROL Console Server fails to start either after the initial installation or after you change the user name or password of the PATROL Console Server account.

If the PATROL Console Server fails to start when you attempt to start it from the services applet, it returns error 1069 or service specific error 100.

<table>
<thead>
<tr>
<th>Possible Explanation</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATROL Console Server account does not meet all of the requirements.</td>
<td>Ensure that the PATROL Console Server account meets all of the requirements listed under “PATROL Console Server Account” on page 33.</td>
</tr>
<tr>
<td>PATROL Console Server account does not exist or is not valid.</td>
<td>Update the PATROL Console Server account and password.</td>
</tr>
<tr>
<td>The user name or password for the Console Server account was changed in the operating system, but that change was not reflected in the patrol.conf file.</td>
<td>For instructions, see “Manually Changing the PATROL Console Server Account or Password on Windows” on page 78.</td>
</tr>
<tr>
<td>The user name or password for the Console Server account was changed in the operating system, but that change was not reflected in the PATROL Console Server service.</td>
<td></td>
</tr>
</tbody>
</table>
**PATROL Console Server Fails to Start on Unix**

On Unix, the PATROL Console Server fails to start either after the initial installation or after you change the user name or password of the PATROL Console Server account.

<table>
<thead>
<tr>
<th>Possible Explanation</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATROL Console Server account does not meet all of the requirements.</td>
<td>Ensure that the PATROL Console Server account meets all of the requirements listed under “PATROL Console Server Account” on page 33.</td>
</tr>
<tr>
<td>You changed the PATROL Console Server account and it no longer has the necessary file permissions.</td>
<td>Change the file permissions so that the new PATROL Console Server account is the owner of the directories that are opened for write access by the PATROL Console Server.</td>
</tr>
<tr>
<td>PATROL Console Server account does not exist or is not valid.</td>
<td>Update the PATROL Console Server account and password.</td>
</tr>
<tr>
<td>The user name or password for the Console Server account was changed in the operating system, but that change was not reflected in the patrol.conf file.</td>
<td>For instructions, see “Manually Changing the PATROL Console Server Account or Password on Unix” on page 79.</td>
</tr>
</tbody>
</table>

**PATROL Console Server on an NFS File System Fails to Start on Unix**

A PATROL Console Server that is installed on an NFS file system fails to start. The PATROL Console Server log file contains the following line:

`Another agent (ID: PATROL_CSERVER_<serviceld>) is running. Please choose another ID.`
On Unix, PATROL Console Server Stops After You Log Out

On Unix, you start the PATROL Console Server without the -detach command line argument. Then, the PATROL Console Server stops when you log out. The PATROL Console Server log file contains the following message:

```
Received hangup signal, stopping ...
```

### Possible Explanation
When you log out, the PATROL Console Server receives the HANGUP signal from the operating system and exits, because it was not started in detached mode.

### Solution
Do not start the PATROL Console Server with the -nodetach command line option.

See “Starting and Stopping the PATROL Console Server on Unix” on page 65.

You Get the Error Message UNINITIALIZED_CONNECTION

### Possible Explanation
Two PATROL components at different security levels are trying to communicate with each other.

The security level is set to 3 or 4 but certificates are missing.

### Solution
Configure all communicating servers at the same level of security.

Ensure the necessary certificates are in place.

For more information, see the PATROL Security User Guide.
You Are Unable to Log Onto the PATROL Console Server with a Domain Account

When you run PATROL Central Operator or PATROL Central Administration, you can log onto the PATROL Console Server with an account that is local to the PATROL Console Server, but not with a domain account.

When you attempt to log on with a domain account, you get the error message “Failed to log onto Console Server... authentication rejected.”

In an Active Directory environment, the PATROL Console Server log file might also contain the error message “OSS_Account::GetUserGroups() failed with error 5: Access is denied.”

<table>
<thead>
<tr>
<th>Possible Explanation</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>The PATROL Console Server account is a local account, so it does not have domain access to authenticate the domain account.</td>
<td>Change the PATROL Console Server account to a domain account. For instructions, see “Changing the PATROL Console Server Account or Password” on page 76.</td>
</tr>
<tr>
<td>&lt;Windows only&gt; In an Active Directory environment, the PATROL Console Server account does not have permission to read information about the domain account.</td>
<td>Ensure that each domain account has the “Read Group Membership” permission for the PATROL Console Server account (or the domain group that contains the PATROL Console Server account). For instructions, see “To Allow ‘Read Group Membership’ for a Single Active Directory Domain Account” on page 276 or “To Allow ‘Read Group Membership’ for Multiple Active Directory Domain Accounts” on page 277.</td>
</tr>
</tbody>
</table>

To Allow ‘Read Group Membership’ for a Single Active Directory Domain Account


2 Under the Users folder, right-click the domain user that failed to log onto the PATROL Console Server. Select Properties and select the Security tab.
3 In the **Name** field, ensure that the PATROL Console Server account is listed or the domain group that contains the PATROL Console Server account is listed.

If it is not listed, add either the PATROL Console Server account or its domain group.

4 In the **Permissions** field, grant **Allow** permissions to **Read Group Membership**.

**To Allow ‘Read Group Membership’ for Multiple Active Directory Domain Accounts**

1 On the Windows 2000 Domain Controller, open **Active Directory Users and Computers**.

2 Right-click the Users folder and choose **New=>Group**. Provide the group name and select **Global** under Group scope. Select **Security** for Group type. Click **OK**.

3 Add the PATROL Console Server account into the new Domain Global group that you just created.

4 Create either an Organizational Unit (OU), or a Domain Global group under the Users folder.

5 Move all domain users that are required to log onto the PATROL Console Server to the OU or Domain Global group that you just created.

6 Right-click the OU or Domain Global group and choose **Properties**, then select the **Security** tab.

7 Click the **Advanced** button, then click the **Permissions** tab.

8 Click the **Add** button and select the Domain Global group created in **step 2** and click **OK**.

9 Click the **Object** tab and check Allow on **Read All Properties**. Ensure that **This object and all child objects** is selected for **Apply onto**.

10 Click **OK** three times to accept the changes.
Connections to PATROL Agents Version 3.5.00 Fail

When you attempt to connect to a PATROL Agent 3.5.00 (for example, via PATROL Central Operator), the connection fails.

This problem occurs only with PATROL Agent 3.5.00.

<table>
<thead>
<tr>
<th>Possible Explanation</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>If a PATROL Agent version 3.5.00 has Operator Access Control Lists (ACLs) defined,</td>
<td>Upgrade the PATROL Agent to version 3.5.01 or later.</td>
</tr>
<tr>
<td>the ACLs cause the connections from the PATROL Console Server to the PATROL Agent</td>
<td>If upgrading the PATROL Agent is not possible, remove any Operator Access</td>
</tr>
<tr>
<td>to fail under certain conditions.</td>
<td>Control Lists (ACLs) defined for that PATROL Agent. For instructions, see</td>
</tr>
<tr>
<td></td>
<td>the PATROL Agent Reference Manual.</td>
</tr>
</tbody>
</table>

Messages About Authentication Problems in the Log File

The PATROL Console Server log file contains the following lines:

```
INFORM: <date> <time>:::Ksc found an authentication problem getting attributes from service 'PATROL_AGENT_<serviceId>'. Please check the impersonation entries for account '<cserverDefaultAccount>' on 'PATROL_CSERVER_<serviceId>'.
```

The Add Managed System Wizard in PATROL Central Operator also does not display platform or version information for some PATROL Agents.
**PATROL Console Server Crashes Under a Heavy Processing Load on Windows 2000**

Under a heavy processing load, a Windows 2000 server running Service Pack 2 could corrupt the heap stack and crash the program.

<table>
<thead>
<tr>
<th>Possible Explanation</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>You do not have the Windows 2000 Service Pack 3 or Service Pack 2 with the Q305227 hot patch applied, which the PATROL Console Server requires.</td>
<td>Install the Windows 2000 Service Pack 3 or Service Pack 2 with the Q305227 hot patch applied. For information on obtaining the hot patch without upgrading to Service Pack 3, see <a href="http://support.microsoft.com">http://support.microsoft.com</a></td>
</tr>
</tbody>
</table>
## Cannot Find Console Server DLLs Using the dllVersion Utility

<table>
<thead>
<tr>
<th>Possible Explanation</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unless you encountered a problem during installation, the dllVersion utility should return the DLLs for the product.</td>
<td>If dllVersion does not return the DLL file names, use the following methods.</td>
</tr>
<tr>
<td></td>
<td>On <strong>Windows</strong>, use the following procedure:</td>
</tr>
<tr>
<td></td>
<td>1. Find the specific .exe or .dll file.</td>
</tr>
<tr>
<td></td>
<td>2. Right-click on the file and select Properties from the pop-up menu.</td>
</tr>
<tr>
<td></td>
<td>3. On the File Properties dialog, select the Version tab. (If there is not a Version tab, open a new failure against the component associated with the DLL.)</td>
</tr>
<tr>
<td></td>
<td>The Version tab provides the following information:</td>
</tr>
<tr>
<td></td>
<td>• in the File Version field, product version</td>
</tr>
<tr>
<td></td>
<td>• in the Product Version field, product version and build string</td>
</tr>
<tr>
<td></td>
<td>On <strong>UNIX</strong> (except Linux), use the <em>what</em> utility.</td>
</tr>
<tr>
<td></td>
<td>This utility is typically installed in <code>/bin/what</code> or <code>/usr/ccs/bin</code>.</td>
</tr>
<tr>
<td></td>
<td>`% what liboss_t.so</td>
</tr>
<tr>
<td></td>
<td>If the <em>what</em> utility is not available, use the <em>strings</em> utility as follows:</td>
</tr>
<tr>
<td></td>
<td>`% strings liboss_t.so</td>
</tr>
<tr>
<td></td>
<td>On <strong>Linux</strong>, use the <em>ident</em> utility.</td>
</tr>
<tr>
<td></td>
<td>This utility is typically installed in <code>/usr/bin</code>.</td>
</tr>
<tr>
<td></td>
<td>`% ident liboss_t.so</td>
</tr>
<tr>
<td></td>
<td>However, it may not be available on all machines since the utility is often only included with development tools. If <em>ident</em> is not available, use the <em>strings</em> utility as follows:</td>
</tr>
<tr>
<td></td>
<td>`% strings liboss_t.so</td>
</tr>
</tbody>
</table>
## Cannot Locate Files for KM Help or Icons

<table>
<thead>
<tr>
<th>Possible Explanation</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATROL Central Operator – Microsoft Windows Edition architecture requires Knowledge Module (KM) console resources to be installed on the console server machine. The console server supports a product- and version-specific directory hierarchy so that different versions of the same KM on different agents can be supported in the same management profile. Thus, each Console Server uses its version-specific help and icon files.</td>
<td>All KMs are packaged so that the Product Selection screen (in the Common Install process) does not require you to select the type of files. Select the KMs you want to use on the Console Server and the installer retrieves the appropriate icon and help files (no .km or .psl files will be installed unless an agent is also present on the machine). <strong>Note:</strong> Since the Console Server does not read .km and .psl files, you do not need to deploy customized .km and .psl files to the console server machine. Customized .km and .psl files only need to be deployed to the agent machines. For response dialog help to appear, use PATROL Central Operator – Microsoft Windows Edition version 7.2.00 or later.</td>
</tr>
</tbody>
</table>

## PATROL Central Operators or Management Profiles Disconnect

<table>
<thead>
<tr>
<th>Possible Explanation</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>PATROL Central Operators or management profiles may disconnect from the PATROL Console Server if there are too many concurrent queries in a multiple cloud setup.</td>
<td>Limit the number of concurrent queries to seven. In other words, do not issue queries on more than seven different PATROL Central Operators at the same time. The average object count per PATROL Central Operator should not be more than 260,000 objects.</td>
</tr>
</tbody>
</table>
Secure Channel Error Messages

The PATROL Console Server log file contains the following error message:

secure channel not allowed

<table>
<thead>
<tr>
<th>Possible Explanation</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>The correct security level is not set in the site policy, or it has been overwritten in the application policy</td>
<td>Set the correct security level in the site policy.</td>
</tr>
<tr>
<td></td>
<td>For more information, see the PATROL Security User Guide.</td>
</tr>
<tr>
<td>The certificates are expired</td>
<td>Generate new certificates and install them on both the PATROL Console Server and any managed nodes associated with these error messages.</td>
</tr>
<tr>
<td></td>
<td>For more information, see the PATROL Security User Guide.</td>
</tr>
<tr>
<td>The certificates are not installed correctly.</td>
<td>Verify that the certificates are installed correctly by using the sslcmd function to examine the various key databases. These databases are files with the .kdb extension.</td>
</tr>
<tr>
<td></td>
<td>For more information, see the PATROL Security User Guide.</td>
</tr>
</tbody>
</table>

Size of Management Profiles are Too Large

<table>
<thead>
<tr>
<th>Possible Explanation</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>If you have a KM in your management profile that creates a lot of instances that are not reused after they are destroyed, your management profile can continually grow and become too large.</td>
<td>Define the LT_TRANSIENT_KM_APPS environment variable in the patrol.conf file to be a comma-delimited list of KMs whose instances are removed from the management profile as soon as they are destroyed on the PATROL Agent.</td>
</tr>
<tr>
<td></td>
<td>For example, add the following to the patrol.conf file:</td>
</tr>
<tr>
<td></td>
<td>set environment</td>
</tr>
<tr>
<td></td>
<td>LT_TRANSIENT_KM_APPS=WTS_USER, WTS_SESSION</td>
</tr>
<tr>
<td></td>
<td>end</td>
</tr>
</tbody>
</table>
RTserver Problems

This section contains troubleshooting information for the following problems:

<table>
<thead>
<tr>
<th>Problem Type</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Error Messages Referring to “subject name ABCDEFG is not unique”</td>
<td>page 283</td>
</tr>
<tr>
<td>RTclient Cannot Connect to an RTserver</td>
<td>page 284</td>
</tr>
<tr>
<td>Auto-location of RTservers Does Not Work</td>
<td>page 286</td>
</tr>
<tr>
<td>The SmartSockets RTserver Service May Not Restart on Windows</td>
<td>page 286</td>
</tr>
<tr>
<td>All Object Query Fails to Complete Correctly</td>
<td>page 287</td>
</tr>
</tbody>
</table>

Error Messages Referring to “subject name ABCDEFG is not unique”

The RTserver log file contains error messages referring to “subject name ABCDEFG is not unique.”

<table>
<thead>
<tr>
<th>Possible Explanation</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>You use _any in the conn_names option in the rtserver.cm file.</td>
<td>Change the conn_names option so that it no longer uses _any. For more information about the conn_names option, see “Configuring RTservers to Listen for Connections” on page 236. For information about using _any, see “Configuring RTserver to Listen for Connections on Computers with Multiple IP Addresses” on page 238.</td>
</tr>
<tr>
<td>Using _any with an RTserver that is part of a cloud causes extra traffic and connection attempts between RTservers in the cloud and produces the above error message.</td>
<td></td>
</tr>
</tbody>
</table>
An RTclient, such as PATROL Console Server, PATROL Agent, or PATROL Central Operator cannot connect to an RTserver.

<table>
<thead>
<tr>
<th>Possible Explanation</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTserver is not running.</td>
<td>Start the RTserver.</td>
</tr>
<tr>
<td>RTclient’s list of RTservers contains the incorrect host name or port number for the RTserver.</td>
<td>Check the RTclient’s list of RTservers to ensure that it matches the host name and port number for the RTserver, as listed in the conn_names option for the RTserver. For example, if the RTclient uses tcp:localhost:2059; the RTserver must have the exact same string in its conn_names option.</td>
</tr>
<tr>
<td>RTclient does not use the RTSERVERS environment variable for its list of RTservers.</td>
<td>Check how the RTclient obtains its list of RTservers. For example, it might use a file instead of the RTSERVERS environment variable, or a command line option might be overriding the environment variable.</td>
</tr>
<tr>
<td>RTclient’s list of RTservers contains white space, such as spaces or tabs, which prevent the variable from being read correctly.</td>
<td>Update the RTclient’s list of RTservers so that it does not contain any white space, such as spaces or tabs. For instructions on how to update the list of RTservers for a specific component, see the documentation for that component.</td>
</tr>
<tr>
<td>RTclient’s list of RTservers uses the $auto:*:port logical connection name; however, the RTserver does not contain the corresponding udp_broadcast:_node:port logical connection name.</td>
<td>Add the corresponding udp_broadcast:_node:port logical connection name to the conn_names option for the RTserver and restart the RTserver.</td>
</tr>
<tr>
<td>Possible Explanation</td>
<td>Solution</td>
</tr>
<tr>
<td>------------------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>The RTclient was disconnected and the RTserver has not yet detected that the</td>
<td>The RTserver will detect that the RTclient has been disconnected when the client_read_timeout and client_keep_alive_timeout options in the rtserver.cm file are reached. However, it may take too long if the default values are used.</td>
</tr>
<tr>
<td>connection has been dropped.</td>
<td>Change the values of the client_read_timeout and client_keep_alive_timeout options in the rtserver.cm file to a shorter time period. For example, set the values to 300 (5 minutes) instead of the default of 1200 (20 minutes).</td>
</tr>
<tr>
<td>You may get the following error message when trying to reconnect:</td>
<td>For instructions about how to edit the rtserver.cm file, see “Editing an rtserver.cm File” on page 233.</td>
</tr>
<tr>
<td>Service name already in use</td>
<td></td>
</tr>
<tr>
<td>The number of clients already connected to the RTserver has reached the limit</td>
<td>If the RTserver is configured to support fewer RTserver clients than what is recommended for its hardware, then increase the value of the max_client_conns option to accommodate additional RTserver clients (up to a limit of 750).</td>
</tr>
<tr>
<td>specified by the max_client_conns option in the rtserver.cm file.</td>
<td>If the RTserver is already supporting the maximum number of clients recommended for its hardware, then the PATROL application should be directed to another RTserver. Based on typical best-practice recommendations, this RTserver is likely to be in another RTserver cloud. If there are no available RTserver clouds with capacity for additional RTserver clients, then you may have to create another RTserver cloud.</td>
</tr>
<tr>
<td></td>
<td>For more information about the RTserver and the RTserver cloud capacity and hardware recommendations, see the PATROL Central Infrastructure Best Practices Guide.</td>
</tr>
</tbody>
</table>
Auto-location of RTservers Does Not Work

An RTclient cannot locate an RTserver by using auto-location.

<table>
<thead>
<tr>
<th>Possible Explanation</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTclient’s list of RTservers does not include the auto:*:port logical connection name.</td>
<td>Add the auto:*:port logical connection name to the RTclient’s list of RTservers.</td>
</tr>
<tr>
<td>The conn_names option for the RTserver does not contain the udp_broadcast:_node:port logical connection name.</td>
<td>Add the udp_broadcast:_node:port logical connection name to the conn_names option for the RTserver.</td>
</tr>
<tr>
<td>RTserver and RTclient are on different subnets.</td>
<td>No action is required. Auto-location does not work across subnets.</td>
</tr>
</tbody>
</table>

The SmartSockets RTserver Service May Not Restart on Windows

Using the Restart option in the Services MMC snap-in window to restart the SmartSockets RTserver service may generate the following pop-up error message:

Could not start the SmartSockets RTserver service on Local Computer

The service did not return an error. This could be an internal Windows error or an internal service error.

If the problem persists, contact your system administrator.

However, in some cases, the Restart option successfully restarts the SmartSockets RTserver service.

<table>
<thead>
<tr>
<th>Possible Explanation</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>The RTserver service may be trying to restart before the WIN32 socket layer has closed.</td>
<td>To restart the SmartSockets RTserver service, click Start in the Services window.</td>
</tr>
</tbody>
</table>
All Object Query Fails to Complete Correctly

All Object queries may fail to complete correctly when operators are running PATROL Agent queries across hundreds of PATROL Agents.

<table>
<thead>
<tr>
<th>Possible Explanation</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>The RTserver does not have sufficient buffer capacity to handle peak data volumes generated by All Object query operations across hundreds of PATROL Agents with many managed objects. When this buffer is reached, a part of the query results is lost and the query may fail to complete successfully.</td>
<td>Increase the value of the client_max_buffer RTserver configuration option. This variable specifies the maximum number of bytes of data that an RTserver process can buffer. This variable accepts an integer value whose default is 200000000 (200 MB). Doubling the value to 400 MB is usually sufficient to handle the peak load conditions that may be causing the query to fail.</td>
</tr>
</tbody>
</table>
Gathering Diagnostic Information

This section contains general information on gathering diagnostic information for PATROL Console Server and RTserver.

Where to Find Diagnostic Information

Table 33 lists locations where you can find diagnostic information for problems with the PATROL Console Server and RTserver on Windows.

Table 33  Locations for Diagnostic Information on Windows

<table>
<thead>
<tr>
<th>Type</th>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation logs</td>
<td>varies with operating system</td>
<td>See “Installation Log” on page 289.</td>
</tr>
<tr>
<td>PATROL Console Server log</td>
<td>%PATROL_ROOT%\log\cserver\</td>
<td>See “PATROL Console Server Logs” on page 290.</td>
</tr>
<tr>
<td>RTserver log</td>
<td>%BMC_ROOT%\common\smartsockets\log</td>
<td>See “Specifying RTserver Logging Options” on page 246.</td>
</tr>
<tr>
<td>Security logs</td>
<td>%BMC_ROOT%\common\security\log_V3.0\</td>
<td>log files related to PATROL Console Server are cserver*.log</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For more information, see the PATROL Security User Guide.</td>
</tr>
</tbody>
</table>
Table 34 lists locations where you can find diagnostic information for problems with the PATROL Console Server and RTserver on Unix.

<table>
<thead>
<tr>
<th>Type</th>
<th>Location</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation logs</td>
<td>varies with operating system</td>
<td>See “Installation Log” on page 289.</td>
</tr>
<tr>
<td>PATROL Console Server log</td>
<td>$PATROL_ROOT/log/cserver/</td>
<td>See “PATROL Console Server Logs” on page 290.</td>
</tr>
<tr>
<td>RTserver log</td>
<td>$BMC_ROOT/common/smartsockets/log</td>
<td>See “Specifying RTserver Logging Options” on page 246.</td>
</tr>
<tr>
<td>Security logs</td>
<td>$BMC_ROOT/common/security/log_V3.0/</td>
<td>log files related to PATROL Console Server are cserver*.log</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For more information, see the PATROL Security User Guide.</td>
</tr>
</tbody>
</table>

### Installation Log

One log file is created each time the installer is run. The name of the log file is a combination of the computer name and a time stamp. The location of the file depends on the operating system:

- On Windows, the log file is located in the `%USERPROFILE%\Application Data\BMCinstall` directory.

  For example, the log file for user patrol on the Windows 2000 server PATROL_1 is located in the `C:\Documents and Settings\patrol\Application Data\BMCinstall` directory. The name of the log file is `PATROL_1-1005340189.log`.

- On Unix, the log file is saved to the `home_directory/BMCINSTALL/` directory.
PATROL Console Server Logs

The PATROL Console Server maintains its log files in the following directory:

- $PATROL_ROOT/log/cserver/ (Unix)
- %PATROL_ROOT%/log/cserver/ (Windows)

The log file name for the currently running instance of PATROL Console Server is cserver-serviceID.log, where serviceID is the service ID of the PATROL Console Server.

The PATROL Console Server maintains log files for the last five times that it was started. Log files for previous times that the PATROL Console Server was started have a number appended to the log file name. The larger the number, the older the log file.

RTserver Logs

The RTserver maintains its log files in the following directories:

- %BMC_ROOT%/common/smartsockets/log (Windows)
- $BMC_ROOT/common/smartsockets/log (Unix)

The log file name for the default instance of RTserver is rtserver.log. If more than one instance of RTserver is running on the computer, the log file name is rtserver.idName.log, where idName is the instance identifier for each RTserver.

On Unix, the start_rtserver.log file is created to capture unexpected startup problems. This file is also located in the $BMC_ROOT/common/smartsockets/log directory. However, all other information is written to the rtserver.log file.

For more information about the RTserver log file, see “Specifying RTserver Logging Options” on page 246. For more information about running more than one RTserver concurrently on the same computer, see “Running Multiple RTservers on the Same Computer” on page 252.
Generating Debug Information

This section includes tasks for generating debug and other diagnostic information.

<table>
<thead>
<tr>
<th>Task</th>
<th>Reference</th>
</tr>
</thead>
<tbody>
<tr>
<td>Determining the PATROL Console Server Version</td>
<td>page 291</td>
</tr>
<tr>
<td>Determining the RTserver Version</td>
<td>page 292</td>
</tr>
</tbody>
</table>

Determining the PATROL Console Server Version

This task describes how to determine the PATROL Console Server version.

To Determine the Version of the PATROL Console Server on Windows

1. Change to the `%PATROL_ROOT%\bin\target` directory.
2. Enter the following command:
   
   `cserver.exe -v`

To Determine the Version of the PATROL Console Server on Unix

1. Change to the `$PATROL_ROOT` directory.
2. Enter the following command:
   
   `./start_cserver.sh -v`
Determining the RTserver Version

This task describes how to determine the RTserver version.

To Determine the Version of the RTserver on Windows

Look at the first line of the `%RTHOME%\log\rtserver.log` file.

To Determine the Version of the RTserver on Unix

Look at the first line of the `$RTHOME/log/rtserver.log` file.

**WARNING**

Although you can also obtain the version number by using the -version command line option, this is not recommended, because it will overwrite any information in the existing log file.
Environment Variables, Directories, and Files

This appendix lists the environment variables, directories and files for the PATROL Console Server and RTserver.

This appendix contains the following topics:

PATROL Console Server Environment Variables, Directories, and Files 294
  Environment Variables Used by PATROL Console Server 294
  PATROL Console Server Directories and Files 296
RTserver Environment Variables, Directories, and Files 307
  Environment Variables Used by the RTserver 307
  RTserver Directories and Files 308
PATROL Console Server Environment Variables, Directories, and Files

This section lists the environment variables, directories, and files that are specific to the PATROL Console Server.

Environment Variables Used by PATROL Console Server

Table 35 lists the environment variable names and their default values as used by the PATROL Console Server on Windows.

Table 35  PATROL Console Server Environment Variables on Windows

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC_COMMON</td>
<td>%BMC_ROOT%\common</td>
</tr>
<tr>
<td>BMC_COMMON_BMC</td>
<td>%BMC_ROOT%\common\bmc</td>
</tr>
<tr>
<td>BMC_COMMON_BMC_BIN</td>
<td>%BMC_ROOT%\common\bmc\bin\target</td>
</tr>
<tr>
<td>BMC_COMMON_BMC_LIB</td>
<td>%BMC_ROOT%\common\bmc\lib</td>
</tr>
<tr>
<td>BMC_COMMON_BMC_MOF</td>
<td>%BMC_ROOT%\common\bmc\lib\mof</td>
</tr>
<tr>
<td>BMC_ROOT</td>
<td>specified at installation—see “Installation Directory” on page 33</td>
</tr>
<tr>
<td>PATROL_LANG</td>
<td>specifies the locale name to be used—works as an override to the standard operating system variables for locale names</td>
</tr>
<tr>
<td>PATROL_ROOT</td>
<td>%BMC_ROOT%\Patrol7</td>
</tr>
<tr>
<td>PATROL_ROOT_BIN</td>
<td>%PATROL_ROOT%\bin\target</td>
</tr>
<tr>
<td>PATROL_ROOT_CONFIG</td>
<td>%PATROL_ROOT%\config</td>
</tr>
<tr>
<td>PATROL_ROOT_ETC</td>
<td>%PATROL_ROOT%\etc</td>
</tr>
<tr>
<td>PATROL_ROOT_GLOBAL_LIB</td>
<td>%PATROL_ROOT%\lib</td>
</tr>
<tr>
<td>PATROL_ROOT_KM</td>
<td>%PATROL_ROOT%\lib\knowledge</td>
</tr>
<tr>
<td>PATROL_ROOT_LAYOUT</td>
<td>%PATROL_ROOT%\log\cserver\layout</td>
</tr>
<tr>
<td>PATROL_ROOT_LOG</td>
<td>%PATROL_ROOT%\log</td>
</tr>
<tr>
<td>PATROL_ROOT_MOF</td>
<td>%PATROL_ROOT%\lib\mof</td>
</tr>
<tr>
<td>PATROL_ROOT_NLS</td>
<td>%PATROL_ROOT%\lib\nls</td>
</tr>
<tr>
<td>PATROL_ROOT_PSL_APPS</td>
<td>%PATROL_ROOT%\lib\psl</td>
</tr>
<tr>
<td>PATROL_ROOT_TMP</td>
<td>%PATROL_ROOT%\tmp</td>
</tr>
<tr>
<td>RTSERVERS</td>
<td>specified at installation—see “RTSERVERS Environment Variable” on page 35</td>
</tr>
</tbody>
</table>
Table 36 lists the environment variable names and their default values as used by the PATROL Console Server on Unix.

Table 36  PATROL Console Server Environment Variables on Unix

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC_COMMON</td>
<td>$BMC_ROOT/common</td>
</tr>
<tr>
<td>BMC_COMMON_BMC</td>
<td>$BMC_ROOT/common/bmc</td>
</tr>
<tr>
<td>BMC_COMMON_BMC_BIN</td>
<td>$BMC_ROOT/common/bmc/bin/target</td>
</tr>
<tr>
<td>BMC_COMMON_BMC_LIB</td>
<td>$BMC_ROOT/common/bmc/lib</td>
</tr>
<tr>
<td>BMC_COMMON_BMC_MOF</td>
<td>$BMC_ROOT/common/bmc/lib/mof</td>
</tr>
<tr>
<td>BMC_ROOT</td>
<td>specified at installation—see “Installation Directory” on page 33</td>
</tr>
<tr>
<td>PATROL_LANG</td>
<td>specifies the locale name to be used—works as an override to the standard operating system variables for locale names</td>
</tr>
<tr>
<td>PATROL_ROOT</td>
<td>$BMC_ROOT/Patrol7</td>
</tr>
<tr>
<td>PATROL_ROOT_BIN</td>
<td>$PATROL_ROOT/bin/target</td>
</tr>
<tr>
<td>PATROL_ROOT_CONFIG</td>
<td>$PATROL_ROOT/config</td>
</tr>
<tr>
<td>PATROL_ROOT_ETC</td>
<td>$PATROL_ROOT/etc</td>
</tr>
<tr>
<td>PATROL_ROOT_GLOBAL_LIB</td>
<td>$PATROL_ROOT/lib</td>
</tr>
<tr>
<td>PATROL_ROOT_KM</td>
<td>$PATROL_ROOT/lib/knowledge</td>
</tr>
<tr>
<td>PATROL_ROOT_LAYOUT</td>
<td>$PATROL_ROOT/log/cserver/layout</td>
</tr>
<tr>
<td>PATROL_ROOT_LOG</td>
<td>$PATROL_ROOT/log</td>
</tr>
<tr>
<td>PATROL_ROOT_MOF</td>
<td>$PATROL_ROOT/lib/mof</td>
</tr>
<tr>
<td>PATROL_ROOT_NLS</td>
<td>$PATROL_ROOT/lib/nls</td>
</tr>
<tr>
<td>PATROL_ROOT_PSL_APPS</td>
<td>$PATROL_ROOT/lib/psl</td>
</tr>
<tr>
<td>PATROL_ROOT_TMP</td>
<td>$PATROL_ROOT/tmp</td>
</tr>
<tr>
<td>RTSERVERS</td>
<td>specified at installation—see “RTSERVERS Environment Variable” on page 35</td>
</tr>
</tbody>
</table>

For information about setting these environment variables, see “Setting Environment Variables for the PATROL Console Server” on page 139.
**PATROL Console Server Directories and Files**

This section lists directories that are specific to PATROL Console Server. It also lists selected files of particular importance for using or configuring PATROL Console Server. It does not list other directories or files for other PATROL components, such as the PATROL Agent or PATROL console themselves. Note that some directories and files are shared by multiple PATROL components.

Most of the directories and files are created during the installation of PATROL Console Server. If a directory or file is created after installation, when it is created is noted in the comments.

**NOTE**

For many of these directories, the PATROL Console Server allows the administrator to specify alternate directories by using of environment variables. For more information, see “Environment Variables Used by PATROL Console Server” on page 294.

Table 37 lists the directories and files for PATROL Console Server on Windows.

<table>
<thead>
<tr>
<th>Directories and Files</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>%BMC_ROOT%</td>
<td>the installation directory</td>
</tr>
<tr>
<td>%BMC_ROOT%\common\</td>
<td>the top-level directory for components shared by multiple PATROL7 products</td>
</tr>
<tr>
<td></td>
<td>These shared components include third-party products as well as shared BMC libraries.</td>
</tr>
<tr>
<td></td>
<td>This directory can be redefined by the BMC_COMMON environment variable.</td>
</tr>
<tr>
<td>%BMC_ROOT%\common\bmc| the top-level directory for BMC shared libraries</td>
<td></td>
</tr>
<tr>
<td></td>
<td>This directory can be redefined by the BMC_COMMON_BMC environment variable.</td>
</tr>
</tbody>
</table>
Table 37  Directories and Files for PATROL Console Server on Windows (Part 2 of 6)

<table>
<thead>
<tr>
<th>Directories and Files</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>%BMC_ROOT%/common\bmc\bin\target\</td>
<td>the directory for BMC shared libraries and executables</td>
</tr>
<tr>
<td></td>
<td>This directory can be redefined by the BMC_COMMON_BMC_BIN environment variable.</td>
</tr>
<tr>
<td>%BMC_ROOT%/common\bmc\lib\</td>
<td>the directory for configuration files used by BMC shared libraries; includes MOF (Managed Object Format) configuration files and message catalogs for different locales</td>
</tr>
<tr>
<td></td>
<td>This directory can be redefined by the BMC_COMMON_BMC_LIB environment variable.</td>
</tr>
<tr>
<td>%BMC_ROOT%/common\bmc\lib\mof\</td>
<td>the directory for MOF configuration files used by BMC shared libraries</td>
</tr>
<tr>
<td></td>
<td>This directory can be redefined by the BMC_COMMON_BMC_MO F environment variable.</td>
</tr>
<tr>
<td>%BMC_ROOT%/common\patrol.d</td>
<td>configuration file</td>
</tr>
<tr>
<td>patrol.conf</td>
<td></td>
</tr>
<tr>
<td>%BMC_ROOT%/Patrol7\</td>
<td>the top-level directory for PATROL Console Server</td>
</tr>
<tr>
<td></td>
<td>This directory can be redefined by the PATROL_ROOT environment variable.</td>
</tr>
<tr>
<td>%PATROL_ROOT%/bin\target\</td>
<td>the directory for PATROL Console Server shared libraries and executables</td>
</tr>
<tr>
<td></td>
<td>This directory can be redefined by the PATROL_ROOT_BIN environment variable.</td>
</tr>
</tbody>
</table>
### Table 37  Directories and Files for PATROL Console Server on Windows (Part 3 of 6)

<table>
<thead>
<tr>
<th>Directories and Files</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>%PATROL_ROOT%\config\</td>
<td>the directory used by PATROL Console Server to store configuration information, including:</td>
</tr>
<tr>
<td></td>
<td>■ all configurable options including advertised service name.</td>
</tr>
<tr>
<td></td>
<td>■ access control database</td>
</tr>
<tr>
<td></td>
<td>■ impersonation database</td>
</tr>
<tr>
<td></td>
<td>This directory can be redefined by the PATROL_ROOT_CONFIG environment variable.</td>
</tr>
<tr>
<td>%PATROL_ROOT%\config\cservice\</td>
<td>access control database that contains rights used by the console modules of PATROL 7.x consoles</td>
</tr>
<tr>
<td>ac_db_serviceID.mof</td>
<td>impersonation databases that hold alias and impersonation tables</td>
</tr>
<tr>
<td>PATROL_CSERVER_serviceID_impdb.mof</td>
<td>console server startup default configuration file</td>
</tr>
<tr>
<td>acfg_cserver_serviceID.mof</td>
<td></td>
</tr>
<tr>
<td>%PATROL_ROOT%\etc\</td>
<td>the directory used by PATROL Console Server for security sensitive files such as the Secure Key Store (SKS), an encrypted file of passwords used for impersonation</td>
</tr>
<tr>
<td></td>
<td>This directory can be redefined by the PATROL_ROOT_ETC environment variable.</td>
</tr>
<tr>
<td>PATROL_CSERVER_serviceID_sks.mk4</td>
<td>holds the Secure Key Store (SKS), a database of impersonation aliases, user IDs, and encrypted passwords</td>
</tr>
</tbody>
</table>

---

**Table37 Directories and Files for PATROL Console Server on Windows (Part 3 of 6)**

- **%PATROL_ROOT%\config\** - the directory used by PATROL Console Server to store configuration information, including:
  - all configurable options including advertised service name.
  - access control database
  - impersonation database
  This directory can be redefined by the PATROL_ROOT_CONFIG environment variable.

- **%PATROL_ROOT%\config\cservice\** - access control database that contains rights used by the console modules of PATROL 7.x consoles.

- **%PATROL_ROOT%\etc\** - the directory used by PATROL Console Server for security sensitive files such as the Secure Key Store (SKS), an encrypted file of passwords used for impersonation.
  This directory can be redefined by the PATROL_ROOT_ETC environment variable.

- **PATROL_CSERVER_serviceID_sks.mk4** - holds the Secure Key Store (SKS), a database of impersonation aliases, user IDs, and encrypted passwords.
### Table 37  Directories and Files for PATROL Console Server on Windows (Part 4 of 6)

<table>
<thead>
<tr>
<th>Directories and Files</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>%PATROL_ROOT%\lib\</td>
<td>the directory for configuration files used by PATROL Console Server. This directory can be redefined by the PATROL_ROOT_GLOBAL_LIB environment variable.</td>
</tr>
<tr>
<td>%PATROL_ROOT%\lib\knowledge\</td>
<td>the directory used by PATROL Console Server for KM data. This directory can be redefined by the PATROL_ROOT_KM environment variable.</td>
</tr>
<tr>
<td>%PATROL_ROOT%\lib\knowledge\default\lib\help\language\</td>
<td>contains Knowledge Module help files for display in PATROL 7.x consoles.</td>
</tr>
<tr>
<td>%PATROL_ROOT%\lib\mof\</td>
<td>the directory for MOF configuration files used by PATROL Console Server. This directory can be redefined by the PATROL_ROOT_MOF environment variable.</td>
</tr>
<tr>
<td>%PATROL_ROOT%\lib\mof\cserver\</td>
<td>default managed system queries for new management profiles. default event filters for new management profiles. default chart schemes for new management profiles.</td>
</tr>
<tr>
<td>%PATROL_ROOT%\lib\nls\</td>
<td>the directory for message catalogs used by PATROL Console Server. This directory can be redefined by the PATROL_ROOT_NLS environment variable.</td>
</tr>
</tbody>
</table>
Table 37  Directories and Files for PATROL Console Server on Windows (Part 5 of 6)

<table>
<thead>
<tr>
<th>Directories and Files</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>%PATROL_ROOT%\lib\psl\</td>
<td>the directory for PSL7 files</td>
</tr>
<tr>
<td></td>
<td>This directory can be redefined by the PATROL_ROOT_PSL_APPS environment variable.</td>
</tr>
<tr>
<td>%PATROL_ROOT%\log\</td>
<td>the directory for log files generated by the PATROL Console Server as well as subdirectories that contain persistent data for various components of the PATROL Console Server, including:</td>
</tr>
<tr>
<td></td>
<td>■ management profiles</td>
</tr>
<tr>
<td></td>
<td>■ service cache databases</td>
</tr>
<tr>
<td></td>
<td>■ managed system group definitions</td>
</tr>
<tr>
<td></td>
<td>This directory can be redefined by the PATROL_ROOT_LOG environment variable.</td>
</tr>
<tr>
<td>%PATROL_ROOT%\log\cserv\</td>
<td>contains console server log messages</td>
</tr>
<tr>
<td>cserver-serviceID.log</td>
<td></td>
</tr>
<tr>
<td>%PATROL_ROOT%\log\cserv\layout\</td>
<td>the directory used by PATROL Console Server to store management profile data</td>
</tr>
<tr>
<td></td>
<td>This directory can be redefined by the PATROL_ROOT_LAYOUT environment variable.</td>
</tr>
<tr>
<td>%PATROL_ROOT%\log\history\</td>
<td>history files for console server events</td>
</tr>
<tr>
<td>PATROL_CSERVER_serviceID_events.mk</td>
<td></td>
</tr>
<tr>
<td>%PATROL_ROOT%\log\layout\PATROL_CSERVER_serviceID\</td>
<td>contains management profiles</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 37  Directories and Files for PATROL Console Server on Windows (Part 6 of 6)

<table>
<thead>
<tr>
<th>Directories and Files</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>%PATROL_ROOT%\security\</td>
<td>security files</td>
</tr>
<tr>
<td>%PATROL_ROOT%\tmp\</td>
<td>the directory used by PATROL Console Server to create temporary files. This directory can be redefined by the PATROL_ROOT_TMP environment variable.</td>
</tr>
</tbody>
</table>

Table 38 lists the directories and files for PATROL Console Server on Unix.

Table 38  Directories and Files for PATROL Console Server on Unix (Part 1 of 6)

<table>
<thead>
<tr>
<th>Directories and Files</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>/etc/patrol.d/</td>
<td>configuration file</td>
</tr>
<tr>
<td>patrol.conf</td>
<td>the installation directory</td>
</tr>
<tr>
<td>$BMC_ROOT/</td>
<td>the top-level directory for components shared by multiple PATROL7 products. These shared components include third-party products as well as shared BMC libraries. This directory can be redefined by the BMC_COMMON environment variable.</td>
</tr>
<tr>
<td>$BMC_ROOT/common/</td>
<td>the top-level directory for BMC shared libraries. This directory can be redefined by the BMC_COMMON_BMC environment variable.</td>
</tr>
<tr>
<td>$BMC_ROOT/common/bmc/</td>
<td>the top-level directory for BMC shared libraries</td>
</tr>
</tbody>
</table>

Table 38 lists the directories and files for PATROL Console Server on Unix.
Table 38  Directories and Files for PATROL Console Server on Unix (Part 2 of 6)

<table>
<thead>
<tr>
<th>Directories and Files</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>$BMC_ROOT/common/bmc/bin/target/</td>
<td>the directory for BMC shared libraries and executables This directory can be redefined by the BMC_COMMON_BMC_BIN environment variable.</td>
</tr>
<tr>
<td>$BMC_ROOT/common/bmc/lib/</td>
<td>the directory for configuration files used by BMC shared libraries; includes MOF (Managed Object Format) configuration files and message catalogs for different locales This directory can be redefined by the BMC_COMMON_BMC_LIB environment variable.</td>
</tr>
<tr>
<td>$BMC_ROOT/common/bmc/lib/mof/</td>
<td>the directory for MOF configuration files used by BMC shared libraries This directory can be redefined by the BMC_COMMON_BMC_MO F environment variable.</td>
</tr>
<tr>
<td>$BMC_ROOT/Patrol7/</td>
<td>the top-level directory for PATROL Console Server This directory can be redefined by the PATROL_ROOT environment variable.</td>
</tr>
<tr>
<td>$PATROL_ROOT/</td>
<td>start_cserver.sh the script for starting the PATROL Console Server stop_cserver.sh the script for stopping the PATROL Console Server</td>
</tr>
</tbody>
</table>
### Table 38  Directories and Files for PATROL Console Server on Unix (Part 3 of 6)

<table>
<thead>
<tr>
<th>Directories and Files</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>$PATROL_ROOT/bin/target/</code></td>
<td>the directory for PATROL Console Server shared libraries and executables</td>
</tr>
<tr>
<td></td>
<td>This directory can be redefined by the <code>PATROL_ROOT_BIN</code> environment variable.</td>
</tr>
<tr>
<td><code>$PATROL_ROOT/config/</code></td>
<td>the directory used by PATROL Console Server to store configuration information, including:</td>
</tr>
<tr>
<td></td>
<td>- all configurable options including advertised service name.</td>
</tr>
<tr>
<td></td>
<td>- access control database</td>
</tr>
<tr>
<td></td>
<td>- impersonation database</td>
</tr>
<tr>
<td></td>
<td>This directory can be redefined by the <code>PATROL_ROOT_CONFIG</code> environment variable.</td>
</tr>
<tr>
<td><code>$PATROL_ROOT/config/cserver/</code></td>
<td></td>
</tr>
<tr>
<td><code>ac_db_serviceID.mof</code></td>
<td>access control database that contains rights used by the console modules of PATROL 7.x consoles</td>
</tr>
<tr>
<td><code>PATROL_CSERVER_serviceID_impdb.mof</code></td>
<td>impersonation databases that hold alias and impersonation tables</td>
</tr>
<tr>
<td><code>acfg_cserver_serviceID.mof</code></td>
<td>console server startup default configuration file</td>
</tr>
</tbody>
</table>
### Table 38  Directories and Files for PATROL Console Server on Unix (Part 4 of 6)

<table>
<thead>
<tr>
<th>Directories and Files</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>$PATROL_ROOT/etc/</td>
<td>the directory used by PATROL Console Server for security sensitive files such as the Secure Key Store (SKS), an encrypted file of passwords used for impersonation. This directory can be redefined by the PATROL_ROOT_ETC environment variable.</td>
</tr>
<tr>
<td>PATROL_CSERVER_serviceID_sks.mk4</td>
<td>holds the Secure Key Store (SKS), a database of impersonation aliases, user IDs, and encrypted passwords.</td>
</tr>
<tr>
<td>$PATROL_ROOT/lib/</td>
<td>the directory for configuration files used by PATROL Console Server. This directory can be redefined by the PATROL_ROOT_GLOBAL_LIB environment variable.</td>
</tr>
<tr>
<td>$PATROL_ROOT/lib/knowledge/</td>
<td>the directory used by PATROL Console Server for KM data. This directory can be redefined by the PATROL_ROOT_KM environment variable.</td>
</tr>
<tr>
<td>$PATROL_ROOT/lib/knowledge/default/lib/help/language/</td>
<td>contains Knowledge Module help files for display in PATROL 7.x consoles.</td>
</tr>
<tr>
<td>$PATROL_ROOT/lib/mof/</td>
<td>the directory for MOF configuration files used by PATROL Console Server. This directory can be redefined by the PATROL_ROOT_MOF environment variable.</td>
</tr>
</tbody>
</table>
### Table 38  Directories and Files for PATROL Console Server on Unix (Part 5 of 6)

<table>
<thead>
<tr>
<th>Directories and Files</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>$PATROL_ROOT/lib/mof/cserver/</td>
<td>default managed system queries for new management profiles</td>
</tr>
<tr>
<td>default_agent_queries.mof</td>
<td></td>
</tr>
<tr>
<td>default_lem_filters.mof</td>
<td>default event filters for new management profiles</td>
</tr>
<tr>
<td>default_chart_schemes.mof</td>
<td>default chart schemes for new management profiles</td>
</tr>
<tr>
<td>$PATROL_ROOT/lib/nls/</td>
<td>the directory for message catalogs used by PATROL Console Server</td>
</tr>
<tr>
<td></td>
<td>This directory can be redefined by the PATROL_ROOT_NLS environment variable.</td>
</tr>
<tr>
<td>$PATROL_ROOT/lib/psl/</td>
<td>the directory for PSL7 files</td>
</tr>
<tr>
<td></td>
<td>This directory can be redefined by the PATROL_ROOT_PSL_APPS environment variable.</td>
</tr>
<tr>
<td>$PATROL_ROOT/log/</td>
<td>the directory for log files generated by the PATROL Console Server as well as subdirectories that contain persistent data for various components of the PATROL Console Server, including:</td>
</tr>
<tr>
<td></td>
<td>- management profiles</td>
</tr>
<tr>
<td></td>
<td>- service cache databases</td>
</tr>
<tr>
<td></td>
<td>- managed system group definitions</td>
</tr>
<tr>
<td></td>
<td>This directory can be redefined by the PATROL_ROOT_LOG environment variable.</td>
</tr>
</tbody>
</table>
### Table 38  Directories and Files for PATROL Console Server on Unix (Part 6 of 6)

<table>
<thead>
<tr>
<th>Directories and Files</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>$PATROL_ROOT/log/cserver/</code></td>
<td></td>
</tr>
<tr>
<td>cserver-serviceID.log</td>
<td>contains console server log messages</td>
</tr>
<tr>
<td><code>$PATROL_ROOT/log/cserver/layout/</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>the directory used by PATROL Console Server to store management profile data</td>
</tr>
<tr>
<td></td>
<td>This directory can be redefined by the <code>PATROL_ROOT_LAYOUT</code> environment variable.</td>
</tr>
<tr>
<td><code>$PATROL_ROOT/log/history/</code></td>
<td></td>
</tr>
<tr>
<td>PATROL_CSERVER_serviceID_events.mk</td>
<td>history files for console server events</td>
</tr>
<tr>
<td><code>$PATROL_ROOT/log/layout/PATROL_CSERVER_serviceID/</code></td>
<td>contains management profiles</td>
</tr>
<tr>
<td><code>$PATROL_ROOT/security</code></td>
<td>security files</td>
</tr>
<tr>
<td><code>$PATROL_ROOT/tmp/</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td>the directory used by PATROL Console Server to create temporary files</td>
</tr>
<tr>
<td></td>
<td>This directory can be redefined by the <code>PATROL_ROOT_TMP</code> environment variable.</td>
</tr>
</tbody>
</table>
### Soft Links on Unix

On Unix, if the installation directory is not the default directory of `/opt/bmc`, then the installation creates the following soft links under `/opt/bmc` which point to the actual installation directories. If the installation directory is the default directory of `/opt/bmc`, then no soft links are created.

Table 39 lists the soft links for PATROL Console Server on Unix.

#### Table 39: Soft Links for PATROL Console Server on Unix

<table>
<thead>
<tr>
<th>Soft Link</th>
<th>Target</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>/opt/bmc/common/bmc/bin/target</code></td>
<td><code>$BMC_ROOT/common/bmc/bin/target</code></td>
</tr>
<tr>
<td><code>/opt/bmc/common/globalc/lib/target</code></td>
<td><code>$BMC_ROOT/common/globalc/bin/target</code></td>
</tr>
<tr>
<td><code>/opt/bmc/Patrol7/bin/target</code></td>
<td><code>$BMC_ROOT/Patrol7/bin/target</code></td>
</tr>
</tbody>
</table>

These soft links are created because the PATROL Console Server runs as a setuid root process, and consequently does not inherit LD_LIBRARY_PATH settings specified in the `start_cserver.sh` script. Therefore, the PATROL Console Server executable has hard-coded these directory names in its library search path.

### RTserver Environment Variables, Directories, and Files

This section lists the environment variables, directories, and files that are specific to the RTserver.

#### Environment Variables Used by the RTserver

Table 40 lists the environment variable names and their default values as used by the RTserver on Windows.

#### Table 40: RTserver Environment Variables on Windows

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC_ROOT</td>
<td>specified at installation. See “Installation Directory” on page 33.</td>
</tr>
<tr>
<td>RTHOME</td>
<td>%BMC_ROOT%\common\smartsockets</td>
</tr>
</tbody>
</table>
Table 41 lists the environment variable names and their default values as used by the RTserver on Unix.

### Table 41 RTserver Environment Variables on Unix

<table>
<thead>
<tr>
<th>Environment Variable</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>BMC_ROOT</td>
<td>specified at installation. See “Installation Directory” on page 33.</td>
</tr>
<tr>
<td>RTHOME</td>
<td>$BMC_ROOT/common/smartsockets</td>
</tr>
</tbody>
</table>

### RTserver Directories and Files

This section lists directories that are specific to the RTserver. It also lists selected files of particular importance for using or configuring the RTserver. It does not list other directories or files for other PATROL components, such as the PATROL Agent or PATROL console themselves. Note that some directories and files are shared by multiple PATROL components.

Most of the directories and files are created during the installation of the base PATROL component (such as the PATROL Agent), or during the installation of the RTserver. If a directory or file is created after installation, when it is created is noted in the comments.

Table 42 lists the directories and files for the RTserver on Windows.

### Table 42 Directories and Files for the RTserver on Windows

<table>
<thead>
<tr>
<th>Directories and Files</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>%RTHOME%\bin\platform\rtserver.exe</td>
<td>the executable binary</td>
</tr>
<tr>
<td>%RTHOME%\standard\rtserver.cm</td>
<td>contains the RTserver configuration data</td>
</tr>
<tr>
<td>%RTHOME%\log\rtserver.log</td>
<td>contains RTserver log messages</td>
</tr>
</tbody>
</table>
Table 43 lists the directories and files for the RTserver on Unix.

### Table 43  Directories and Files for the RTserver on Unix

<table>
<thead>
<tr>
<th>Directories and Files</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>$RTHOME</td>
<td></td>
</tr>
<tr>
<td>start_rtserver.sh</td>
<td>the script for starting the RTserver</td>
</tr>
<tr>
<td>stop_rtserver.sh</td>
<td>the script for stopping the RTserver</td>
</tr>
<tr>
<td>$RTHOME/bin/platform/</td>
<td></td>
</tr>
<tr>
<td>rtserver</td>
<td>the executable binary</td>
</tr>
<tr>
<td>$RTHOME/standard/</td>
<td></td>
</tr>
<tr>
<td>rtserver.cm</td>
<td>contains the RTserver configuration data</td>
</tr>
<tr>
<td>$RTHOME/log/</td>
<td></td>
</tr>
<tr>
<td>rtserver.log</td>
<td>contains RTserver log messages</td>
</tr>
<tr>
<td>start_rtserver.log</td>
<td>contains RTserver startup messages</td>
</tr>
</tbody>
</table>
# Index

## Symbols

% CS commands 143

## A

about
- command line utilities 176
- Distribution Server 32
- impersonation 71
- installation utility 31
- job specification strings 128
- logical connection names 232
- PATROL Console Server account 33
- RTserver account 37
- RTSERVERS environment variable 35
- security levels 36

access
- control database 75
- permissions
  - about 75
  - administering 184

accounts
- logging on
  - for installation 44
  - with domain account 276
- order for PATROL Agent authentication 83
- PATROL Console Server account 33
- PATROL default account 33
- root account 35
- RTserver account 37

acfg_cserver_serviceID.mof file 81

ACLs
- on the PATROL Agent 60
- operator 60
- saving at regular intervals 119

admin_copy utility 177

admincli
- changing PATROL Console Server account or password 77
- utility 180

administering (continued)
- managed system group operations 190
- management profiles 140
- advanced RTserver options 249
- AIX
  - supported combinations 30
  - supported versions 28
- alias table administration 186
- architecture
  - multi-cloud configuration 22
  - planning RTserver clouds 229
  - single-cloud configuration 21
- assigning rights 186
- audit logs
  - configuring 87
  - options 88
- auditLevel 88
- authentication
  - about 70
  - of users 19
  - problems 278
  - process 71
- auto-location
  - problems 286
  - using 260
- autoSaveInterval 116, 119
- autoSaveMode 101
- autoSaveTimer 102
- autoScanServices 85, 116

## B

backups
- data and configuration files 121
- incremental 125
- manual 120
  - full 121
  - online 191
- online 125
- refreshing copy of 171
- using admincli 191

BMC Software, contacting 2

built-in commands 143
changing
  default account 185
  PATROL Console Server account or password 76
  RTserver port number 267
  service ID 136
chart schemes, creating 141
checkAccessControlForParameters 102
choosing service IDs 269
cleanupInterval 116
collapsing top-level objects 112
collapsing top-level children limit 102, 112
computers per management profile 112
compression
  configuration option 118
  configuring 118
collapsing top-level children limit 102, 112
command line arguments
  admin_copy utility 179
  admincli utility 183
  confManager utility 195
  hg_export utility 204
  hg_import utility 202
  lt_counters utility 207
  lt_db_utility 217
  lt_export utility 223
  lt_import utility 219
  PATROL Console Server 67
  rtservice 259
  sec_encrypt_p3x utility 198
  stopPatrolService utility 200
collapsing top-level objects 112
collapseTopLevelChildrenLimit 102, 112
client connections, configuring 245
client_max_buffer 251
clouds, planning architecture 229
cleanupInterval 116
collapsing top-level objects 112
command line arguments
  admin_copy utility 179
  admincli utility 183
  confManager utility 195
  hg_export utility 204
  hg_import utility 202
  lt_counters utility 207
  lt_db_utility 217
  lt_export utility 223
  lt_import utility 219
  PATROL Console Server 67
  rtservice 259
  sec_encrypt_p3x utility 198
  stopPatrolService utility 200
utilities
  about 176
  admin_copy 177
  admincli 180
  confManager 194
  for management profiles 206
general 177
  hg_export 204
  hg_import 202
  lt_counters 206
  lt_db_utility 216
  lt_export 222
  lt_import 218
  sec_encrypt_p3x 198
  stopPatrolService 199
commands, built-in 143
communications
  enabling PATROL Agent 61
  RTserver 232
compression
  configuration option 118
  configuring 118
A   B   C   D   E   F   G   H   I   J   K   L   M   N   O   P   Q   R   S   T   U   V   W   X   Y   Z

RTserver
about 227
advanced options 249
clouds 239
instances 253
listening for connections 236
scanning for PATROL Agent information 85
scheduled online backups 126
security levels on multiple RTserver clouds 99
service IDs 268
service names 268
single RTserver cloud connection 90

confManager utility 194
conn_names 252
conn_names option 237
connecting
failure with PATROL Agents 3.5.00 278
multiple RTserver cloud 93
RTclient problems 284
single RTserver cloud 90
uninitialized connection 275
conTimeout 92, 97
controlling scanning 85
crashes, under heavy processing load 279
creating
additional process thread pools 117
chart schemes 141
copy for PATROL Console Server instances 173
event filters 141
managed system queries 141
customer support 3

data and configuration files to back up 121
-debug 67
debug information 291
default
PATROL Console Server account about 33
changing 185
PATROL groups 73
RTserver instance, removing 257
-detach 67
determining version
PATROL Console Server 291
RTserver 292
diagnostic information 288
different security levels 36
directories
moving 153
used by
PATROL Console Server 296
RTserver 308
disabling scanning 85
disconnection, PATROL Central Operator or management profiles 281
Distribution Server 32
DLLs, cannot find 280
dllVersion utility problems 280
documentation, related 23
domain accounts
logging on 276
PATROL Console Server account 34

e
editing
rtserver.cm file 233
startup configuration file 82
enabling
multiple clouds 93
scanning 85
encrypting passwords 198
environment variables
moving PATROL Console Server directories 153
RTSERVERS 35, 243
setting 139
used by
RTserver 307
used by PATROL Console Server 294
EPD site, installing from 43
event filters, creating 141
examples
changing an RTserver port number 267
cluster failover scenarios 160
different security levels on multiple clouds 100
forming an RTserver cloud 241
job specification strings
invalid 135
valid 132
multi-cloud configuration 98
overload protection 111
rtserver.cm file 234
setting up
RTserver failover for RTclients 262
RTservers in a firewall environment 264
exporting
managed system groups 204
management profiles 222

f
failover
cluster environment 148
manual 147
RTservers and RTclients 262
failures
object queries 287
secure channels 282
failures (continued)
starting
  on NFS file system 274
  on Unix 274
  on Windows 273
features
  PATROL Console Server 16
  RTserver 20
files
  data and configuration backup 121
  full backup 121
  used by
    PATROL Console Server 296
    RTserver 308
firewalls, RTservers and 263
format, service names 268
full backups
  manual 121
  online 125

G
general command line utilities 177
groups
  on PATROL Console Server 72
  setting up 70

H
-h 67
hangup signal 275
hardware requirements 30
heavy processing loads, crashes 279
help 67
Help files for KMs 17
hg_export utility 204
hg_import utility 202
hot patch for Windows 2000 SP2 279

I
icons for KMs 17
-id 67
impersonation
  about 71
  example table
    with named users 87
    with wildcard users 86
  files, saving at regular intervals 119
  of users 19
  table administration 187
importing managed system groups 202
incremental backups 125
-install 68
installable components 27

J
Japanese resource files
  about 38
  installation directory 38
job specification strings
  about 128
  examples
    invalid 135
    valid 132
  fields 129
  valid and invalid values 131

K
keepAliveTimeout 93, 97
KMs
  icons
    about 17
  cannot locate 281
  online Help files
    about 17
  related resources 17
Knowledge Modules. See KMs
Known Services Cache. See KSC
KSC 115
  configuration options 116
  configuring 115
kscRemoteUserContext 85, 116
L

Linux, supported versions 28
listing user sessions 190
lockd daemon 33
logging on, domain accounts 276
logging options
   RTserver 246
logical connection names
   about 232
   conn_names option 236
   multiple IP addresses 238
   RTSERVERS variable 244
   server_names option 240
logs 288
   installation 289
   PATROL Console Server 290
   RTserver 290
lt_counters utility 206
lt_db_util utility 216
lt_export utility 222
lt_import utility 218

M

managed system
   groups
      administration 190
      exporting 204
      importing 202
      queries, creating 141
management profiles
   administrating 140
   and upgrading 39
   command line utilities for 206
   configuration options 101
   configuring 100
   database file 216
   disconnect 281
   exporting 222
   importing 218
   limiting size 114
   performance counters 206
   storing 16
   too large 282
   managing
      permissions 75
      rights 73
manual
   backups 120
   failover 147
   online backup 191
maxAppLogFileSize 88
maxAuditLogFileSize 89
maxComputerCount 101, 110
maxMessagesInAppLog 88
maxMessagesInAuditLog 89
maxObjectCount 101, 110
maxProfilesOpen 101, 110
maxReadWriteProfilesOpen 101, 110
maxVirtualMemorySize 101, 109
missing KM Help 281
missing KM icons 281
mlmAuthOrder 83
MMC snap-in 44
monitoring requests 18
moving
   PATROL Console Server
      directories for shared drive 153
      to different computer 138
   RTserver configuration directory 155
msgSuppressMask 103
multi-cloud configuration
      architecture 22
      configuring 93
      example 98
multiple
   IP addresses 238
   RTserver clouds 93
   RTservers on same computer 252

N

network name, Windows cluster 156
newLayoutPreloadList 101
NFS file system 33
   -nodetach 67
   -nostderlog 69

O

object queries fail 287
online backups
   about 125
   manual 191
   scheduled 126
operator ACLs 60
overload protection 109

P

PAM support 63
passwords
   changing for PATROL Console Server account 76
   encrypting 198
   root account 35
patadm group 73
patop group 73
patpop group 73
PATROL 7.x2
architecture 20
supported consoles 16

PATROL Agents
account order for authentication 83
ACLs on 60
authenticating to 83
connecting to 83
enabling communication 61
preparing to use 60
RTclients 232
scanning for information 85
setting RTSERVERS environment variable 61
tasks for using with PATROL Console Server 60
version 3.5.00, connection failure 278
viewing all events 142

PATROL Central Operator
disconnects 281
RTclient 232
running PATROL Console Server commands 143
upgrading PATROL Console Server 39

PATROL Console Server
account
about 33
changing 76
requirements 34
administering and configuring 57
backups
directories 121
manual 120
online 125
built-in commands 143
changing service ID 136
cluster environment 145
command line arguments 67
command line utilities
about 176
for management profiles 206
general 177
components 27
configuration options 81
configuring
for Windows cluster 156
PAM support 63
resource in Windows cluster 158

crashes 279
creating copy for other instances 173
default PATROL groups on 73
determining version 291
directories and files, table of 296
environment variables
for moving directories 153
table of 294
fails to start
on NFS file system 274
on Unix 274
on Windows 273

PATROL Console Server (continued)
features 16
hardware requirements 30
installation
directory 33
how to 43
overview 25
log files 290
moving 138
overload protection 109
preparing to use 59
refreshing backup copy 171
replicating 171
required information
for installing 33
for uninstalling 55
restoring 120
role 16
RTclient 232
running commands from PATROL Central Operator 143
service 64
service ID 64
setting up in Windows cluster environment 150
soft links 307
starting and stopping
on Unix 65
on Windows 64
system requirements 28
tasks for
authentication 19
impersonation 19
monitoring environment 18
permissions 19
user rights 19
troubleshooting 271
upgrading 39
users and groups on 72
verifying it is running 65

PATROL default account 33
PATROL Knowledge Modules. See KMs

patrol.conf file
encrypting passwords 198
environment variables 139
modifying with confManager 194

PATROL Console Server account 76

PATROL_ROOT_CONFIG 153
PATROL_ROOT_ETC 153
PATROL_ROOT_LAYOUT 153
PATROL_ROOT_LOG 153
PATROL_ROOT_TMP 153
patscadm group 73
patwatch group 73
percentAddressSpaceLimit 101, 109
permissions
administering 184
managing 75

316  PATROL Console Server and RTserver Getting Started
planning RTserver cloud architecture 229
pop-up blocker software 44
ports
changing for RTserver 267
RTserver 37
preventing PATROL Console Server overload 109
primaryBackupPath 127
primaryBackupSchedule 127
privilegedUserMaxComputersPerProfile 101, 115
problems
authentication 278
secure channels 282
process thread pools, configuring 117
processing monitoring requests 18
product support 3
protocol compression
configuration option 118
configuring 118
root
access, providing after installation utility 54
account 35
RTclients
auto-location 260
configuring connections 245
connecting to RTservers 243
connection problems 284
examples 232
failover 262
firewalls 263
service names 268
RTserver
accepting connections 236
account 37
advanced options 249
auto-location 260
clouds
architecture, planning 229
configuring 239
example 241
communications 232
configuring
clouds 239
firewall environment 263
for Windows cluster 156
number of client connections 245
overview 227
resource in Windows cluster 159
controlling which is used 80
determining version 292
directories and files, table of 308
environment variables 307
failover 262
features 20
firewalls 263
forming a cloud 240
installation
as Windows service 257
directory 37
how to 43
overview 25
required information 37
instances 253
listening for connections 236
log files 290
moving configuration directory 155
multiple on same computer 252
multiple-cloud configuration 93
order for starting and stopping 229
port number
about 37
changing 267
removing default instance 257
role 20
service does not restart 286
setting up in Windows cluster environment 150

Q
queryBatchInterval 107
queryBatchSize 107

R
Real Time server. See RTserver
Red Hat Enterprise Linux, supported versions 28
refreshing backup copy 171
-remove 69
removing
default RTserver instance 257
replicating PATROL Console Server 171
required information
installing
PATROL Console Server 33
RTserver 37
uninstalling
PATROL Console Server 55
RTserver 55
requirements
hardware 30
PATROL Console Server account 34
platform support 28
resource
for PATROL Console Server in Windows cluster 158
for RTserver in Windows cluster 159
group in Windows cluster 157
restart problems 286
restoring PATROL Console Server 120
retentionInterval 116
rights
assigning 186
managing 73
RTserver (continued)
  single-cloud configuration 90
  specifying logging options 246
  standalone mode 239
  starting and stopping
  on Unix 231
  on Windows 230
  system requirements 28
  troubleshooting 271
  uninstalling 55
  upgrading
  about 40
  how to 42
  -rtserver 69
  rtserver.cm file 233
  rtServerLocators 91, 95
  rtServerNamespaceId 95
  RTSERVERS environment variable
  about 35
  list of RTservers from 243
  setting on PATROL Agents 61
  rtservice utility 258
  running multiple RTservers on computer 252

S
  saving ACLs and impersonations at regular intervals 119
  scanning PATROL Agent information 85
  schedule strings
    about 128
    fields 129
    invalid examples 135
    valid and invalid values 131
    valid examples 132
  scheduled online backups 126
  scripts
    config_cserver.sh 54
    config_rtservers.psl 61
  sec_encrypt_p3x utility 198
  secondaryBackupPath 127
  secondaryBackupSchedule 127
  secure channel problems 282
  security levels
    about 36
    configuring on multiple RTserver clouds 99
    interoperability 36
  server_keep_alive_timeout 250
  server_names 252
  server_names option 239
  server_read_timeout 250
  server_reconnect_interval 249
  service IDs
    changing 136
    choosing 269
    choosing for PATROL Console Server 64
    configuring 268
  service IDs (continued)
    for Windows cluster 157
    requirements 270
    uses 269
  service names
    configuring 268
    format 269
  serviceName 92, 96
  serviceNameAliases 92, 96
  services
    installing PATROL Console Server as 68
    MMC snap-in 44
    PATROL Console Server 64
    removing PATROL Console Server as 69
    SmartSockets RTserver 230
  setting environment variables 139
  setting up
    PATROL Console Server and RTserver on Windows
    cluster environment 150
    RTclients for RTserver failover 262
    users and groups 70
  single RTserver cloud 90
  single-cloud configuration
    architecture 21
    configuring 90
  SmartSockets RTserver service 230
  soft links, on Unix 307
  Solaris, supported versions 28
  specifying RTserver logging options 246
  starting
    multiple RTservers
      on Unix computer 255
      on Windows computer 256
    PATROL Console Server
      on Unix 65
      on Windows 64
    RTserver
      on Unix 231
      on Windows 230
  startup
    configuration file 82
    failures
      on NFS file system 274
      on Unix 274
      on Windows 273
  stopPatrolService utility 199
  stopping
    multiple RTservers
      on Unix computer 255
      on Windows computer 256
    PATROL Console Server
      on Unix 65
      on Windows 64
    RTserver
      on Unix 231
      on Windows 230
Index 319

A   B   C   D   E   F   G   H   I   J   K   L   M   N   O   P   Q   R   S   T   U   V   W   X   Y   Z

storing
  KM-related resources 17
  management profiles 16
  permissions 19
  rights 19

subject name ABCDEFG 283

support, customer 3

supported platforms
  AIX 28
  Red Hat Enterprise Linux 28
  Solaris 28
  Windows 2000 29
  Windows Server 2003 29

system requirements 28

T

technical support 3
  thread pools, configuring 117
  threadPoolSize 117
  top-level objects, collapsing 112
  trace_file 253
  troubleshooting 271

U

uninitialized connection 275
  -uninstall 69
  uninstalling 55

Unix, soft links on 307

unprivilegedUserMaxComputersPerProfile 101, 115

upgrading
  about 38
  PATROL Console Server 39
  RTserver
    about 40
    how to 42
  useAuditLog 89
  useCloudSpecificCommSecurity 95, 99
  useP3xLabelsWhenObjectNotHome 102

users
  accounts and groups 72
  authenticating 19, 70
  impersonating 19, 71
  listing sessions 190
  setting up 70

utilities
  about 176
  admin_copy 177
  admincli 180
  confManager 194
  for management profiles 206
  general 177
  hg_export 204
  hg_import 202

utilities (continued)
  installation 31
  lt_counters 206
  lt_db_util 216
  lt_export 222
  lt_import 218
  rtservice 258
  sec_encrypt_p3x 198
  stopPatrolService 199

V

-v 69

valid job specification strings 131

verboseConsoleMessages 103

verifying
  cluster failover 159
  PATROL Console Server
    running on Unix 66
    running on Windows 65
  uses shared directories 154

versions, determining
  PATROL Console Server 291
  RTserver 292

viewing PATROL Agent events 142

W

Windows 2000
  hot patch 279

Windows cluster, PATROL Console Server and RTserver
  in 156

Windows Server 2003, supported versions 29

worksheets for installation 41
Third Party Legal Notices and Disclaimers

This section contains third-party legal notices and disclaimers for the PATROL Console Server and RTserver.

**zlib.h**

Interface of the ‘zlib’ general purpose compression library version 1.2.2, October 3rd, 2004

Copyright (C) 1995-2004 Jean-loup Gailly and Mark Adler

This software is provided ‘as-is’, without any express or implied warranty. In no event will the authors be held liable for any damages arising from the use of this software.

Permission is granted to anyone to use this software for any purpose, including commercial applications, and to alter it and redistribute it freely, subject to the following restrictions:

1. The origin of this software must not be misrepresented; you must not claim that you wrote the original software. If you use this software in a product, an acknowledgment in the product documentation would be appreciated but is not required.

2. Altered source versions must be plainly marked as such, and must not be misrepresented as being the original software.

3. This notice may not be removed or altered from any source distribution.

Jean-loup GaillyMark Adler
jloup@gzip.orgmadler@alumni.caltech.edu

The data format used by the zlib library is described by RFCs (Request for Comments) 1950 to 1952 in the files http://www.ietf.org/rfc/rfc1950.txt (zlib format), rfc1951.txt (deflate format) and rfc1952.txt (gzip format).