Hexis
CYBER SOLUTIONS
a KEYW company

Administration Guide

HawkEye AP 5.0.1

February 5, 2014

Hexis Cyber Solutions, Inc.
2800 Campus Drive, Suite 150
San Mateo, CA 94403
+1 (650) 830-0484
fax: +1 (650) 631-2810
www.hexiscyber.com

Copyright

This document is unpublished and the foregoing notice is affixed to protect Hexis Cyber Solutions, Inc. in the event of inadvertent publication. All rights reserved.

No part of this document may be reproduced in any form, including photocopying or transmission electronically to any computer, without prior written consent of Hexis Cyber Solutions, Inc. The information contained in this document is confidential and proprietary to Hexis Cyber Solutions, Inc., and may not be used or disclosed except as expressly authorized in writing by Hexis Cyber Solutions, Inc.

Trademarks

The Hexis Cyber Solutions® logo and HawkEye AP are registered trademarks of Hexis Cyber Solutions, Inc., in the United States. Other company, product names or logos mentioned or reproduced in this document may be trademarks or registered trademarks of their respective companies and are hereby acknowledged.

Patents

The software described in this manual contains technology protected by U.S. Patent 7,024,414.

Permitted Use

This document is provided by Hexis Cyber Solutions, Inc., and/or its subsidiaries to entities using the services described herein pursuant to a separate services and/or software license agreement. This document may be used by such entities only for the limited purpose of using the services described herein as authorized by such services agreement. Any other use of this document is strictly prohibited, unless expressly authorized by Hexis Cyber Solutions, Inc., in writing.

Disclaimer and Limitation of Liability

Except as otherwise expressly provided in the applicable services and/or software license agreement, Hexis Cyber Solutions, Inc., and its subsidiaries make no representations or warranties of any kind, whether express or implied, regarding the accuracy of this document and disclaim the implied warranties of merchantability, fitness for a particular purpose, title and non-infringement of third party rights, to the maximum extent permitted by law. Hexis Cyber Solutions, Inc., and its subsidiaries assume no responsibility for any damage or loss resulting from the use of this document. This document is subject to revision from time to time.

Confidential Information

The information contained in this document is confidential and proprietary to Hexis Cyber Solutions, Inc., and its subsidiaries and may not be used or disclosed except as expressly authorized in writing by Hexis Cyber Solutions, Inc. This document is released to the possessor on a strictly secure basis. No authority, express or implied, exists to permit discussion with any third party. By continuing to possess this document, the bearer accepts all direct and indirect responsibility for keeping these materials in confidence. The bearer acknowledges the implicit material commercial harm to Hexis Cyber Solutions, Inc., of unauthorized disclosure of the contents herein, including an acknowledgment of responsibility for such damages caused to Hexis Cyber Solutions, Inc., if any.

Destination Control Statement

ECCN (Export Control Classification Number):5D002 ENC

These commodities, technology, or software were exported from the United States in accordance with the Export Administration Regulations. Diversion contrary to United States law is prohibited.

TABLE OF CONTENTS

PREFACE

This book, the *Administration Guide*, describes the 5.0.1version of HawkEye AP software.

System Administrators manage the following components and features of a HawkEye AP deployment, which are discussed in this manual:

- Event Data Warehouse (EDW)
- HawkEye AP Console
- HawkEye AP Real-Time
- HawkEye AP Application Manager
- HawkEye Collector
- Host software—some event sources require installation and configuration of HawkEye AP software on host machines; HawkEye AP also facilitates monitoring of source health
- User management—creation of users and roles and configuration of their authentication
- Nearline storage—configuration and management of nearline storage devices for ready access to archived data
- Assets and system alerts—creation of enterprise assets and monitoring of system alerts and health

This Preface contains the following sections:

Audience for this Book

This book is directed to system administrators who manage a HawkEye AP deployment. Administrators should be familiar with Linux administration including: SSH, managing users and groups, file system permissions and management, network configuration, and using text editors such as VI or emacs.

Administration Guide Organization

This book contains the following chapters:

- Chapter 1: Introduction—Provides an overview of HawkEye AP system administration tasks and HawkEye AP architecture
- Chapter 2: Configuring and Managing HawkEye AP—Describes the utilities that HawkEye AP provides for comparing files, listing hosts in an EDW instance, collecting status information,

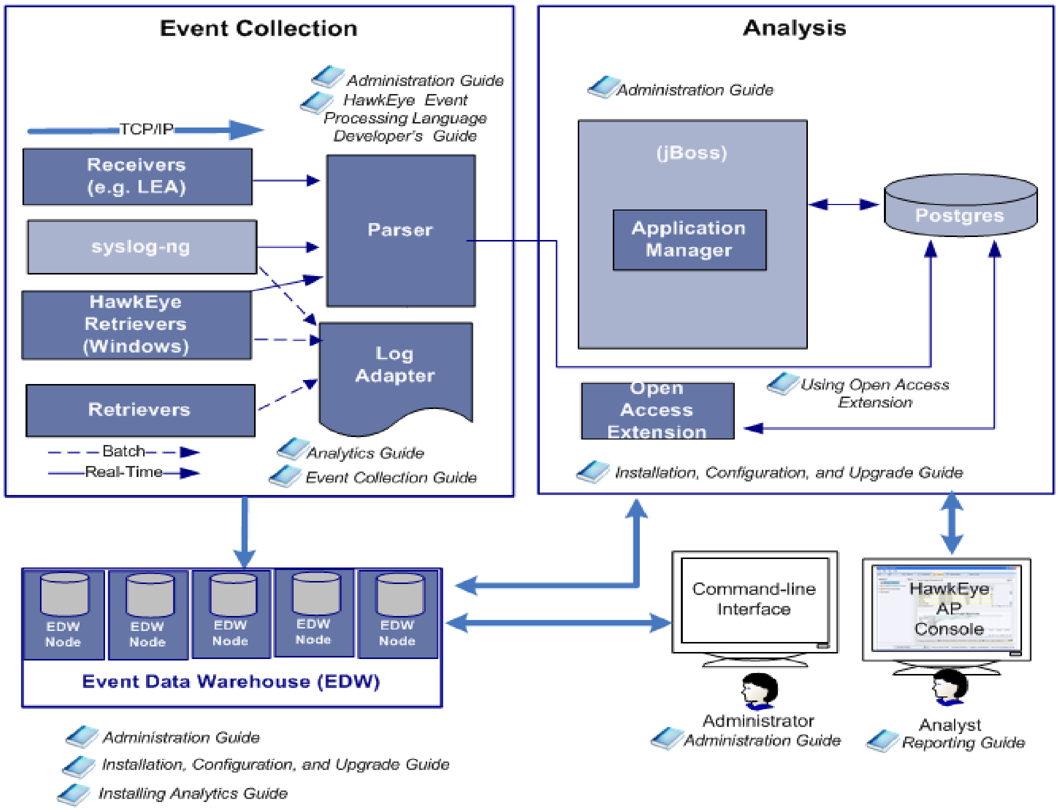
setting up and managing EDW instances, running commands across the hosts in an EDW instance, copying files and directories across the hosts in an EDW instance, and viewing operating system status across all hosts.

- Chapter 3: Loading, Querying, and Managing the EDW—Describes the utilities that HawkEye AP provides for loading, querying, managing, viewing, and retiring data; also describes the system tables
- Chapter 4: Administering an EDW Instance—Describes how to back up and restore EDW tables, restore hosts in an EDW instance, monitor the system, and archive data.
- Chapter 5: Recovering a Failed EDW Node—Describes the process for handling a failed EDW node.
- Chapter 6: Administering HawkEye AP Console and the Application Manager—Describes how to access and manage HawkEye AP Console and the Application Manager.
- Chapter 7: Administering the Collector—Describes how to perform various administrative tasks for the optimal operation of the Collector, including starting the Collector, monitoring Collector operational logs for errors, handling bad load operations, and backing up processed log files.
- Chapter 8: Administering Users and Authentication—Describes how to configure authentication and authorization and how to create and administer users, roles and permissions.
- Chapter 9: Archiving to Nearline Storage—Describes how to archive data to nearline storage.
- Chapter 10: Administering Assets and Monitoring Alerts—Describes how to create and manage enterprise and HawkEye AP system assets and how to monitor security, system, and exceptions alerts.
- Chapter 11: Monitoring Source Health—Describes source health monitoring, which monitors log sources for potential collection failures; documents how to configure your EDW instance to monitor specific source types.
- Chapter 12: Troubleshooting—Describes basic troubleshooting.
- Appendix A: Log File Appendix—Presents reference material on log files.
- Appendix B: Error Codes—Presents reference material on error codes.
- Appendix C: Time Zones—Presents reference material on time zones.

Road Map to HawkEye AP Documentation

This document, the *Administration Guide*, is part of the larger documentation set of your HawkEye AP system. Figure P-1 illustrates HawkEye AP components and modules in the context of their function within the HawkEye AP system.

Figure P-1: Road Map to HawkEye AP Documentation



The table below describes all the manuals in the HawkEye AP documentation set and the user roles to which they are directed.

Role	Tasks	Documentation
Business Analyst	• Use and create dashboards • View and create reports • Monitor enterprise and exception alerts • Schedule reports & dashboards • Create Alerting Rules from Templates	*Reporting Guide*
Business Analyst	• Learn about Analytics • Use IntelliSchema views • Learn about the Foundation and Compliance Analytics packages • Learn about additional Analytics packages	*Analytics Guide*

Role	Tasks	Documentation
Report Developer or Security Analyst	• Use Sensage SQL, Sensage SQL functions, and libraries to create reports or query the EDW • Create and use Perl code in Sensage SQL statements • Use the DBD Driver to query HawkEye AP from other locations	*Reporting Guide*
Security System Administrator	• Configure retrievers, receivers, and collectors • Configure HawkEye Retriever • Create log adapter PTL files	*Event Collection Guide*
Security Analyst	• Create parsing rules, alerting rules, and configurable alerting rule templates • Manage rules	*HawkEye Event Processing Language Developers Guide*
System Administrator	• Install HawkEye AP • Configure HawkEye AP and its components	*Installation, Configuration, and Upgrade Guide*
System Administrator	• Install Analytics	*Installing Analytics*
System Administrator	• Manage the HawkEye Event Data Warehouse (EDW) • Manage the Collector • Manage users, groups, and permissions • Archive to nearline storage • Manage assets & monitor security alerts • Monitor log source health • Monitor system health • Troubleshoot • Error Messages	*Administration Guide*
System Administrator	• Install and configure log adapters	*Analytics Guide*
Developer	• Access EDW data using open standards as ANSI SQL, ODBC, and JDBC	*Using Open Access Extension*
Legal	Monitor third-party licenses	*Third-Party Open Source Licensing*

TIP: You can access the manuals listed above from:

- HawkEye AP Console online help

  Click **Help** > **Help Contents**.

- HawkEye AP Welcome page

  Click the **Documentation** hyperlink.

For more details, see "Logging into HawkEye AP Console", on page 183.

Conventions Used in HawkEye AP Documentation

This convention...	Indicates...	Example
bold text	Names of user interface items, such as field names, buttons, menu choices, and keystrokes	Click **Clear Filter**.
italic text	Indicates a variable name or a new term the first time it appears	`http://`*`<host>`*`:`*`<port>`*`/index.mhtml`
`Courier text`	Indicates a literal value, such as a command name, file name, information typed by the user, or information displayed by the system	`atquery localhost:8072 myquery.sql`
SMALL CAPS	Indicates a key on the computer keyboard	Press ENTER.
`{ }`	In a syntax line, curly braces surround a set of options from which you must choose one and only one. **NOTE**: Syntax specifications for SELECT statements include curly braces as part of the `{INCLUDE_BAD_LOADS]` keyword.	`{ start | stop | restart }`
`[ ]`	In a syntax line, square brackets surround an optional parameter	`atquery [options] `*`<host>`*`:`*`<port>`*` -`

`|`	In a syntax line, a pipe within square brackets or curly braces separates a choice between mutually exclusive parameters
	NOTE: Syntax for defining a Nearline Storage Address (NSA) includes a pipe.
		`{ start | stop | restart }`
		`[g|m]`

`...`	In a syntax line, ellipses indicate a repetition of the previous parameter	The following example indicates you can enter multiple, comma-separated options: *`<option>`*`[, `*`<option>`*`[…]]`

backslash (\)	A backslash in command-line syntax or in a command example behaves as the escape character on Unix. It removes any special meaning from the character immediately following it. In HawkEye AP documentation, a backslash nullifies the special meaning of the newline character as a command terminator. Without the backslash, pressing ENTER at the end of the line causes the Unix system to execute the text preceding the ENTER. Without the backslash, you must allow long commands to wrap over multiple lines as a single line.	`atquery --user=administrator \`
		`--pass=pass:p@ss localhost:8072\`
		`-e='SELECT * FROM system.users;'`

LICENSE

Beginning with HawkEye AP 5.0.0 , you were required to have a license key to run your cluster. If you upgrade to the 5.0.1 version of HawkEye AP without a license key the system starts up, but you cannot query the data out of the EDW Each EDW cluster will require its own, valid license key.

Your license is contained in a license key file that you must copy to each EDW node in your cluster. A successful query against your EDW cluster confirms that you have correctly installed a valid license key for your cluster.

If you haven’t received your license key, contact your Hexis Cyber Solutions representative. When you receive the key, you need to copy it over to the following placeholder files on each EDW node in your cluster.

```
<install-root>/latest/etc/sls/instance/<cluster-name>/LICENSE
```

When the new license key, LICENSE, is copied to all EDW nodes, you can restart each EDW normally.

Do not alter the license key file in any way. Even a whitespace change will render the file invalid as the embedded digital signature is computed against the exact sequence of bytes in the original license key file.

If there is an issue with the LICENSE file, you will see one of the following errors when submitting queries to the EDW cluster:

Trial License has expired - The license was tied to a trial period, which has expired.

Missing or Invalid License -The LICENSE file specified in the athttpd.conf file on some node is not valid or is missing.

EDW configured for more cores then the license allows - The number of physical cores that the cluster is configured to use is larger than the licensed limit.

Product is operating under a Trial License and evidence of clock tampering has been noted - the system is running under a trial license and there is evidence that the system clock has been pushed back.

NOTE: "Cluster size" is defined in terms of physical cores. Many Intel CPUs support hyperthreading in which a single physical core appears to the Linux kernel as two logical cores. The cluster.xml file defines how many logical cores each EDW node will use. The EDW licensing logic looks at the cluster.xml file and the hardware that the EDW is running on to determine the total number of physical cores that the cluster has been configured to use. Note that if the cluster.xml file does not specify a number of cores for an EDW node, it is assumed that it will use all cores on that server.

If you need to resolve a licensing issue, contact Sensage support.

CONTACTING TECHNICAL SUPPORT

For additional help, email support@hexiscyber.com or call +1 650 830-0484, Option 2. Also see the Hexis Cyber Solutions Technical Support web page at http://www.hexiscyber.com/content/

for HawkEye AP documentation, product downloads, and additional information on contacting support to escalate help on issues that impact your production environment.

CHAPTER 1

Introduction

This chapter introduces you to the components of a HawkEye AP system and of HawkEye AP, Console the web interface that facilitates several administration tasks.

This chapter contains these sections:

- "Overview of HawkEye AP System Administration", next
- "Overview of HawkEye AP Components and Processing", on page 22
- "EDW Architecture", on page 24

OVERVIEW OF HAWKEYE AP SYSTEM ADMINISTRATION

HawkEye AP system administrators work mostly in a Linux environment and should be comfortable using Linux in command-line mode. Administrators should also be familiar with SSH, managing users and groups, file system permissions and management, network configuration, logging, and using such text editors as VI or emacs.

If your HawkEye AP deployment collects log data from Windows machines, firewalls, routers, or applications, familiarity with those systems is also required.

HawkEye AP system administrators perform the following tasks:

- **Installation and configuration**—Installation and initial configuration of your HawkEye AP software is discussed in the *Installation, Configuration, and Upgrade Guide.*
- **Upgrading**—Upgrading your HawkEye AP deployment to a newer version is discussed in Example Upgrade Path in Chapter 4, "Upgrading HawkEye AP" in the *Installation, Configuration, and Upgrade Guide*.
- **Administration**—Administration tasks discussed in this manual include:
  - Configuring batch collection of event data—After basic installation, you configure HawkEye AP to collect batched event data by installing log adapters and configuring the Collector, Retrievers, and Loaders.
  - Configuring real-time modules—Optionally, you configure HawkEye AP real-time components such as Parsers to receive and process real-time streams of event data.
  - Managing the EDW data store—As you configure HawkEye AP, you load test data and run test queries against that data. You can also create tables, views, and column filters, manage users and permissions for data access, monitor performance, and archive or retire data.
  - Managing the EDW instance—You configure and manage clustered nodes in an EDW instance and configure nearline storage devices.
  - Manage Near-line storage—you can archive event data stored in the EDW to a variety of nearline storage devices such as EMC Centera, NetApp Snaplock, Hitachi HCAP, Fujitsu Eternus, and Remote NFS (Network) or CIFS (Common Internet) File System.
  - Manage access to HawkEye AP Console—Users open a browser and use HTTP or HTTPS to access HawkEye AP Console. You configure authentication and authorization for this access and provide users with the URL to access HawkEye AP Console.

- Backing up and restoring event data and raw log files.
- Managing users, permissions and authentication—Permissions for your HawkEye AP deployment and its users must be managed carefully to limit access to information and ensure a secure environment.
- Managing the report cache—You manage the retention of cached report data.
- Network configuration—You configure network connections between HawkEye AP components and the machines or devices from which you are collecting event data.
- Monitoring system performance and source health—Your HawkEye AP software provides tools to monitor the performance of the HawkEye AP deployment and event data sources. You configure these components to issue alerts when performance is diminished or event data collection is interrupted.

Overview of HawkEye AP Components and Processing

The HawkEye AP system supports the industry's most comprehensive real-time and historical event management capabilities. Its patented data model and compression technology facilitate online storage of massive volumes of data across an entire network. The HawkEye AP architecture enables rapid querying of and visibility into security threats.

Linear scalability through clustering provides the flexibility needed to support new applications and devices as requirements grow. Integration capabilities allow HawkEye AP to use external enterprise authentication authorities to validate and manage HawkEye AP users, and external nearline storage devices to archive HawkEye AP data. In addition, you can forward alerts to enterprise security management products, pagers, phones, and portals.

A HawkEye AP system comprises the following core components:

- **Event Data Warehouse (**EDW**)**—stores event data from multiple sources in a scalable and highly compressed format. Parallel processing enables clustered servers to execute as a single instance, allowing high-speed loading and querying on terabytes of data.
- **Analyzer—**is a set of services running on the head node of a HawkEye AP cluster. These services include:
  - **Application Manager**—manages reporting functions and communicates with HawkEye AP Console to display reports, alerts, and dashboards. Command-line utilities also communicate with the Application Manager to access stored data and configuration information. The Application Manager also manages report cache entries and schedules. The Application Manager is implemented as a set of services that run within a jBoss Application Server, which is installed and configured transparently during installation of HawkEye AP software.
  - **Open Access Extension (OAE)**—provides access to data stored in the EDW via standard database connectivity tools such as ODBC and JDBC.
  - **Postgres**—is a database that stores report cache entries, configuration data, and other run-time objects used internally by HawkEye AP. Postgres is installed and configured automatically when you install HawkEye AP software. Postgres also receives real-time alert messages from the Parser, and raises alerts to the Application Manager, which passes them to the HawkEye AP Console.
  - **LDAP**—provides authentication services for a HawkEye AP deployment.
- **HawkEye AP Console**—provides a rich client, system-independent user interface for monitoring, analyzing, resolving, and reporting real-time and historical event data. The console supports alert drill-down and reporting, ad-hoc querying, and scheduled reporting.

- **Collector**—pulls event data from disparate sources. It uses *retrievers* that define how the Collector obtains batched event data from event sources and *log adapters* that define how incoming batched data is parsed and stored in the EDW. The Collector uses a file system retriever to pull data that is stored in *syslog-ng*, the industry standard in receiving event data from internal and external sources.

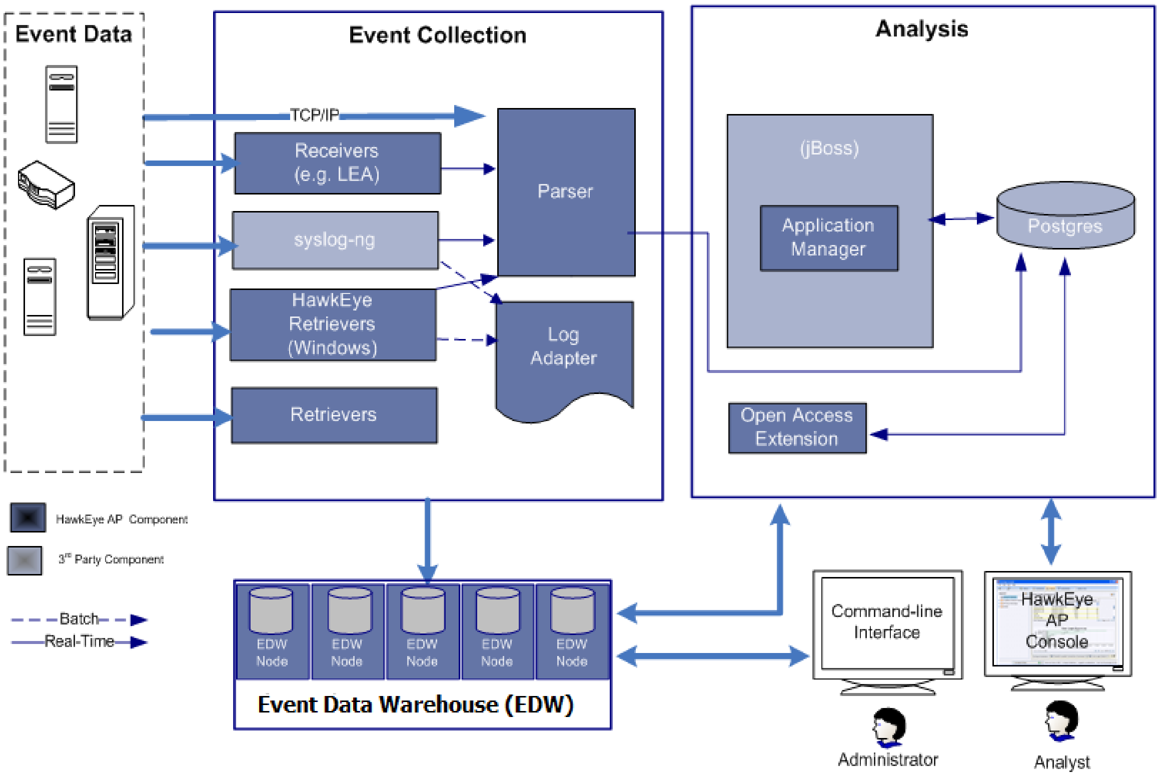
- **Parser** —processes streaming data from event sources in real time. It gets data over streams from syslog-ng, the HawkEye Retriever (for Windows), and optionally other TCP-based streams.

  Most installations include one Parser, though more than one can potentially be configured. Parsers use Parsing and Alerting rules written in HawkEye Event Processing Language (HEPL) to examine incoming data and send alerts when the data matches patterns defined in the rules. These rules can also raise alerts based on correlation of multiple events from multiple sources. Alerts appear in the Security Alerts widget of HawkEye AP Console.

- **Syslog-ng**—The industry standard in bifurcating data into the batch and real-time streaming modes.

Figure 1-1 illustrates the complete HawkEye AP architecture.

Figure 1-1: HawkEye AP Architecture and Data



As illustrated in Figure 1-1, events enter the HawkEye AP system from any of many external systems, such as network devices and software applications. Events enter in one way:

- **Real-time streaming**—events flow into the HawkEye AP system as a real-time stream from network devices and software applications that publish the events.

As raw event data enters the system, it typically flows directly to the Parser, although in some configurations, a Receiver receives the data and passes it to the Parser. The Parser uses the parser rule specific to the log source that recorded the event and assigns a unique ID to the data. The Parser can also use Alerting rules that correlate data from multiple sources. When incoming events match an Alerting rule's criteria, the Parser raises security alerts to the Application Manager. The Application Manager then:

- maps security alerts to predefined organization assets
- persists the security alert in its data store
- forwards the security alert to HawkEye AP Console and to specified notification recipients by email.

HawkEye AP Console displays the security alerts and enables reports on the data.

Analysts can access security alerts and reports through the HawkEye AP Console, which provides a powerful graphic interface.

To learn more about security alerts and viewing reports, see Viewing Security Alerts in Chapter 2, “Using Dashboards”“and Viewing Reports in Chapter 2, “Using Dashboards” in the *Reporting Guide*.

- Batched—Events are collected from log files and other event repositories maintained by network devices, operating systems, and software applications.

  The Collector polls a data source or repository to retrieve event data, which it loads into the EDW. The EDW makes the event data available to the Application Manager for report.

  Administration users can access EDW data either by using command-line interface (CLI) utilities on a Linux system or by viewing reports through the HawkEye AP Console.

For more information:

- To learn more about configuring retrievers and the Collector, see Chapter 3: Collector Configuration in the *Event Collection Guide*.
- To learn more about administering the Collector, see Chapter 7: Administering the Collector.
- To learn more about monitoring security alerts, see Chapter 10: Administering Assets and Monitoring Alerts.
- To learn more about monitoring enterprise alerts, see Viewing Security Alerts in Chapter 2, “Using Dashboards” in the *Reporting Guide*.
- For more information about the EDW and Sensage SQL, see:
  - Chapter 3: Loading, Querying, and Managing the EDW
  - Chapter 2: Configuring and Managing HawkEye AP
  - Sensage SQL in Chapter 10, “Sensage SQL” in the *Reporting Guide*
  - SQL Functions in Chapter 11, “SQL Functions” in the *Reporting Guide*

EDW Architecture

- “About the Event Data Warehouse”, next

- “Software for the EDW”, on page 25

About the Event Data Warehouse

The Event Data Warehouse (EDW) is a database built for and dedicated to loading, storing, and analyzing log data. The software uses sophisticated, application-level clustering to perform all load and query tasks fully in parallel across an arbitrary number of hosts. This architecture allows users to load and query massive data volumes in a single, logical database instance without partitioning. The EDW uses a proprietary data model that achieves high levels of compression, while still making all data fully available to query.

Software for the EDW

The EDW software is typically installed across five or more hosts, which make up an EDW instance. All log data is loaded, in parallel, across each host. This ensures that data loads at the fastest possible rate (given the number of hosts in your EDW instance) and provides for high availability and data redundancy.

The EDW instance is known as an *application-level cluster*; that is, any work unit received by one host must be executed across all hosts.

In essence, the EDW is deployed so you can:

1 Load log data into it.

2 Query log data out of it.

The EDW command `atload` performs the loading process in a batch mode from log data that is collected from some other source. To load the data, you need a PTL file that will draw out and reorganize the information from your log files into a format that the EDW is able to understand and load. Hexis Cyber Solutions can make a large number of log adapters available. Please contact your sales representative for details.

For redundancy, two copies of every piece of data is included across the file system—for example, in a five-host EDW instance, data from a single load may appear distributed in the following way:

Load01 -->	Primary -->	Load01a	Load01b	Load01c	Load01d	Load01e
	Secondary -->	Load01e	Load01a	Load01b	Load01c	Load01d

After you use atload to load data into the EDW, you can use the `atquery` command to retrieve data from this distributed system.

CHAPTER 2

Configuring and Managing HawkEye AP

This chapter provides the syntax of the HawkEye AP `clsetup` utility and shows how this utility is used to configure and manage HawkEye AP. Also included are utility tools for general tasks that support configuration and management.

- "General Syntax for the clsetup Utility", on page 28

  Describes command-line options used only in the `clsetup` utility.

- "Configuring a HawkEye AP Deployment with Clsetup", on page 34

  Describes how to configure a HawkEye AP deployment using the `clsetup` utility.

- "Managing HawkEye AP with Clsetup", on page 54

  Describes how to manage a HawkEye AP deployment using `clsetup`.

- "General Syntax for Utility Tools", on page 61

  Describes command-line options used in all the configuration utilities.

- "Using Utility Tools", on page 64

  Describes how to use utility tools such as `cldiff`, `clssh`, `clsync`, `cltop`, and `clhosts` to perform general task such as copying files and directories.

GENERAL SYNTAX FOR THE CLSETUP UTILITY

This section describes the syntax used with the clsetup utility and contains the following subsections:

- “General Syntax for clsetup”, next
- “Clsetup General Options”, on page 31

General Syntax for clsetup

The `clsetup` utility uses the following general syntax:

```
clsetup <general_options> <action> <component>
         <specific_options> [<instance_name>]
```

The left side of the `clsetup` command is generic to all components:

```
clsetup <general_options> <action>
```

where:

- *<general_options>*—any of the general options described in “Clsetup General Options”, on page 31 and “General Syntax for Utility Tools”, on page 61
- *<action>*—one of several commands, such as `configure`, `add`, or `start`
- *<component>*—one of the HawkEye AP components (EDW, Real-Time, and Collector)
- *<specific_options>*—any of the options specific to the command and the EDW component.
- *<instance_name>*—the instance name of the EDW component

Note that the remainder of the command line is specific to the component being configured or managed. For example, use the following syntax when configuring the EDW:

```
clsetup [<general_options>] configure sls <mandatory_fields> \
    [<specific_options>]
```

where *<mandatory_fields>* are any of the mandatory fields specific to the command and the EDW component. In addition to specific mandatory fields, however, there are a few shared ones. For more information, see “Shared Mandatory Fields with Default Values”, on page 64.

For information on how to specify a value for each option, see the following sections:

- “Options Default Values”, on page 62
- “Getting Help”, on page 62

Specifying Values for Mandatory Fields

Some mandatory fields are specific to a `clsetup` command. You must specify a value for most of these fields. For example, the following command creates an EDW instance. It includes all mandatory fields for which you must specify a value:

```
clsetup add sls MyNewInstance \
    --hosts host01,host02,host03 \
    --short-desc "short meaningful text" \
    --description "longer meaningful text"
```

Some mandatory fields do not require a value because the system provides a default value. You only specify these fields on the command line to change the default value. For example, the command to create an EDW instance includes two additional mandatory fields (`dsrootdir` and `sls-tempdir-space`), both of which have default values.

- Specify a value for `dsrootdir` only to use a non-default location of the `dsroot` directory for the instance. The default location is:

  ```
  <Sensage_Home>/latest/data/sls/instance/<instance_name>/dsroot
  ```

- Specify a value for `sls-tempdir-space` only to use fewer gigabytes than the minimum size specified by the default. The default size is `16`GB.

Each of the sections below documents the mandatory fields specific to each command.

Some mandatory fields are shared by all `clsetup` commands. You do not need to specify a value for these mandatory fields unless the system-provided default values are unsuitable for your system. For more information, see “Shared Mandatory Fields with Default Values”, on page 64.

Specifying Mandatory-Field Values on the Command Line

When you use `clsetup` to add an EDW instance or a Real-Time module or to configure or reconfigure the EDW or the other, Non-EDW components, you do not need to specify any fields on the command line. If you run the command without specifying any fields or without specifying all mandatory fields, the configuration file opens in your default editor. You can use the editor to specify values for all fields.

If you prefer to add or configure the EDW or HawkEye AP component options entirely from the command-line, you must specify a value for every mandatory field that does not have a default value. You should also specify values for specific options of the command.

From the command line, you can specify a value for any field in the configuration file that is prefixed by the double dash (`--`). If you specify values for all mandatory fields, the form does not open in the editor. If you do not specify values for all mandatory fields, the form opens in the editor, and contains the values you specified from the command line as well as default values.

NOTE: If you want to use `clsetup` within a script, include the `--noedit` option. This option keeps the script from opening the configuration file in an editor under any circumstances. When you include this flag, invalid or missing values cause a fatal error.

Running clsetup Over an EDW Instance

You must have SSH trust enabled for the root user on all hosts in the EDW instance in order for the `clsetup` command to run successfully.

Hexis Cyber Solutions recommends that, when you run `clsetup` as root over an EDW instance, use "`sudo clsetup`" with root trust enabled among all hosts. For more information, see Setting up the HawkEye AP System User and Group in Chapter 1, “Installing HawkEye AP” in the *Installation, Configuration, and Upgrade Guide*.

You can invoke the `clsetup` command in two ways:

- **as root**—this requires you to know the root user’s password:

  ```
  su -
  clsetup ...
  ```

- **as yourself**—this requires you to be enabled for sudo, but you need know only your own password:

  ```
  sudo clsetup ...
  ```

  For more information, see Root Privilege and the sudo Command in Chapter 2, “Configuring HawkEye AP” in the *Installation, Configuration, and Upgrade Guide*.

Specifying Hosts

When you install HawkEye AP, specify every host that will be part of you HawkEye AP deployment. When you configure the EDW and the other, non-EDW components, you can specify all the hosts in the deployment; this creates a unified deployment, which is a simpler kind of deployment to set up and administer. Alternatively, you can configure the EDW for one subset of hosts in the deployment and configure a different subset of hosts for the non-EDW components; this creates a divided deployment, which is a more complex kind deployment to set up and administer.

For more information, see Knowing Your Hosts in Chapter 1, “Installing HawkEye AP” in the *Installation, Configuration, and Upgrade Guide*.

- If you configure all the hosts to create a unified deployment:
  - When you configure the EDW, specify all of the hosts in the HawkEye AP deployment.
  - When you add an EDW instance, specify the subset of hosts allocated to the EDW component.
  - When you configure the remaining components:
    - *specify all of the hosts in the HawkEye AP deployment.*
    - *set the* `sls-host` *flag to specify one of the EDW hosts that stores the data your components will access; configure its port in the* `sls-port` *flag.*
- If you configure subsets of the host to create divided deployment:
  - When you configure the EDW, specify the subset of hosts allocated to the EDW component.
  - When you add an EDW instance, specify the subset of hosts allocated to the EDW component.
  - When you configure the remaining components:
    - *specify only the host NOT specified as an EDW host*
    - *set the* `sls-host` *flag to specify one of the EDW hosts that stores the data your components will access; configure its port in the* `sls-port` *flag.*

- ◆ *set the* `remote-sls` *flag to* `"yes"`

NOTE: You can specify hosts in the following formats:

- a comma-separated list: `host1.foo.com,host2.foo.com,host4.foo.com`
- a collection of hosts with the same prefix: `host{1,2,4}.foo.com` or `{host1,myhost2}.foo.com`
- a range that uses exclamation mark (`!`) to exclude a specific host: `host{1-4,!3}.foo.com`

Hexis Cyber Solutions recommends that, when you run `clsetup` as root over an EDW instance, you use `"sudo clsetup"` with root trust enabled among all hosts. For more information, see Setting Up SSH Trust in Chapter 1, "Installing HawkEye AP" and Root Privilege and the sudo Command in Chapter 2, "Configuring HawkEye AP" in the *Installation, Configuration, and Upgrade Guide*.

Clsetup General Options

The `Clsetup` utility has options to configure and manage Hexis Cyber Solutions components. For information about the general options available to all cluster utility tools see "General Syntax for Utility Tools", on page 61.

Getting Help

There are several options that display help information.

- The `--help` option with arguments provides an abbreviated usage description of the arguments; for more information, see "Abbreviated Usage Description on Specific Arguments", next.
- The `--help` option with no arguments provides detailed usage description; for more information, see "Detailed Usage Description of All Arguments", on page 32.
- The `help` option with arguments provides the complete set of options for those arguments in the form of a syntax diagram; for more information, see "Full Set of Options on Specific Arguments: Syntax Diagram", on page 32.
- The `help` option with no arguments provides the complete set of options for the `clsetup` utility in the form of a syntax diagram; for more information, see "Full Syntax for Help on EDW Component", on page 33.

Abbreviated Usage Description on Specific Arguments

Use the syntax below to display abbreviated `clsetup` syntax options for only the specified command on the specified Hexis Cyber Solutions component.

```
--help <command> <component>
```

The following examples illustrate how to get help on starts the EDW and stopping the Real-Time component:

```
clsetup --help start sls
```

```
clsetup --help stop rt
```

The examples below illustrate the output of the commands above.

EDW Output

```
Usage:

clsetup <gen_options> start sls <spec_options> <instance_ref>

 Start specified instance.

Arguments:
  instance_ref
    Reference to an SLS instance (either by name,
    or in the form host:name or host:port).
    Value can also be set via the "SLS_INSTANCE" environment variable.
```

Real-Time Component Output

```
Usage:

clsetup <gen_options> stop rt <spec_options> <module_type> <module_name>

 Stop specified instance.

Arguments:
  component_type
    Start components of specified type only
  component_name
    Start only specified components

Options:
  --host
    Start components running on specified host.
```

Detailed Usage Description of All Arguments

Use the syntax below to display full `clsetup` syntax options for all commands on all components.

```
--help
```

The following example illustrates how to get help on all clsetup options:

```
clsetup --help
```

Full Set of Options on Specific Arguments: Syntax Diagram

Use the syntax below to display all `clsetup` syntax options for the specified command on the specified component in the form of a syntax diagram

```
help <command> <component>
```

The following example illustrates how to display the full syntax for starting the EDW:

```
clsetup help start sls
```

The output for the above command is:

```
Usage:

clsetup [--[no]help] [--[no]longhelp] [--config=<valid_path>] [--[no]edit]
  [--editor=<valid_command>] [--[no]debug] [--[no]parse-only]
  [--[no]preview] [--[no]relax] [--[no]progress] [--[no]remote-preview]
  [--verbose=<noise-level>] [--[no]version] [--[no]welcome] [--user=<name>]
  [--[no]root] [--[no]serialize] [--[no]sudo] [--timeout=<int>]
  [--login-timeout=<int>] start sls <instance_ref>
```

Full Syntax for Help on EDW Component

Use the syntax below to display abbreviated syntax for the full set of `clsetup` options.

```
help
```

The following example illustrates how to get abbreviated help on all clsetup options:

```
clsetup help
```

Clsetup Options with True or False Values

When you request command-line help, some options have a `true` or `false` value that they display as the number `"1"` within parentheses. For example, when you request help on a utility, the edit option displays as:

```
--edit (1)
```

In this case, the number `"1"` indicates that the option defaults to `"true"`. This chapter documents options that default to `true` differently from the output of command-line help. For example, the edit flag is documented as:

```
--[no]edit
```

This option is explained in the documentation as:

> Invokes editor to enter missing or incorrect parameters; default is `--edit`. If you specify `--noedit`, does not open the editor when parameters are missing or incorrect.

Clsetup Execution Options

Control of Execution

```
--relax
```

> Relaxes various checks, transforming some fatal errors to warnings.

Remote Execution

NOTE: The recommended way to run `clsetup` as root over an EDW instance is to use "`sudo clsetup`" with root trust enabled among all hosts. For more information, see Setting Up SSH Trust in Chapter 1, “Installing HawkEye AP” in the *Installation, Configuration, and Upgrade Guide*.

`--user`

Uses the specified userid for remote execution

`--root`

Uses root for remote execution

Aliases

```
edit = "--edit add"
enable = "add"
reconfigure = "--edit configure"
prepare = "configure"
```

Invokes editor to enter missing or incorrect parameters; default is --edit. If you specify --noedit, does not open the editor when parameters are missing or incorrect.

Configuring a HawkEye AP Deployment with Clsetup

This section describes how to configure a HawkEye AP deployment: using the `clsetup` utility.

Adding, Editing, or Upgrading an EDW Instance (clsetup)

To add a new EDW instance, edit the configuration or upgrade the software of an existing instance, use the following syntax:

```
clsetup [<general_options>] { add | edit } sls <mandatory_fields>
    [<specific_options>] <instance_name>
```

IMPORTANT:

- After you install and test a new version of HawkEye APon your system, run `clsetup add sls --upgrade` on an existing instance to upgrade it to the new version. For more information, see Example Upgrade Path in Chapter 4, “Upgrading HawkEye AP” in the *Installation, Configuration, and Upgrade Guide*.

- If you install a new version of HawkEye AP to your deployment and want to edit the configuration of an existing instance without upgrading the instance, run the previous version of the `clsetup` command by specifying its full path. Do not specify the `--upgrade` flag. Version-specific clsetup commands are located in:

```
<Sensage_Home>/<version>/bin
```

- If you have only one version of HawkEye AP on your system and you want to edit the configuration of an existing instance instead of upgrading it, run `clsetup` with the `edit` argument rather than the `add` argument.

NOTE: See also: “Starting, Stopping and Restarting an EDW Instance (clsetup)”, on page 54

Arguments

`<instance_name>`

The instance name is a required argument that provides the name of the EDW instance.

For example: *`my_instance`*

NOTE: Use the <*instance_name*> argument only for instances that reside on the same host from which you run the command. If you specify only one instance reference, the command includes all hosts that are part of that instance.

As a convenience, you can omit this argument if you set the environment variable `SLS_INSTANCE` in the session where you run the clsetup command.

Mandatory Fields

These fields are mandatory when you add an instance. When you edit or upgrade an instance, they are optional and change the current values for the existing instance.

`--short-desc`

A single-line description of this instance.

`--description`

Detailed description of this instance.

Specific Options

`--owner-name`

Owner of this instance (that is, the administrator).

`--owner-email`

Email address; no required formatting rules.

`--owner-contacts`

Phone number; no required formatting rules.

`--backwards-compatible`

Setting this option allows you to run version newer versions of the EDW software in an environment where the new authentication features are not desired. This flag sets the **guest** user to have full access to the EDW, emulating the behavior of pre-3.0 versions of the EDW.

`--hosts`

Hosts to include in the EDW instance.

`--port`

Port used to communicate with the EDW instance.

`--sls-user`

Owner of processes and files in the EDW instance.

`--dsrootdir`

Location of data store. The default and recommended value is:

`<Sensage_Home>/latest/data`

`--sls-tempdir`

Location of a large workspace to be used for storing temporary data. The default is:

`<Sensage_Home>/latest/etc/sls/instance/<instance_name>/temp`

`--upgrade`

Upgrades the software of the instance to the newest HawkEye AP version in your deployment. For more information, see Example Upgrade Path in Chapter 4, “Upgrading HawkEye AP” in the *Installation, Configuration, and Upgrade Guide*.

`--read-only`

Prohibits loads and deletes.

`--couple-to`

Links the data store of the current instance to that of another instance. This option may be used to configure a hierarchical installation to avoid issues with VPNs and remote offices.

`--ldap-couple-to=<instance_name>`

Uses and shares access permissions with the specified instance, which must exist and have valid LDAP information (see following note). Currently, you cannot chain `--ldap-couple-to` (that is, you cannot couple to a coupled instance). Any access permission changes made to one instance immediately affect the other instance.

NOTE: The `"clsetup add sls"` command verifies that an LDAP server is running and has at least the root LDAP entries. The command also loads instance-specific access permission entries unless they are already there, in which case the existing entries are considered authoritative (no attempt is made to merge).

`--sls-tempdir-space=<n>`

Specifies minimum space requirement in gigabytes. The default is `16` GB.

`--normal-runqueue-quota=<n>`

`S`pecifies the maximum number of concurrent tasks. Any additional requests are queued and not executed until one of the other tasks completes. The default value is `3`.

`--critical-runqueue-quota=<n>`

Specifies the maximum number of concurrent tasks. Any additional requests are queued and not executed until one of the other tasks completes. The default value is `10`.

`--expire-date`

Reserved for future use

`--max-space`

Reserved for future use

Aliases

`clsetup edit sls`

Alias for `clsetup add sls` when the instance already exists

Configuring or Reconfiguring an EDW Instance (clsetup)

To configure or reconfigure an EDW instance, use the following syntax:

```
clsetup [<general_options>] { configure | reconfigure } sls
    <mandatory_fields> [<specific_options>]
```

Mandatory fields

`--sls-user=<username>`

Specifies the owner of the EDW instance processes and files. Typically, this is the HawkEye AP system user. The user must exist on all host in the EDW instance and belong to the group you specify in the `--sls-group` flag.

`--sls-group=<group_name>`

Specifies the group owner of all EDW instance processes and files. Typically, is the HawkEye AP system user.

Mandatory Fields With Acceptable Default Values

```
--gnu-dir
--rsync-dir
--ssh-dir
```

For information about these fields, see“Shared Mandatory Fields with Default Values”, on page 64.

Specific Options

`--hosts`

List of hosts that are configured for the EDW component and may be included as hosts in EDW instances.

`--data-root`

Defines the default location for large data stores. Various components place their data directories below the specified location.

`--ldap-instances`

Specifies a list of LDAP servers to which the EDW component can connect. Hexis Cyber Solutions recommends that these be different machines from the ones used by the EDW, but preferably on the same network.

`--ldap-organization`

Specifies the name of the organization that uses the EDW. The name defines the root of the LDAP tree and is used to generate certificates. If not specified, it will be extracted from the fully qualified domain name of the hosts allocated to the EDW component.

`--[no]ldap-use-tls`

Specifies that communications with the LDAP server be encrypted (using transport layer security). If set to `--noldap-use-tls`, specifies that communications with the LDAP server *not* be encrypted. Defaults to `--ldap-use-tls`.

`--nss-cachedir`

Specifies the area where the nearline storage server keeps files retrieved from the storage device. For more information, see Chapter 9: Archiving to Nearline Storage.

`--nss-cachefiles`

Specifies the cache file limit, the maximum number of files in the NSS cache and is particularly important on highly fragmented systems. It prevents excessive creation of cache files. The value of this parameter defaults to `10000`.

--nss-cachesize

The nearline FIFO should be two times the thread pool size + 1. The default value is `10000000000`. Contact your Hexis Cyber Solutions representative for assistance in determining the appropriate cache size. For more information, see Chapter 9: Archiving to Nearline Storage.

```
--nss-threadpoolsize
```

Specifies the number of threads that represent the number of concurrent actions performed on the nearline storage device at any one time. The value of this parameter depends on the size of the nearline storage device. The default value is `12`. Contact your Hexis Cyber Solutions representative for assistance in determining the appropriate thread pool size. For more information, see Chapter 9: Archiving to Nearline Storage.

Depends on the size of the Centera. The number of threads represent the number of concurrent actions performed on the Centera at any one time. Contact your Hexis Cyber Solutions representative for assistance in determining the appropriate thread pool size. For more information, see Chapter 9: Archiving to Nearline Storage.

```
--nss-<NLS_device>-blobsize
```

This parameter allows you to change the blobsize of <*NLS_device*>.

The value of this parameter defaults to `100000000` for all nearline storage devices except remote file systems. For a remote file system, the value of this parameter defaults to `2000000000`. Typically there is no need to change the default value of this parameter.

Available options for this parameter are:

```
--nss-centera-blobsize
--nss-snaplock-blobsize
--nss-hcap-blobsize
--nss-eternus-blobsize
--nss-directory-blobsize.
```

For more information, see Chapter 9: Archiving to Nearline Storage

```
--nss-diskreadbuffer
```

Typically there is no need to change the default value of this parameter. The default value is `1000000`.

Aliases

```
--edit configure = reconfigure
```

Configuring the Application Manager and the Real-time Component(clsetup)

To configure the Application Manager, the Collector and the Parser, use the following syntax:

```
clsetup [<general_options>] configure sensage
    <mandatory_fields> [<specific_options>]
```

NOTE: Running this command starts `atpgsql` (the Application Manager's data store) as well as the Application Manager.

NOTE: See also: “Starting, Stopping and Restarting the Real-Time Modules (clsetup)”, on page 59 and “Listing all Real-Time Modules (clsetup)”, on page 58.

Mandatory Fields

`--appserver-host—`

The `--appserver-host` option specifies the host where the Application Manager runs. Specify a host in your HawkEye AP deployment that is not allocated to the EDW component, unless you are setting up a single-host deployment. For more information, see Knowing Your Hosts in Chapter 1, “Installing HawkEye AP” in the *Installation, Configuration, and Upgrade Guide*.

`--appserver-port—`

The port number used by Application Manager to send and receive requests. The default value is 443.

`--collector-hosts`

Specifies all host(s) on which the Collector runs

`--collector-datadir`

Specifies the root directory for all volatile files in the Collector, including temporary log files and state files. The directory you specify must not exist on an NFS mounted file system. The default directory is:

```
<Sensage_Home>/latest/data/collector
```

For more information, see Root Directories Used by the Collector in Chapter 3, “Collector Configuration” in the *Event Collection Guide*.

`--data-root={"[default]"|<path>}`

This is the data root directory used by all HawkEye AP components. Each component has its own subdirectory under this root directory.

NOTE: The `"[default]"` value instructs the system to determine the path automatically. Change this value only to specify your own path.

`--mail-host`

The STMP server.

`--mail-user`

Username to be used for connecting to mail host; defaults to `lms`.

`--mail-protocol`

Mail protocol to be used; defaults to `SMTP`.

`--notify-from`

From-header for all notifications.

`--notify-low`

Destination for low-priority notifications.

`--notify-medium`

Destination for medium-priority notifications.

`--notify-urgent`

Destination for urgent notifications.

`sensage-user=<user>`

Owner of instance processes and files. Generally, you specify the HawkEye AP system user. This user must exist on all hosts in the HawkEye AP deployment, and the user must belong to the group you specify in the `--sensage-group` flag. =

`--sensage-group=<group>`

Group owner of all instance processes and files. Generally, you specify the HawkEye AP system group. The user you specify with --sensage-user must be a member of this group.

`--hosts`

List of hosts to be used by the Real-Time component and its modules.

`--data-root={"[default]"|<path>}`

Default location for large data stores.

NOTE: The `"[default]"` value instructs the system to determine the path automatically. Change this value only to specify your own path.

`--controller-proxy-host`

Host used by client to access the Application Manager; defaults to the value set for `--appserver-host`

`--controller-tempdir=<path>`

Temporary directory to hold records rejected during EDW upload; defaults to:

```
<SenSage_Home>/latest/data/controller/temp
```

`--controller-ip={"[default]"|<ip-address>}`

In an Application Manager host with multiple Network Interface Cards (NICs), this is the IP address of the NIC for the Application Manager that you want bound to for communication with the Parser and receivers. Without this option, the Application Manager binds to all NICs in the Application Manager host.

For example, assume you run the Application Manager on a host with two NICs: one attached to a publicly accessible network, and one attached to a private network where the Parser and receivers run. Use the `--controller-ip` option to ensure that the Application Manager binds to and communicates only over the private network.

NOTE: The "[default]" value instructs the Application Manager to bind to all NICs in the Application Manager host. Specify an IP address only to select which single NIC the Application Manager uses for communication with other Real-Time modules.

`--controller-max-alerts=<n>`

Maximum number of alerts that the Application Manager retains for display in HawkEye AP Console; defaults to `300`

`--controller-alert-retention-days=<n>`

Number of days that the Application Manager retains alerts for display in HawkEye AP Console; defaults to `8`

`--sls-host`

A host of the EDW (SLS) instance that stores the data your components will access; you must specify one of the hosts on which you added an EDW instance.

`--sls-port`

Port used by target EDW instance.

`--remote-sls`

When you configured the EDW, if you specified only the hosts on which the EDW instance runs and not those additional host(s) required to run non-EDW components, set the `remote-sls` flag to `"yes"` and run `clsetup` to specify the host(s) for the non-EDW components. In addition to specifying the non-EDW hosts, you must specify one of the existing EDW hosts and the port of the existing EDW host. For example, if you configured the EDW to run on host1, host2, and host3 and now you want to configure the non-EDW components to run on host4, you would run a command like the following:

```
clsetup configure sensage --hosts=host4 --remote-sls=yes \
    --sls-host=host1 --sls-port=8072
```

`--sls-user`

EDW user name to use for loads into target EDW.

`--sls-default-namespace`

Specifies the EDW namespace to use to load unidentified events (events that fail to parse or for which there is no PTL file); the EDW uses the `catchall.ptl` to process these events and loads them in the root namespace, which is indicated by empty quotation marks (`""`).

`--remote-db`

The Application Manager uses the open-source database Postgres to persist Parser Rules and the alerts themselves. If you do not already have a Postgres database installed and running, keep `remote-db` set to `no` and the configuration utility will create the database for you. In this case, the value you specify for `db-host` must be one of the hosts specified in `hosts`. If you do have a Postgres database installed and running, change `remote-db` to `yes`. In this case, the value you specify for `db-host` can be different from the values specified for `hosts`. For more information, see --db-host below.

`--[no]db-reload`

If set to `--db-reload` and you already have a Postgres database installed and running, causes the existing database to be dropped and reloaded. If set to `--nodb-reload`, does *not* cause the existing database to be dropped and reloaded. Defaults to `--nodb-reload`.

`--db-host`

The host on which to run the Application Manager's persistent data store.

`--db-port`

The port used by the Application Manager's persistent data store.

`--db-name`

The name of the Application Manager's persistent data store.

`--db-root-user`

The name of the root user of the Application Manager's persistent data store; defaults to `lms`.

`--db-root-password`

The password of the root user of the Application Manager's persistent data store; no default value.

`--db-user`

The name of the user of the Application Manager's persistent data store; defaults to `controller`.

`--db-password`

The password of the user of the Application Manager's persistent data store; defaults to `controller`.

`--db-datadir`

The data directory for the Application Manager's persistent data store; defaults to: `<`*`Sensage_Home`*`>/latest/data/atpgsql`.

`--db-logdir`

The log directory for the Application Manager's persistent data store; defaults to: `<`*`Sensage_Home`*`>/latest/var/log/atpgsql`.

NOTE: As of HawkEye AP version 4.x, the 5 `--sls-upload` parameters described below apply only to uploading of real-time alerts to the EDW.

```
--sls-upload-start-time=<hh.mm.ss>
```

Start time for upload of real-time alerts; in hh:mm:ss format; defaults to 23:00:00

```
--sls-upload-timezone=<time_zone>
```

Time zone for --sls-upload-start-time; defaults to GMT

```
--sls-upload-duration=<minutes>
```

Upload duration for real-time alerts in minutes; defaults to 180

```
--sls-event-upload-interval=<minutes>
```

Event upload repetition interval for real-time alerts in hours; defaults to 60

Mandatory Fields With Acceptable Default Values

```
--gnu-dir
--java-dir
--openssl-dir
--rsync-dir
--ssh-dir
```

For information about these fields, see “Shared Mandatory Fields with Default Values”, on page 64.

Specific Options

```
--appserver-is-encrypted—
```

If set to “true” or not specified, the Console Manager will use SSL to encrypt communications and users must specify the https protocol in the URL used to access the HawkEye AP Welcome page. If set to “false”, the Console Manager will not encrypt communications. By default, this option is set to "true" and the appserver module is configured for encrypted communication.

```
--appserver-internal-port
```

The value for this port is normally assigned automatically. Edit this option only if the port you specify with --`appserver-port` conflicts with another use of that port number.

Adding or Editing a Parser Real-time Module and the Receivers (clsetup rt)

The Real-Time component consists of the Parser and one receiver. You must add the Parser and receivers separately:

- **Parser** —parses source log entries from incoming event data according to parsing rules specific to their log sources, and raises alerts to the Application Manager via the Postgres database. See “Adding or Editing the Parser Real-Time Module”, next

- **Receivers**—obtain source log entries from LEA log sources,. See “Adding or Editing Receivers”, on page 49.

Adding or Editing the Parser Real-Time Module

To add a new Parser or edit an existing one, use the following syntax:

```
clsetup <general_options> { add | edit } rt --host=<host_name>
    [--options "<module_options>"] [--datadir=/<fullpath>/<directory>]
    [--datadir-size=<size>] parser [<module_name>]
```

IMPORTANT:

You must run this command as root. After the Parser starts and it has completed its privileged operations, Hexis Cyber Solutions drops the root privilege and uses the user privilege specified during configuration in the *sensage-user* parameter, which is described in “sensage-user=<user>”, on page 41.

- The `clsetup add rt parser [<parser_name>]` command fails if the named parser already exists. To reconfigure an existing module, specify the `clsetup edit rt` command, or run the `clsetup add rt parser [<parser_name>]` command with the `--edit` flag.
- If the Parser fails with the following error message, it cannot connect to the Postgres database:

```
Aug 14 12:09:19 ### Sensage: 1|2012-08-
14T19:09:19.205052Z|18964|18971|MyParser|DBConnPQ.cpp|69|DATABASE|VISIBLE-
ERROR|ACTION:EXCEPTION;MSG:DBConnPQ failure host=<host_name> port=5432
dbname=controller user=controller password=controller sslmode=prefer: could
not connect to server: Connection refused\\n  Is the server running on host
"<host_name>" and accepting\\n        TCP/IP connections on port 5432?\\n;
```

Verify the following: Postgres is running, the port configurations and login credentials are correct, an iptables allows traffic on the port

See also: “Starting, Stopping and Restarting the Real-Time Modules (clsetup)”, on page 59 and “Listing all Real-Time Modules (clsetup)”, on page 58.

Arguments

parser

The real-time module is `Parser`.

NOTE: Adding Collector, appserver, and Application Manager is not supported.

`<parser_name>`

The parser name must be unique throughout your HawkEye AP deployment. If you do not specify a name, the name defaults to parser.

Mandatory Fields

`--host`

The host where you want the Parser to run. Specify a host in your HawkEye AP deployment that is not allocated to the EDW component.

Parser Options

You can specify Parser-specific options with `--options="<module_options>"`. Enclose the entire set of options you specify in quotation marks (`""`). Use spaces to separate multiple options within the quotation marks.

`-C=<cache_size>`

This option controls the operation of the `dns` function in Parser rules. It specifies the maximum number of entries in the hosts look-up cache. The value you specify sets an approximate upper bound on the amount of memory consumed by the cache. The upper bound is imprecise because the size of each entry varies depending on the length of the host name. The fixed portion of an entry is 32 bytes.

If the Parser resolves a host name through DNS (Domain Name System) and the host's look-up cache is full, the Parser makes room for a new entry by discarding an expired entry. If no expired entries can be found, the Parser discards the oldest valid entry.

`-e=<expiry_time>`

This option controls the operation of the `dns` function in Parser rules. It specifies the length of time in seconds after which entries in the hosts look-up cache are invalid. The Parser resolves the host name to the IP address in the entry, as long as the entry has not expired. If the entry has expired, the Parser discards it, resolves the host name through DNS, and creates a new entry.

The default is 5400, or the equivalent of 90 minutes.

`-f=<path_and_filename>`

This option controls the operation of the `dnstoint` function in Parser rules. It specifies the location and name of a standard hosts file that you or someone in your organization maintains. The hosts file must reside on the HawkEye AP host where the Parser runs. If you specify a relative path, the root is:

```
<Sensage_Home>/latest/etc/rt/parser
```

When the Parser starts up and the -f option is specified, it loads entries from the hosts file into the hosts look-up cache. Entries in the cache that were loaded from the hosts file never expire. If you replace the hosts file on disk with a new version, the Parser discards entries in the cache that were loaded from the hosts file and loads entries into the cache from the new file version. The Parser uses the file modification date and time to determine if the hosts file has changed.

`-I=<polling_interval>`

This option controls the operation of the `dns` function in Parser rules. It specifies the length of time in seconds at which the Parser checks the hosts file, which you specify with the `-f` option. If the modification date and time on the hosts file has changed, the Parser refreshes the hosts look-up cache with entries from the new file version.

The default is 300, or the equivalent of 5 minutes.

`-parser-id=<N>`

This option controls the manually setting of the GUID (global unique identifier) used to track one-to-one association of EDW data, postgres data, and trigger data. Note that the GUID is a combination of the parsed microsecond timestamp and the 64-bit sequence number, which is composed of the 52-bit sequence and the 12-bit parser ID in the high bits, incremented for each new event

Whenever the parser is restarted, the 52-bit sequence number (uidSeq) is incremented by 10,0000; using this options allows you to override the automated setting for the GUID's Parser ID.

The default is for the Parser ID is 0.

For more details on using GUIDs, see "Using GUIDs in an Alerting Rule", on page 62 in the *HawkEye Event Processing Language Developers Guide.*

`--state-size=<size in gigabytes>`

Specifies the amount of memory allocated to store the data in state tables. Set --state-size within the --options parameter.

`--port=<num>`

Specifies the port the Parser listens on. The default is 5014.

The following option is relevant to using the trickle feed option.

`--atload-path`

Path of atload command [/opt/sensage/latest/bin/atload_perl].

The following six optional parameters are invoked by rules that call the LAL sendRow function. They can be used to adjust the timing and maximum size of trickle feed loads. The first parameter to be met for a given table triggers a load. Attempts to queue data for EDW loading are stopped when these disk limits are exceeded.

For information on the trickle feed option, see Chapter 3: Loading, Querying, and Managing the EDW, in this guide. For information on the sendRow function and Trigger Alerting Functions, see Trigger Alerting Functions, in Chapter 3, "Writing Alerting Rules with HawkEye Event Processing Language, in the *HawkEye Event Processing Language Developers Guide*.

`--load-bytes`

Specifies the number of bytes to collect before loading to the EDW. The default is 1gb.

`--load-minutes`

Specifies the number of minutes to wait before loading to the EDW. The default is 10 minutes.

`--load-rows`

Specifies the number of rows to collect before loading to the EDW. The default is 250k

`--load-retry-minutes`

Specifies the number of minutes to wait before retrying a failed load. The default is 60 minutes

```
--min-disk
```

Specifies how much space is required. The default is 5 percent.

```
--min-inode
```

NOTE: The compact parameters, when specified, have the following attributes that result in a compact operation when the value for the given parameter is reached.

- threshold-megs— Number of loads after which a compact operation is triggered (default - no trigger on this).
- threshold-rows— Numbers of rows after which a compact operation is triggered (default - no trigger on this).
- threshold-loads— Number of megabytes after which a compact operation is triggered (default - no trigger on this)
- compact-hours— Compact hours after midnight that triggers the compact operation (default - a list containing the number 0 (midnight hour)

If none of the threshold options are chosen, the Collector performs compaction according to the compact-hours option defined in the parameter.

Defaults for the following four --sls-<param> parameters are set automatically at sysconfig/sengage but can be overridden.

```
--sls-host
```

EDW host.

```
--sls-password
```

EDW instance password.

```
--sls-secret
```

EDW instance secret.

```
--sls-user
```

EDW instance user.

```
--var
```

Directory where the parser data is queued for EDW loads.

Example

To add a Parser named `ParserPrimary` on the localhost, run the following command:

```
clsetup add rt --host=localhost Parser ParserPrimary
```

Tuning Uploads: Parser EDW Data Queueing and Disk Space Limitations

If the Parser determines that disk space is approaching the reserve limit, it begins raising system alerts to the HawkEye AP Console. The Parser also raises alerts when EDW loads fail -- the most likely cause being low disk space. If the reserve limit is reached, the Parser can no longer queue data for upload to the EDW and data is lost. Hexis Cyber Solutions recommends configuring generous extra disk space to allow for spikes in uploads. The following events can cause sudden increases in data load:

- A Distributed Denial of Service (DDOS) attack or other unexpectedly high event volume.
- High EDW activity resulting in slowed loads.
- Loss of an EDW node.
- Maintenance activities such as adding a node.

The log file is located in the standard system log at:

```
/var/log/messages
```

The log message states one of the following:

- `Out of room in queue, incoming data throttled`
- `Asking Parser to close blocks early`

Adding or Editing Receivers

Generally, some of receiver’s configuration parameters are set automatically based on the type of receiver. Other parameters must be set explicitly when you add the receiver.

This section includes the following topics:

- “Syntax”, next
- “Arguments”, on page 50
- “Mandatory Options”, on page 51
- “Shared Receiver Options”, on page 50
- “LEA Receiver Options”, on page 51

Syntax

To add a new receiver, run the configuration utility with the following syntax, which is shown for the LEA Receiver:

```
clsetup add rt--host=<host_name> --options="--file=
<Sensage_Home>/latest/data/rt/<LEA_receiver_name>.state.txt
--stateDB=<Sensage_Home>/latest/data/rt/<LEA_receiver_name>.state
--sic=OPSEC_SSLCA --opsecConfig=
<Sensage_Home>/latest/data/rt/opsec.conf --mode=UNIFIED"
leareceiver <LEA_receiver_name>/
```

You must run this command as root. After the receiver starts and completes its privileged operations, HawkEye AP reverts from root privilege to the user privilege specified during configuration with the `sensage-user` parameter.

IMPORTANT: If you use the `clsetup` command to edit the options for an existing receiver, the command stops and starts the receiver so the new option values can take effect.

Arguments

`<receiver_type>receiver`

Specify the type of receiver you want to add as in **leareceiver**. For more information on these types of receivers, see Chapter 2: Configuring Parsers and Receivers in the *Event Collection Guide*.

`<receiver_name>`

The name of the receiver. The name defaults to the type of receiver. The name must be unique throughout your HawkEye AP deployment.

Mandatory Fields

`--host <host_name>`

The name of the host where the receiver should run. Specify a host in the deployment that is not allocated to the EDW component.

Shared Receiver Options

The following options are shared among all receiver types. Generally these shared options are set automatically for you when you add receivers.

`-c={<x>.<x>.<x>.<x>:<y>|<host_name>.<domain>}`

Specifies the Parser to which the receiver connects. You can specify the IP address and port number, or you can specify the fully qualified DNS name.

IMPORTANT: Do not change the value of the -c option. It is set for you automatically when you add the receiver.

`-L={NULL|stdout|stderr}`

Adds log destinations for the receiver to record operational information.

`-l={NULL|stdout|stderr}`

Replaces the syslog destination for the receiver to record operational information with the destination specified.

`-n`

The name of the receiver. The value for this option is set automatically for you when you add a receiver. Do not use this option to rename a receiver. Instead, remove the current receiver and add a new one with the name you want.

`-P=/<path>/<filename>`

The path and filename of the process ID (PID) file for the receiver. A PID file contains the process ID of the receiver so that you can kill it if necessary. This parameter is set automatically for you when you add the receiver, with the following value:

```
/<HawkEye AP_Home>/latest/var/run/sensage_<module_name>
```

```
-U=<username>
```

The user for the receiver process and files. Generally the user, the HawkEye AP system user, is the same for all HawkEye AP components. You set the HawkEye AP system user with the sensage-user option when you configured the non-HawkEye AP components; you set the HawkEye AP system user with the sls-user option when you configured the EDW component and when you specified the target EDW.

This value defaults to the sensage-user automatically when you add the receiver. Use the “-U” option only if you want the receiver to run under the authorization of someone other than the HawkEye AP system user.

```
--load-balance
```

If you specify this flag, the Receiver load balances among several Parsers. By default, this options is turned off, meaning that only a single Parser is used. If that Parser becomes available, the Receiver uses an automatically-configured standby Parser instance.

For more information, see Chapter 2: Configuring Parsers and Receivers in the *Event Collection Guide*.

LEA Receiver Options

To add a new LEA Receiver or edit the configuration of an existing one, use the following syntax:

```
clsetup { add | edit } rt --host=<host_name> --options="--file=
<Sensage_Home>/latest/data/rt/<LEA_receiver_name>.state.txt --stateDB=
<Sensage_Home>/latest/data/rt/<LEA_receiver_name>.state --sic=OPSEC_SSLCA --
opsecConfig=<Sensage_Home>/latest/data/rt/opsec.conf --mode=UNIFIED" leareceiver
<LEA_receiver_name>
```

IMPORTANT: LEA Receivers require the initial plain-text state file and an OPSEC configuration file.You create and update your LEA configuration file separately from the clsetup command. The file is a Unix state file. LEA Receivers fail to start if you do not specify your configuration file when you add or edit them.

The following options are supported by LEA receivers.

Mandatory Options

```
-stateDB=/<path>/<state_filename>
```

The name of a Unix state file that contains detailed configuration information for the LEA receiver. When you add an LEA receiver, the state file does not exist. The receiver creates and maintains it after it connects successfully for the first time with its LEA servers.

NOTE: This option is required.

For more information on the state file and general information on using LEA receivers, see Chapter 2: Configuring Parsers and Receivers in the *Event Collection Guide*.

-file=/<*path*>/<*text_filename*>

The name of a text file that contains bootstrap information for the LEA receiver, providing a set of servers to connect to and initial file retrieval. This option is also used for backwards compatibility with earlier versions of the LEA receiver. Add an LEA Receiver with this option if you have an existing text-based configuration file. After you add the receiver and it starts up, it copies the information from the text file and places it a new Unix state-based configuration file. Then you should edit the receiver you added and remove the value for the `--file` option. If you start an existing LEA receiver and you do specify a text file, any information it contains is added the Unix state file that the receiver maintains.

NOTE: This option is required when you add a HawkEye AP LEA receiver and run it for the first time.

The following options are mandatory because they must match the server's configuration.

--SIC[no}=<*sic_method*>

The SIC (Secure Internal Communication) method you want to use between the firewall and the LEA receiver. The following values refer to basic forms of secure connections and/or authentication; please refer to Check Point documentation for precise definitions: NO_SIC (no connection encryption, clear text) OPSEC_SSL, OPSEC_SSLCA, and OPSEC_NONE.

If you specify `-nosic`, no secure communication is attempted.

--mode={RAW|SEMI|UNIFIED}

The OPSEC record mode. Required if you specify a value for the `-SIC` option. The default value for `-mode` is `UNIFIED`.

--opsecConfig=/<*path*>/<*opsec_filename*>

The OPSEC configuration file. You must specify a value for this option when you specify a value for `--SIC`. For details, see "Enabling SIC for the OPSEC client", on page 79 of the *Installation, Configuration, and Upgrade Guide*.

Specific Options

--dest=<*destination*>

This destination is used to send the data for parsing. The default is the IP and port specified for the first configured Parser. Specifying a specific destination such as an IP address can be useful for initial debugging. For example:

```
--dest=127.0.0.1:3333
nc -l 3333
```

The Parser listens on port 3333 and parses the data stream as it arrives.

--throttle=<*integer*>

Throttle the flow of incoming messages down to approximately the number of messages per second specified by *<integer>*. Use this option to avoid heavy processing loads on firewalls and the network. If you do not place a throttle on the receiver, it draws in log events as fast as they occur, which can overburden the firewall or clog the network with excess traffic. The default is 0.

`--lagtime=<integer>`

Lag behind streamed log events for the number of seconds specified by *<integer>*. Lagging assists with UNIFIED record mode. Specify this option only when you specify `UNIFIED` with the `-mode` option.

`--logheartbeat=<integer>`

The frequency in seconds at which the receiver generates information for each LEA session. The default value is `120`.

`--maxColThreads=<integer>`

The maximum number of receiving threads to run simultaneously. The default value is `1`. Use this option to increase the number of collected log files you want to process in parallel.

For more information, see Configuring LEA Receivers in Chapter 2, “Configuring Parsers and Receivers” in the *Event Collection Guide.*

Debugging Options

`--offline`

Terminate the receiver at the end of the current, or active log file.

`--terminateEOF`

Removing a Parser Real-Time Module or a Receiver (clsetup)

To remove a real-time process, use the following syntax:

```
clsetup [<general_options>] remove rt [<specific_options>]
    parser|<receiver_type> <parser_name>|<receiver_name>
```

Arguments

`parser|<receiver_type>`

Removes only processes of the specified type: the parsers or the receivers; if omitted, removes all real-time processes regardless of type.

`<parser_name>|<receiver_name>`

Removes only the process specified; if omitted, removes all processes of the specified type.

Adding, Changing, and Removing Source Health Definitions (clsetup)

To configure source health monitoring, use the following syntax:

```
clsetup [<general_options>] [add | change | remove] sourcehealth \
  <shm_name> [--definition=<filename>]
```

You add a definition file for each log source that you want to monitor for source health. The file identifies the tables to be monitored, the column or columns that identify log sources within the table, the monitoring frequency for the table, and the variance from historical volumes allowed before a source health alert is raised.

NOTE:

- You can use the `change` option to modify any attribute value at any time. If you change any of the scheduling attributes, previously collected data remains unchanged.
- If you change the query or the table and expression elements, the system removes the current set of historical event data and the list of sources. If you change the tables or log sources, it is assumed that the previous data is no longer relevant.
- When you remove a definition, the system removes all information about the definition.

For more information, see Chapter 11: Monitoring Source Health.

Arguments

<filename>

Specifies a file that contains the source health monitoring definition for a specific log source in the target EDW (SLS) instance of the Real-Time component. Although the format of this file is XML, you do not need to include a `.xml` extension.

Managing HawkEye AP with Clsetup

This section describes using `clsetup` utility to perform specific management tasks:

- “Starting, Stopping, and Restarting Your HawkEye AP Deployment (clsetup)”, on page 58
- “Listing all EDW Instances (clsetup)”, on page 55
- “Disabling or Deactivating an EDW Instance (clsetup)”, on page 56
- “Obliterating an EDW Instance (clsetup)”, on page 57
- “Listing all Real-Time Modules (clsetup)”, on page 58
- “Starting, Stopping and Restarting the Real-Time Modules (clsetup)”, on page 59
- “Starting, Stopping, and Restarting the Collector (clsetup)”, on page 60
- “Listing Source Health Definitions (clsetup)”, on page 61
- “Checking Operating Status (clsetup)”, on page 61

Starting, Stopping and Restarting an EDW Instance (clsetup)

```
clsetup [<general_options>] { start | stop | restart }
    sls <instance_name>.
```

Arguments

<instance_name>

The instance reference is a required argument that provides the name of the EDW instance. You can specify the instance reference in any of the following formats:

<host>:*<port>*

For example: `host1.myco.com:8072`

<host>:*<instance_name>*

For example: `host1.myco.com:my_instance`

<instance_name>

For example: `my_instance`

NOTE: Use the *<instance_name>* argument only for instances that reside on the same host from which you run the command. If you specify only one instance reference, the command includes all hosts that are part of that instance.

NOTE:

- You can use *<host>*:*<port>* and *<host>*:*<instance_name>* for instances that reside on the local host or remote hosts.
- As a convenience, you can omit this argument if you set the environment variable `SLS_INSTANCE` in the session where you run the `clsetup` command. For example:

```
SLS_INSTANCE=localhost:8072
```

Listing all EDW Instances (clsetup)

Display configuration details of the EDW and its instances:

```
clsetup [<general_options>] list sls [<specific_options>] [<instance_name>]
```

Arguments

<instance_name>

Specifies which EDW instance for which you a detailed configuration list. If you omit this argument, detailed configuration lists are provided for all EDW instances.

You can specify the instance reference in any of the following formats:

<host>:*<port>*

For example: `host1.myco.com:8072`

<host>:*<instance_name>*

For example: `host1.myco.com:my_instance`

`<instance_name>`

For example: *`my_instance`*

NOTE: Use the *`<instance_name>`* argument only for instances that reside on the same host from which you run the command. If you specify only one instance reference, the command includes all hosts that are part of that instance.

NOTE:

- You can use *`<host>`*`:`*`<port>`* and *`<host>`*`:`*`<instance_name>`* for instances that reside on the local host or remote hosts.
- As a convenience, you can omit this argument if you set the environment variable `SLS_INSTANCE` in the session where you run the clsetup command. For example:

```
SLS_INSTANCE=localhost:8072
```

Specific Options

`--show-datadir`

Displays the location and status of the data directories, in particular which instances are sharing that data directory (for example, when two instances are coupled).

`--long`

Lists the complete set of parameters specified available for `clsetup`.

`--short`

Lists only instance references, one per line. Useful for scripts when you want a short list of all instances.

`--restrict-to-host`

Lists only instances running on the specified host.

`--restrict-to-instance`

Lists only instances running on the hosts used by the specified instance.

Aliases

```
-s = "--short"
-l = "--long"
```

Disabling or Deactivating an EDW Instance (clsetup)

Deactivate the specified instance:

```
clsetup [<general_options>] deactivate sls
    { <instance_name> | <host>:<port> | <host>:<instance_name> }
```

After you deactivate an EDW instance, the clsetup command acts as if it does not exist. It is not shown when you list the EDW instances, and you cannot start it. However, the log data remains in the data store for the instance.

Use the `add` command to activate an instance that you previously deactivated; for example:

```
clsetup add sls { <instance_name> | <host>:<port> | <host>:<instance_name> }
```

You must use the same *<instance_name>* or *<port>* that you used when you initially added the EDW instance.

Arguments

```
<instance_name> | <host>:<port> | <host>:<instance_name>
```

Reference to an EDW instance. Use <*instance_name*> only for instances that reside on the same host from which you run the `clsetup` command. Use *<host>*:*<port>* or *<host>*:*<instance_name>* for instances that reside on the local host or remote hosts.

NOTE: As a convenience, you can omit this argument if you set the environment variable `SLS_INSTANCE` in the session where you run the `clsetup` command.

Reactivating

The instance data remains after you run the command above. You can later reactivate the instance using:

```
clsetup add sls <instance_name>
```

Adding a Host to an EDW Instance

To add a host to an existing EDW Instance:

```
clsetup extend sls <instance_name> --hosts=<list of all hosts, including the
new host(s)>
```

For complete instructions, see Expanding an EDW Instance in Chapter 2, “Configuring HawkEye AP” of the HawkEye AP *Installation, Configuration and upgrade Guide*.

Obliterating an EDW Instance (clsetup)

To obliterate an EDW instance, use the following syntax:

```
clsetup [general_options] obliterate sls [<specific_options>]
    { <instance_name> | <host>:<port> | <host>:<instance_name> }
```

The instance is not obliterated until you confirm your intent. You can confirm interactively after you issue the command, or you can confirm with the specific option `--confirm`.

IMPORTANT: When you obliterate an EDW instance, the data-store directories for the instance are deleted from the file system. Log data previously stored in the instance cannot be recovered.

Arguments

```
{ <instance_name> | <host>:<port> | <host>:<instance_name> }
```

Reference to an EDW instance. Use <*instance_name*> only for instances that reside on the same host from which you run the `clsetup` command. Use <*host*>:<*port*> or <*host*>:<*instance_name*> for instances that reside on the local host or remote hosts.

NOTE: As a convenience, you can omit this argument if you set the environment variable `SLS_INSTANCE` in the session where you run the `clsetup` command.

Specific Options

`--confirm`

Suppresses interactive confirmation of the command.

Starting, Stopping, and Restarting Your HawkEye AP Deployment (clsetup)

To start and stop your entire HawkEye AP deployment, run the following command as root. The syntax is:

```
clsetup [<general_options>] { start | stop | restart } sensage
```

This command stops, starts, or restarts the EDW and the other HawkEye AP components on the host from which you run the command. Although you must run the command as root, when HawkEye AP starts, it runs as the HawkEye AP user. See Rootless Operation in Chapter 1, “Installing HawkEye AP” in the *Installation, Configuration, and Upgrade Guide.*

Listing all Real-Time Modules (clsetup)

To list the configuration details of the real-time modules, use the following syntax:

```
clsetup [general_options] list rt [specific_options]
    [<module_type> [<module_name>]]
```

Arguments

<module_type>

Lists only modules of the specified type; if omitted, lists all real-time modules regardless of type.

<module_name>

Lists only the module specified; if omitted, lists all modules of the specified type.

Specific Options

`--host`

Lists modules running on the specified host.

NOTE: See also “Configuring the Application Manager and the Real-time Component(clsetup)”, on page 39.

Starting, Stopping and Restarting the Real-Time Modules (clsetup)

Use the `clsetup` command to start, stop and restart the real-time modules. The syntax is:

```
clsetup [<general_options>] { start | stop | restart } rt \
  [<module_type>] [<module_name>] [--host=<host>]
```

NOTE: See also “Configuring the Application Manager and the Real-time Component(clsetup)”, on page 39.

IMPORTANT: As described below, the options and combinations of options you specify determine which components are affected by this command:

- **no options**

`restart`—restarts the Parser, receivers, the Collector, and the Application Manager.

`start` and `stop`—start and stop all real-time modules.

- **host only**

`restart`—restarts the Parser, receivers, the Collector, and the Application Manager only on the specified host.

`start` and `stop`—start and stop all Real-Time modules only on the specified host.

- **module type only**

`restart`, `start`, and `stop`— restart, start, or stop all modules of the specified type.

Example—You have configured more than one Parser and you specify "`Parser`" as the module type, the command operates on all Parsers.

- **module type and host**

`restart`, `start`, and `stop`— restart, start, or stop only modules of the specified type on the specified host.

Example—You have configured more than one Parser on different hosts and you specify "`Parser`" as the module type, the command operates on only on Parser modules on the specified host.

- **module type and module name**

`restart`, `start`, and `stop`— restart, start, or stop only the specified module of the specified type

Example—You have configured more than one Parser and you specify "`Parser`" as the module type and identify the module by name, the command operates on only the specified Parser module.

Arguments

<module_type>

Specifies the type of module to start, stop, or restart. The module types are: `parser`, `leareceiver`, `snmpreceiver`, `Collector`, and `appserver`. If this argument is omitted, `clsetup` starts, stops, or restarts all modules.

<module_name>

Specifies the name of the module to start, stop, or restart. If this argument is omitted, `clsetup` starts, stops, or restarts all modules of the specified type.

Specific Options

`--host=<host>`

Starts, stops, or restarts the Real-Time component or specified modules running only on the specified host(s).

Starting, Stopping, and Restarting the Collector (clsetup)

Use the following command to start, stop and restart the Collector. The syntax is:

```
clsetup [<general_options>] { start | stop | restart } rt collector \
[--host=<host>]
```

- The `stop` argument forces shutdown of the Collector. To cause HawkEye AP to attempt to complete current operations before shutting down the Collector, run the following command:

  ```
  <Sensage_Home>/latest/bin/collectd [--host=<host>] graceful
  ```

- The `restart` argument causes HawkEye AP to attempt to complete current operations before shutting down the Collector and then restarts the Collector.
- Although the Collector is a component and is not a module of the Real-Time component, you must include “`rt`” in the `clsetup` command to control the Collector.

NOTE: See also: “Adding, Changing, and Removing Source Health Definitions (clsetup)”, on page 54 and Chapter 7: Administering the Collector.

Specific Options

`--host`

Starts, stops, or restarts the Collector that is running only on the specified host.

If you configured the Collector to run on more than one host, you are running more than one Collector. To operate on a specific Collector, specify its host in the `--host` flag. To operate on all Collectors, omit the
`--host` flag.

Listing Source Health Definitions (clsetup)

To list all source health definitions or the text that defines one, use the `clsetup` command with the following syntax:

```
clsetup list sourcehealth [<shm_name>]
```

If you omit the source-health-monitor name, the command returns the name of every existing definition.

If you provide the source-health-monitor name, the command returns the text that defines the specified definition, if one exists.

Checking Operating Status (clsetup)

Occasionally you want to know whether a particular HawkEye AP component or module or a specific EDW instance is running or not. The clsetup command lets you check their status.

```
--status
```

Returns the operating status of EDW instances, all non-EDW components and modules, specific non-EDW components, Real-Time modules of a particular type, and specific Real-Time modules. If the specified EDW instance, other component, or module is running successfully, the status is:

```
[  OK  ]
```

—or—

```
is running...
```

To check the status of all EDW instances and non-EDW components and modules

```
clsetup status sensage
```

To check the status of the Real-Time component and its modules

```
clsetup status rt [<module_type> [<module_name>]]
```

To check the status of all EDW instances

```
clsetup status sls
```

To check the status of a specific EDW instance

```
clsetup status sls <instance_name>
```

General Syntax for Utility Tools

This section describes the general syntax used with utility tools and contains the following sections:

- “Options Default Values”, next

- “Getting Help”, on page 62
- “Editing Parameters”, on page 62
- “Control of execution”, on page 63
- “Remote Execution”, on page 64
- “Shared Mandatory Fields with Default Values”, on page 64

Each of these utilities has arguments and options. The options fall into two categories: those generally available to all utilities, and those available only to specific utilities. As shown below, you specify an option by preceding its name with two dashes, for example: `--hosts`. To specify a value for the option, you can precede the value either with a space or an equals sign (=). Therefore, both of the following examples are valid:

```
--hosts=my_machine
--hosts my_machine
```

NOTE: If you specify the equals sign (=), do not put a space on either side of it.

Options Default Values

There are options with defaults; if the option is not explicitly specified, the default value is used. In most cases, the online help for the tools will show you what the default value is by showing the default value in parenthesis.

Some options have a default value. For example, the value of the `--hosts` option defaults to `localhost`. To use the default value, do not specify the flag on the command line. To use a non-default value, include the flag and specify the desired value.

Getting Help

`--help`

Displays usage information

`--longhelp`

Displays detailed usage information

Editing Parameters

`--config <`*`valid_path`*`>`

Reads parameters from specified files.

`--[no]edit`

Invokes editor to enter missing or incorrect parameters; default is `--edit`. If you specify `--noedit`, does not open the editor when parameters are missing or incorrect.

`--editor [<`*`editor`*`>]`

Uses specified editor to edit missing or erroneous parameter; defaults to `vi`. Alternatively, you can specify the value in the `EDITOR` or `VISUAL` environment variable.

Control of execution

`--debug`

Displays debug data while the command is running

`--debug`

Displays debug data while the command is running

`--fips`

Turns on FIPs mode to enforce the Federal Information Processing Standard encryption protocol for U.S government processing.

`--force`

Ignores errors

`--parse-only`

Parses and validates the command line options, but does not execute

`--preview`

Displays actions that would be executed, but stays on local host.

`--[no]progress`

Displays progress indicators by default. If you specify `--noprogress`, disables display of progress indicators;

`--remote-preview`

Displays actions that would be executed, including those on remote hosts

`--verbose=<`*`information_level`*`>`

Displays increasingly detailed progress information

`--version`

Displays version information

`--[no]welcome`

Displays welcome banner with change number; defaults to `--welcome`

Remote Execution

`--prompt-password`

Always prompt for passwords; does not cache or reuse passwords from other hosts

`--serialize`

Executes tasks on one host at a time; does not run in parallel

`--timeout=<seconds>`

Expiration time in seconds if a task cannot successfully execute.

Shared Mandatory Fields with Default Values

In the CLSETUP syntax, fields/options marked as "mandatory" are those in which the user must specify path information for locating/placing certain resources on each physical server. The configuration utility provides default values for all of these fields./options. If you do not change the default value, you do not need to specify these fields on the command line. Note that HawkEye AP requires that resources are placed in consistent locations across all servers.

`--gnu-dir`

Specifies the valid path on all machines for the directory containing GNU utilities (including `tar` and `gunzip`).

`--java-dir`

Specifies the valid path on all machines for the directory that contains a usable Java JDK

`--rsync-dir`

Specifies the valid path on all machines for the directory containing the `rsync` program

`--openssl-dir`

Specifies the valid path on all machines for the directory containing the `openssl` program

`--ssh-dir`

Specifies the valid path on all machines for the directory containing the `ssh` programs

USING UTILITY TOOLS

This section describes utility tools to perform the following general tasks:

- “Comparing Files or Directory-listings Across the Cluster (cldiff)”, on page 65
- “Running a Command on Each Host in the Cluster (clssh)”, on page 67
- “Copying Files and Directories to Each Host (clsync)”, on page 69
- “Merging the Results from Top on Each Host in the Cluster (cltop)”, on page 71

- “Listing hosts in the cluster (clhosts)”, on page 73

Comparing Files or Directory-listings Across the Cluster (cldiff)

This section describes the `cldiff` utility. For each file or directory specified compares the specified file with identically named files on each host in the cluster. For directories, `cldiff` compares directory listings.

This section contains the following topics:

- “Synopsis”, next
- “Arguments”, on page 65
- “Options”, on page 65
- “Running a Command on Each Host in the Cluster (clssh)”, on page 67

Synopsis

```
cldiff [general_options] [specific_options] <source_item> [<source_item> […]]
```

Arguments

`<source_item>`

You must specify at least one source item. Additional source items are optional. Separate multiple items with spaces. You can specify files, directories, or combinations of both, as the following three examples show:

```
cldiff file1 tmp/file2.txt file3
cldiff dir1 dir2
cldiff file1 tmp/file2.txt dir1
```

If you specify a path without a leading slash (`/`), the utility resolves the path from the current directory. The utility expects to find the files and directories in the same current directory on the other hosts in the cluster. If the corresponding files are in a different location on the other hosts, use the `--target-dir` option to specify the alternate, remote location.

Options

`[general_options]`

See “General Syntax for Utility Tools”, on page 61.

`[specific_options]`

`--hosts`

Executes command on hosts specified in the list; defaults to `localhost`

`--sls-instance`

Executes command on every host on which the specified instance is running; value can also be set through the `SLS_INSTANCE` environment variable

`--target-dir`

The target directory on remote hosts where the copies to be compared are located. Use this option if the location of files to compare are in different locations on the remote hosts than on the local host.

If the remote locations on remote hosts differ by host, use the `<host>:<path>` syntax for specifying hosts with the `--hosts` option described above.

`--exclude`

When it examines directories for files to compare, the `cldiff` utility ignores files and directories with base names that match the specified pattern. Separate multiple patterns with spaces or use multiple --exclude options. Enclose the patterns within single quotes, for example:

```
"cldiff -exclude=’**/dsroot/**’/opt /sensage/latest/etc"
```

`--diff-command`

The `cldiff` utility uses the `"rsync -n (preview)"` command to discover files that are different. When it discovers files that are different, `cldiff` uses the `"diff -c"` command to report the differences.

You can modify the way in which differences are reported by specifying the options you want the diff command to run with; for example:

```
--diff-command (diff -c, default)
```

```
--diff-command (diff -c)
```

Example

Following is an example of the cldiff command executed using the --exclude option on a two-node EDW instance named `mysls`.

```
[dev-01 latest]# cldiff --exclude='*/dsroot/*' --sls-instance=mysls /opt/
sensage/latest/etc/sls/instance/mysls
cldiff - clsetup-4.7.0, change #73647;

Initializing (ssh) root@dev-01.sensage.com
Initializing (ssh) root@dev-02.sensage.com
dev-01:Done | dev-02:Done
All done.

====[dev-01.sensage.com]====
Identical files:
= /opt/sensage/latest/etc/sls/instance/mysls/PERMISSIONS
= /opt/sensage/latest/etc/sls/instance/mysls/add.xml
= /opt/sensage/latest/etc/sls/instance/mysls/athttpd.conf
= /opt/sensage/latest/etc/sls/instance/mysls/cluster.xml
= /opt/sensage/latest/etc/sls/instance/mysls/docroot
= /opt/sensage/latest/etc/sls/instance/mysls/dsroot
= /opt/sensage/latest/etc/sls/instance/mysls/ldap.conf
= /opt/sensage/latest/etc/sls/instance/mysls/md5
= /opt/sensage/latest/etc/sls/instance/mysls/md5.errors
= /opt/sensage/latest/etc/sls/instance/mysls/sensage_sls_mysls
= /opt/sensage/latest/etc/sls/instance/mysls/shared_secret.asc

====[dev-02.sensage.com]====
Only on "dev-02.sensage.com":
> /opt/sensage/latest/etc/sls/instance/mysls/omnisight.ldif

Identical files:
= /opt/sensage/latest/etc/sls/instance/mysls/PERMISSIONS
= /opt/sensage/latest/etc/sls/instance/mysls/add.xml
= /opt/sensage/latest/etc/sls/instance/mysls/athttpd.conf
= /opt/sensage/latest/etc/sls/instance/mysls/cluster.xml
= /opt/sensage/latest/etc/sls/instance/mysls/docroot
= /opt/sensage/latest/etc/sls/instance/mysls/dsroot
= /opt/sensage/latest/etc/sls/instance/mysls/ldap.conf
= /opt/sensage/latest/etc/sls/instance/mysls/md5
= /opt/sensage/latest/etc/sls/instance/mysls/md5.errors
= /opt/sensage/latest/etc/sls/instance/mysls/sensage_sls_mysls
= /opt/sensage/latest/etc/sls/instance/mysls/shared_secret.asc

====[Summary]====
Identical on all hosts:
= /opt/sensage/latest/etc/sls/instance/mysls/PERMISSIONS
= /opt/sensage/latest/etc/sls/instance/mysls/add.xml
= /opt/sensage/latest/etc/sls/instance/mysls/athttpd.conf
= /opt/sensage/latest/etc/sls/instance/mysls/cluster.xml
= /opt/sensage/latest/etc/sls/instance/mysls/docroot
= /opt/sensage/latest/etc/sls/instance/mysls/dsroot
= /opt/sensage/latest/etc/sls/instance/mysls/ldap.conf
= /opt/sensage/latest/etc/sls/instance/mysls/md5
= /opt/sensage/latest/etc/sls/instance/mysls/md5.errors
= /opt/sensage/latest/etc/sls/instance/mysls/sensage_sls_mysls
= /opt/sensage/latest/etc/sls/instance/mysls/shared_secret.asc

None of the specified files are different.

None of the specified files exist only on localhost.

Exist only on remote hosts:
> /opt/sensage/latest/etc/sls/instance/mysls/omnisight.ldif dev-02.sensage.com

[root@dev-01 latest]#
```

Running a Command on Each Host in the Cluster (clssh)

This section describes the `clssh` utility, which runs the specified command in parallel on a specified set of hosts.

This section contains the following topics:

- "Synopsis", next
- "Arguments", on page 68
- "Options", on page 68

- “Example”, on page 69
- “Returns”, on page 69

Synopsis

```
clssh [general_options] [specific_options] <ssh_command> [ssh_options]
```

Arguments

`<ssh_command>`

You must specify an `ssh` command to run.

Options

`[general_options]`

See “General Syntax for Utility Tools”, on page 61.

`[specific_options]`

`--hosts`

Executes command on hosts specified in the list; defaults to `localhost`

IMPORTANT: If you use curly braces `{}` to factor out common prefixes (for example, `host{0-9}`) and you run `clssh` on `csh` or `tcsh`, enclose the braces in quotation marks (for example, `"host{0-9}"`)

`--sls-instance`

Executes command on every host on which the specified instance is running; value can also be set through the `SLS_INSTANCE` environment variable.

`--[no]cwd`

Uses current working directory on remote hosts; if `--nocwd` is specified, uses the user’s home directory; defaults to `--cwd`.

`--in-dir ([current directory])`

Uses specified directory on remote hosts; if is specified, uses the user’s home directory; defaults to the user’s current directory.

`--user`

Uses the specified userid for remote execution.

`--root`

Uses root for remote execution.

`--make-root-trust`

Generate SSH key pairs for root trust and distribute them. For more information, see Setting Up SSH Trust in Chapter 1, “Installing HawkEye AP” in the *Installation, Configuration, and Upgrade Guide*.

Example

```
% bin/clssh ls /home/lms/cg_2002
Welcome to clssh version #8245.

Initializing (ssh) cim@host01.myco.com

cim@host01.myco.com's password:
Initializing (ssh) cim@host01.myco.com
* host01:-Done- | host08:-Done-
All done.

====[host01.myco.com]====
clsetup
bin
conf
contrib
docroot
dsroot
example_logs
lib
misc

====[host08.myco.com]====
clsetup
bin
conf
contrib
docroot
dsroot
example_logs
lib
misc
```

Returns

The standard output and standard error from executing the command on each host, organized by host.

Copying Files and Directories to Each Host (clsync)

This section describes the `clsync` utility, which synchronizes local files and directories with files and directories on specified remote hosts.

- For files, ensures that the file is replicated in the same location on each host.
- For directories, this utility recursively synchronizes the contents of the directory.

This utility uses the `rsync` program.

This section contains the following topics:

- “Synopsis”, next

- "Arguments", on page 70
- "Options", on page 70
- "Returns", on page 71
- "Returns", on page 71

Synopsis

```
clsync [general_options] [specific_options] <source_item> [<source_item> […]]
```

Arguments

<source_item>

You must specify at least one source item. Additional source items are optional. Separate multiple items with spaces. You can specify files, directories, or combinations of both, as the following three examples show:

```
clsync file1 tmp/file2.txt file3
clsync dir1 dir2
clsync file1 tmp/file2.txt dir1
```

If you specify a path without a leading slash (`/`), the utility resolves the path from the current directory. The utility synchronizes the local files and directories with those in the same current directory on the other hosts in the cluster. If the corresponding files are in a different location on the other hosts, use the `--target-dir` option to specify the alternate, remote location.

Options

`[general_options]`

See "General Syntax for Utility Tools", on page 61.

`[specific_options]`

`--hosts`

Executes command on hosts specified in the list; defaults to `localhost`

`--sls-instance`

Executes command on every host on which the specified instance is running; value can also be set through the `SLS_INSTANC`e environment variable

`--target-dir`

The target directory on remote hosts where the files and directories to be synchronized are located. Use this option if the location of files to synchronize are in different locations on the remote hosts than on the local host.

If the remote locations on remote hosts differ by host, use the *<host>*:*<path>* syntax for specifying hosts with the `--hosts` option described above.

`--exclude`

When it examines directories for files to synchronize, the `clsync` utility ignores files and directories with base names that match the specified pattern. Separate multiple patterns with spaces or use multiple --exclude options. Enclose the patterns within single quotes, for example:

```
clsync --exclude='**/dsroot/**' /opt/sensage/latest/etc
```

`--include`

Use the `--include` option to selectively override excluded items. When it ignores files and directories because of the `--exclude` option, the `clsync` utility does not ignore files and directories with base names that match the pattern specified for `--include`. Separate multiple patterns with spaces or use multiple --include options. Enclose the patterns within single quotes, for example:

```
clsync --exclude='**/dsroot/**' --include='*.xml' /opt/sensage/latest/etc
```

Returns

A report of what `clsync` did, organized by host.

Example

Following is an example of the clsync command executed on a two-node cluster: dev-01,dev-02. In this example, a file is missing on dev-02. The clsync command identifies the missing file and copies it from dev-01 to dev-02, bringing the two nodes back in sync.

```
[root@dev-01 latest]# clsync --hosts=dev-01,dev-02 /opt/sensage/latest/sql/
clsync - clsetup-4.7.0, change #73647;

Initializing (ssh) root@dev-01.sensage.com
Initializing (ssh) root@dev-02.sensage.com
Pending: 0 Lagging: - last file copied: .../oae/sls_name.sql
Done

====[dev-01.sensage.com]====
/opt/sensage/latest/sql/ -> <same place>: no files copied.
No files copied.

====[dev-02.sensage.com]====
/opt/sensage/latest/sql/oae/sls_name.sql -> <same place>

Total number of files copied: 1

[root@dev-01 latest]#
```

Merging the Results from Top on Each Host in the Cluster (cltop)

This section describes the `cltop` utility, which creates a merged view of the operating system status across an entire cluster of computers. In other words, it is a cluster-wide version of the `top` program.

This section contains the following topics:

- “Synopsis”, next

- “Arguments”, on page 72
- “Options”, on page 72
- “Example”, on page 72
- “Returns”, on page 73

Synopsis

```
cltop [general_options] <instance_name>
```

Arguments

```
<instance_name>
```

The instance reference is a required argument that provides the name of the EDW (SLS) instance. You can specify the instance reference in any of the following formats:

```
<host>:<port>
```

For example: `host1.myco.com:8072`

```
<host>:<instance_name>
```

For example: `host1.myco.com:my_instance`

```
<instance_name>
```

For example: `my_instance`

NOTE: Use the `<instance_name>` argument only for instances that reside on the same host from which you run the command. If you specify only one instance reference, the command includes all hosts that are part of that instance.

NOTE:

- You can use `<host>:<port>` and `<host>:<instance_name>` for instances that reside on the local host or remote hosts.
- As a convenience, you can omit this argument if you set the environment variable `SLS_INSTANCE` in the session where you run the clsetup command. For example:

```
SLS_INSTANCE=localhost:8072
```

Options

```
[general_options]
```

See “General Syntax for Utility Tools”, on page 61.

Example

```
% bin/cltop

Welcome to cltop version #8245.
```

```
Initializing (ssh) cimarron@p02.hq.myco.com

cimarron@p02.hq.myco.com's password:
Initializing (ssh) cimarron@p08.hq.myco.com
* p02.hq:-Done- | p08.hq:-Done-
All done.
NODE NODE  ..........CPU usage.........  .............. RAM usage ...........
 NUM ROLES idle      busy  (user+system)  total     free     used  shared    cache
  02  DMNQ    0.0% |  0.6% =>  0.5+0.1  1,507MB     4MB 1,502MB     0MB      0MB
  08  DMNQ    3.0% |  3.5% =>  0.5+3.0  1,004MB    21MB   983MB     0MB      0MB

NODE PROCESS VMEM  RAM  SHMEM PROC CPU                 INSTALL
 NUM   ID    (MB)  (MB) (MB)  STAT used  TIME PORT START  NAME   ACTION OTHER
  08 13277   10.2  10.0   1.8 S N  9.5%  0:29   10 Dec31     10    10 /usr/local/
bin/perl
  02 17851    2.5   0.3   0.3 S    0.0%  0:00 2002  2:14  cg_2002  init
  02 17854    2.1   0.2   0.2 S    0.0%  0:00 2002  2:14  cg_2002  init
  08  9522    5.6   5.5   4.8 S    0.0%  0:00 2002  2:14  cg_2002  init
  08 31134    5.6   5.6   4.9 S    0.0%  0:00 2005  3:11  cg_2005  init
  08  9519    5.0   5.0   4.1 S    0.0%  0:00 2002  2:14  cg_2002  init

KEY:
 - PROC STAT:
    D   uninterruptible sleep (usually IO)
    R   runnable (on run queue)
    S   sleeping
    T   traced or stopped
    Z   a defunct ('zombie') process
    W   has no resident pages
    <   high-priority process
    N   low-priority task
    L   has pages locked into memory
```

Returns

Prints out of memory usage, CPU usage and processes.

Listing hosts in the cluster (clhosts)

This section describes the `clhosts` utility, which lists the hosts in the cluster in a form suitable for use with other programs.

This section contains the following topics:

- "Synopsis", next
- "Arguments", on page 74
- "Options", on page 74
- "Example", on page 75

Synopsis

```
clhosts [general_options] [specific_options] <instance_name>
```

Arguments

<instance_name>

The instance reference is a required argument that provides the name of the EDW (SLS) instance. You can specify the instance reference in any of the following formats:

<host>:*<port>*

For example: `host1.myco.com:8072`

<host>:*<instance_name>*

For example: `host1.myco.com:`*my_instance*

<instance_name>

For example: *my_instance*

NOTE: Use the *<instance_name>* argument only for instances that reside on the same host from which you run the command. If you specify only one instance reference, the command includes all hosts that are part of that instance.

NOTE:

- You can use *<host>*:*<port>* and *<host>*:*<instance_name>* for instances that reside on the local host or remote hosts.
- As a convenience, you can omit this argument if you set the environment variable `SLS_INSTANCE` in the session where you run the clsetup command. For example:

```
SLS_INSTANCE=localhost:8072
```

Options

[*general_options*]

See “General Syntax for Utility Tools”, on page 61.

[*specific_options*]

`--cluster-xml`

Displays `cluster.xml` file

`--show-datadir`

Outputs *<host>*:*<datadir>* for all hosts

`--show-port`

Outputs *<host>*:*<port>* form of the instance reference.

NOTE: When you execute `clhosts --show-port` *<host>*:*<port>*, the results display the port of the local machine on which the command executes. You can use this information to convert *<name>* or *<host>*:*<name>* to *<host>*:*<port>*, which is the only format that the data-store utilities accept. For more information, see Chapter 3: Loading, Querying, and Managing the EDW.

Example

```
% clhosts doctests
Welcome to clhosts version #8245.

lms02.myco.com
lms08.myco.com
```

CHAPTER 3

Loading, Querying, and Managing the EDW

This chapter contains the following sections:

Loading Data into the EDW

This section describes the atload utility. It contains the following topics:

Synopsis

```
atload [<options>] <cluster_list> <table_name> <ptl_file>
    { [<log_file> [<log_file> [...]] | - }
```

Description

The atload utility loads source log events from the specified log file(s) into the specified table on an EDW instance. If the specified table does not exist, it creates the table. The *<ptl_file>* argument specifies a file that describes how the log events are transformed and loaded into an EDW instance.

The atload utility supports loading multiple log files with one invocation of the command. If you specify the names of multiple log files on the command line, they are loaded in the order specified. When specifying multiple log files, separate each name with a space. In addition, you can specify archive files that contain multiple log files.

International Support for Loading Data

The character encoding of *<ptl_file>* must be UTF-8. The character encoding of *<log_file>* can be ISO 8859-1 or one of the character encoding schemes you specify with the --data-char-encoding option. If not specified, the character encoding of log files processed by the atload command is assumed to be ISO 8859-1.

To determine the character encodings available in your environment, run the following command:

```
iconv -l
```

A list of supported character encodings displays.

For more information on character encoding, see "International Support in the EDW", on page 159.

Arguments

Argument	Description
<cluster_list>	*<cluster_list>* represents a comma-separated list of *<host>:<port>* pairs. The list must be enclosed in quotation marks ("). Specifying a list rather than a single host:port pair allows the utility to run even if the first of the specified EDW nodes is down. Each host:port pair identifies the EDW (SLS) instance; for example, edw01:8072. The command tries each host:port pair in order (left to right) until it finds one that allows a TCP (Transmission Control Protocol) connection. Failure to establish a TCP connection indicates that the EDW host is down. If the connection succeeds but the command fails, the command does not attempt to establish additional connections. Example: atload --user=administrator --password=changeme "edw01:8072,edw02:8072,edw03:8072" mytable src.ptl src.log
<table_name>	Destination table name (created if it doesn't exist).
<ptl_file>	The file that describes how to parse the log data.
<log_file>	One or more data files to load into the table. Use "-" to read log data from standard input.

Table names can be 1 to 255 characters long and can consist of letters, numbers and underscores (_).

Options

`--archive-add-filename`

Passes in the name of each file in an archive. When you load multiple log files from archive files, such as a .zip, .tar or .tar.gz files, this option prefixes the archive file name with the names of the interior files as each is processed. This option affects the $_log_filename automatic expression macro that can be used in PTL files to load the source log-file name as a column value in each loaded row.

For more information on $_log_filename, see “Automatic Expression Macros”, on page 87 and “Loading Information about the Log File”, on page 88.

NOTE: The --archive-add-filename option is relevant only when loading batched data, not streamed data.

`--archive-tempdir=<`*`directory_name`*`>`

Specifies the temporary directory to use for loading operations that involve archive files. Use this option to control where temporary files are stored when loading archive files, such as .zip, .tar, or .tar.gz.

The default is the current working directory.

`--batchload`

Loads all files in a single operation rather than individually. Batch loads can improve performance, but the rows loaded from all files have the same value in the _uploadid column.

`--clocksync`

Corrects for client-side clock drift.

IMPORTANT: Only use this option if the client running atload is also the client on which the log file being loaded was generated. This option attempts to compensate for clock mismatches between the client and the EDW server when determining the timestamp for rows loaded into the table.

`--compression={none|low|high}`

Specifies the level of compression for storing the new data. High compression decreases the disk space consumed by the data store of the EDW instance, but it increases the load time due extra processing overhead. The supported values are none, low, and high.

The default is high.

`--data-char-encoding=<`*`encoding`*`>`

Specifies the character encoding of source log files processed by the atload command. Supported values are:

- UTF-8
- UTF-16
- EUC-JP

- Shift-JIS
- UJIS

The character encoding of the log files processed during a single invocation of atload must all be the same. For example, assume the following command-line invocation:

```
atload --data-char-encoding='UTF-8' host5:8072 mytable \
    MyPtl.ptl log1.log log2.log log3.log
```

The log files log1.log, log2.log, and log3.log must all be written in UTF-8.

Regardless of the character encoding you specify for the log files, the character encoding of the PTL file must be UTF-8. The atload command converts source log events to UTF-8 before they are evaluated by the regular expression in the PTL file and stored as rows in the EDW.

The default encoding for log files is ISO 8859-1.

`--dbgprint-request`

Dumps the client-server request, then exits. This option forces atload to print the requests that it would have sent to the EDW and stop without actually loading the data. Use this option for debugging purposes.

`--filetype=<`*`type`*`>`

Specifies the data type of the files being loaded. Sometimes the files being loaded do not end with the proper extension, and the EDW cannot determine if decompression is required. Allowed values for <*type*> are ascii, gzip, bzip2, zip, tar, targz, and tarbz.

NOTE: The file you are loading must be of the type you specify.

`--finalsummary`

Prints a final summary table after completing its operation. Each line in the table reports information about one file-loading operation, including the number or log events loaded, performance statistics, number of retries, and success or failure.

`--help, --longhelp`

Displays online help for command in short form (--help), or with additional detail (--longhelp).

`--matchfailures={<`*`filename`*`>|-}`

Displays regular-expression match failures in the terminal or console window, or writes them to a file. The PTL file contains a regular expression that parses source log events from the log file. For debugging purposes it is useful to see the source log lines that the regular expression fails to match. The supported values are:

- <*filename*>—write match failures to the specified file. The file is created if it *does not* exist; the file is appended to if it *does* exist.

  NOTE: Although you can specify the --matchfailures option when atload runs on a schedule managed by the Collector, the preferred method is to use the <ParseFailures> element in loader definitions and include a <LogFile> element.

- -(a dash)—displays match failures in the terminal or console window where atload is running. Only specify this option when you run atload from the command line.

NOTE: Specifying the --matchfailures option slows the operation of atload when there are many match failures. This can occur if the incorrect PTL file is specified, because the majority of source log lines will fail to match the regular expression in the PTL file.

This option is overridden by the --raw option.

```
--namespace=<namespace>
```

The namespace in which to manage items. The default namespace is default. To specify the top level, use "" (empty quotation marks).

NOTE: When you run atload, you can specify a table name explicitly (for example, ns1.ns2.ns3.mytable) or use the --namespace=<*namespace*> option. When you use the namespace option, it specifies the namespace once for the entire command; the utility prefixes the namespace to each table name used in the rest of the request.

```
--on-error={continue|stop}
```

Specifies how to process remaining log files after one of them fails to load. This option applies only when you load multiple files with one invocation of atload. The supported values are:

- continue—proceed to the next log file and try to load it
- stop—do not load any remaining log files

The default is continue.

```
--override=<name>:<value>
```

Specifies values that override the default values declared for expression macros in the SQL section of the PTL file. Repeat this option for each expression macro that you want to override.

For more information, see Expression Macros in Chapter 10, “Sensage SQL” in the *Reporting Guide*.

```
--preprocregex=<regex>
--preprocschema=<name>:<type>[,<name>:<type>[...]]
```

Applies a regular expression to each source log line being loaded before it is processed by the indicated PTL file. These options let you strip out information that was prefixed to the log lines during preprocessing and store it in extra columns in the table.

For example, you can prefix each line of log data with the name of the source host that generated the data. The assumption is that a PTL file for processing the original log data already exists and that you want to re-use that PTL file for loading the annotated log data.

The regular expression indicates how to separate the annotations prefixed to log lines from the original log data. One of the matches from the regular expression must be the original line of log data. You can construct the regular expression to also return matches for the various annotations, which allows you to load those into the table along with the standard columns for that data source.

The schema indicates how to name the matches that are derived from the regular expression. One of the matches must be named ’_REST_OF_LINE_’ and must have a type of varchar. This is the match that contains the original log line of data and is fed to the PTL file. Other matches from the regular expression, if there are any, become additional columns in the target table.

Example

```
atload --preprocregex='([^,]*),(.*)' \
--preprocschema='UID:VARCHAR,_REST_OF_LINE_:VARCHAR' \
<host>:<port> <table_name> <filename>.ptl <log_file>
```

`--raw`

Prints the raw XML load results that the EDW returns, instead of printing formatted results. Use this option for debugging purposes.

This option overrides the --matchfailures option.

`--retries=<n>`

Specifies the number of times to retry loading an aborted load operation. Sometimes errors that cause load operations to stall and eventually abort correct themselves. Aborting stalled load operations without retrying a few times results in unnecessary and time-consuming recovery procedures.

The default is 3 retries.

`--retry-sleep=<n>`

Specifies the number of seconds to wait before retrying an aborted load operation. Retrying a load operation immediately after aborting does not give sufficient time for transient errors to correct themselves.

The default is 45 seconds.

`--segmentsize=<n>`

Specifies the number of rows stored in a new EDW leaf node. (The EDW stores data in a tree structure with leaf nodes.) The default value is 250,000 rows. This value is correct for most applications and should not be changed. It is recommended that you Consult Hexis Cyber Solutions Technical Support before changing this parameter.

`--timeout=<n>`

Specifies the number of seconds to wait before a stalled load operation aborts. Sometimes transient errors, such as network latency, cause loads to stall. Transient errors usually resolve themselves, and the load operation resumes at a regular pace. Other errors, such as hardware failures or locking contention, can persist indefinitely. Setting a time-out value allows a permanently stalled load to be aborted so that other load operations can occur. Set the value to 0 to wait indefinitely without timing out.

The default value is 0.

```
--trickle
```

Specifies that the EDW instance use the trickle feed (or trickle load) option for that LOAD, which means that the log data is loaded into an intermediate storage layout in the EDW. Trickle feed can also be set with the following method in Java EDW API: **LoadRequest::setTrickleFlag**.

To decide whether to use the trickle feed option, weigh user requirements against pros and cons and determine the best trade-off. As log files grow larger (or there is less overlap between them) a tipping point is reached in which performing normal, non-trickle loads makes most sense. On the other hand, when files are larger and there are longer time limits between the time-of-collection and the time-of-availability, the use of the trickle feed option (merge-load) starts to make more sense. Always consider the customer requirements against ideal scenarios to determine whether trickle feed is the appropriate loading solution.

Do use the trickle feed option in these cases:

- You have small files, collected at erratic intervals, no timestamp relationship between them, and short time limits between the time-of-collection and the time-of-availability. In this case, use the **--trickle** flag to load each file as it is collected. Optionally, you can add the **--trickle** flag to the Collector's **config.xml** file as an ATLOAD option in the appropriate cLoader definition.

  NOTE: Although there are no hard and fast fules for using the trickle feed option, optimum loading conditions (for example, when there are timestamp relationships between files) make it less likely that using the trickle load option will even be helpful.

- You have a data set being loaded has a small number of rows (less than the run length of a leaf times the number of nodes in the EDW cluster).This type of loading ensures that the collected log data is available within a short period time in the EDW. In such a case the loader is typically configured to run every few minutes to push whatever data has been collected into the EDW.
- You have data being loaded that is expected to ovelap (in terms of the time range of the log data) with other data sets that have been or soon will be loaded. This type of loading is for periodic data collection from a large number of sources in parallel; for example, pulling the last day's log data from 10,000 servers once a day, which results in as many log files that individually cover the same period of time. In this case, the result is 10,000 log files each of which covered the same 24-hour period.

The latter two use cases cause excessive fragmentation in the EDW datastore, resulting in significant performance issues when the loads are placed directly into the EDW long term storage layout. The trickle load operation directs the data set into an intermediate storage layout which supports efficient appending of out-of-order data. This avoids the load-time performance problems associated with the fragmentation of data in the long-term storage layout.

Both of these use cases cause excessive fragmentation in the EDW datastore, resulting in significant performance issues when the loads are placed directly into the EDW long term storage layout.The trickle load operation directs the data set into an intermediate storage layout which supports efficient appending of out-of-order data. This prevents load-time performance problems associated with the fragmentation of data in the long-term storage layout.

Queries against tables that have trickle-loaded data can show some performance degradation in some cases. Most notably bloom filters do not accelerate access to data in the intermediate storage layout. If you are using trickle loads, provide a schedule for periodically running a compact operation against the target table. Schedule the compact operation to occur once a day during a period of time where there is typically little other activity. The compact operation will retrieve whatever log data has been accumulated in the intermediate storage layout and merge.

When the Collector invokes EDW COMPACT operation, this allows the EDW data structures to stay small and prevents unnecessary delays when a manual compact operation is invoked after large amount of data is accumulated in TFL data structures.

Compaction parameters are defined within a <`Loader`> section that includes:

- Compaction schedule and /or
- A threshold determined by rows and/or bytes and/or load counts
- Loaders used against a given table. (more than one may be used)
- Compaction threshold statistics stored globally (within a given Collector) on a table-by-table basis.
- Compaction to lock out loads to a table globally (within given Collector).

The COMPACT parameters, when used, have the following attributes that result in a compact operation when the value for the given parameter is reached.

- loadsThreshold— Number of loads after which a compact operation is triggered (default - no trigger on this).
- rowsThreshold— Numbers of rows after which a compact operation is triggered (default - no trigger on this).
- megsThreshold— Number of megabytes after which a compact operation is triggered (default - no trigger on this)
- Schedule— Compact hours after midnight that triggers the compact operation (default - a list containing the number 0 (midnight hour)

If none of the threshold options are chosen, the Collector performs compaction according to the Schedule option defined in the parameter.

Examples:

```
<!-- simple parameterized compact ->
<Compact loadsThreshold="100" rowsThreshold="100000000"
megsThreshold="1000000000"/>

<!-- one parameter only, compact only at size threshold ->
<Compact megsThreshold="1000"/>

<!-- compact only at one minute after midnight - Schedule>
<Compact> <Schedule>1 0 * * *</Schedule> </Compact>

<!-- always compact at one minute after midnight, or after 10 gigs is loaded
since last compact -->
<Compact megsThreshold="10000">
<Schedule>1 0 * * *</Schedule>
</Compact>
```

`--verbose=<`*`n`*`>`

Where <n> is one of the following verbosity levels:

Verbosity Level	Meaning	Description
1	error	Displays errors
2	warning	Displays errors and warnings
3	progress	Displays errors, warnings, progress indication with ellipses (`...`), and parsing or row-loading errors with exclamation marks (`!`)
4	everything	Displays, errors, warnings, progress indication, parsing errors, and informational messages.

The default verbosity level is 2.

```
--version
```

Displays version of atload.

Parameters

Authentication Options

```
--user=<user_name>
--user=default
```

Runs the command under the authorization of *<user_name>*, who must be a user defined in the EDW instance. If you specify default, the atload command runs under the authorization of the user specified by the ADDAMARK_USER environment variable.

NOTE: To log in as the **guest** user, do not specify the `--pass` or `--shared-secret` flags.

The default is to run the command under the authorization of the currently logged-in user.

```
--pass=<password>
--pass=file:<path_and_filename>
```

Specifies the password that authenticates the user that you specified with the --user option. You can specify the password in clear text on the command line. You can place the password in a file and specify its location and file name on the command line. If you omit the --pass option, the atload command uses the password specified by the ADDAMARK_PASSWORD environment variable.

For more information, see “Setting the ADDAMARK_PASSWORD Environment Variable”, on page 86.

```
--shared-secret=<secret>
--shared-secret=file:<path_and_filename>
--shared-secret=default
```

Specifies the shared secret for the hosts in the EDW instance. You can specify the shared secret in clear text on the command line. You can place the shared secret in a file and specify its location and file name on the command line. If you specify default, the atload command uses the shared secret specified by the ADDAMARK_SHARED_SECRET environment variable.

The value for --pass takes precedence over the value for --shared-secret if both are present.

```
--skip-queues
```

Bypasses normal EDW task queueing.

You must specify the --shared-secret option when you use the --skip-queues option.

Setting the ADDAMARK_PASSWORD Environment Variable

To set the ADDAMARK_PASSWORD environment variable (for example, in the Bourne shell), submit one of the following commands.

- To set a clear-text password in the environment variable:

  Syntax

  ```
  bash$ export ADDAMARK_PASSWORD=<password>
  ```

  Example

  ```
  bash$ export ADDAMARK_PASSWORD=s0mep@ss
  ```

- To store the location and name of the file that contains a list of valid passwords:

Syntax

```
bash$ export ADDAMARK_PASSWORD=file:<path_and_filename>
```

Example

```
bash$ export ADDAMARK_PASSWORD=file:/local/home/sls/passwords
```

Hexis Cyber Solutions recommends that you use SSH or a similar protocol over a secure session to submit all authentication commands.

Loading Archive Files

When atload processes archive files, it uses the appropriate module to read the archive contents and extracts internal files into an individual `.gzip` files. It then uses the regular atload mechanism to load the `.gzip` files.

Large archive files can cause atload to run out of disk space on the client machine. To avoid running out of disk space, use the --archive-tempdir option to control where temporary files are stored.

Because the name of each internal file in an archive is lost after the data is combined, use the --archive-add-filename option to prefix every newline-separated record with the archive internal file name. If you use --archive-add-filename, you need to change your PTL file to parse this new field. For example, add ... to the regular expression, and add filename:VARCHAR to the list of parse fields.

Automatic Expression Macros

The atload utility provides these expression macros automatically for your use in the PTL SELECT statements in PTL files:

- $_log_filename
- $_log_srchost

For an example of their use, see “Loading Information about the Log File”, on page 88

$_log_filename

The automatic expression macro $_log_filename carries the filename of the log file currently being loaded. Files in Collector log queues often carry useful information in their filenames. You should parse and store this information as column values in the row being loaded. For example, you should store the entire filename and parse other useful information from it, such as the time the log file was created.

$_log_srchost

The automatic expression macro $_log_srchost carries the name of the machine from which the log file is being loaded. You should capture this information and store it as a column value in the row being loaded. This helps maintain the chain of custody for the log entry.

Loading Information about the Log File

The atload utility allows you to store information about the log files from which rows were loaded in a column of the data. Use the $_log_filename and $log_srchost expression macros in your PTL SELECT statements. The atload command declares the expression macros automatically.

For example, you have log entries from multiple log sources in a large file named:

```
snmp-2002-01-09_04:00:00.gz
```

The filename carries the timestamp when the log file was created—its so-called roll time. The log file is being loaded from a host named h03. The following PTL SELECT statement uses $_log_filename and $log_srchost to store the roll time and host name as column values in the row being loaded.

```
SELECT _strptime(_pregmatch($_log_filename, "(\d+-\d+-\d+_\d+:\d+:\d+)\.gz$"),
                                            "%Y-%m-%d_%H:%M:%S") AS _rolltime,
      $_log_srchost AS _srchost
  FROM stdin
;
```

TIP: Follow the convention of using an underscore (_) to begin the names of columns with metadata about the log entry to distinguish them from columns with data parsed from the log entry itself.

Reading Log Data from stdin

The atload utility reads the log data from standard input when you specify '-' as the log filename.

When you specify the log data in this manner, the data does not have a file name with an extension that identifies its file type. For example, when you load data as myload.tgz, the EDW recognizes the .tgz suffix as the tgz file type. Hexis Cyber Solutions recommends that you include the --filetype option when you specify '-' as a log file name. If you do not specify a file type, atload attempts to guess the type from the stdin input stream by examining the first few bytes.

Tracking Uploads in the EDW

In addition to loading the log data into the specified table(s), atload saves information in the EDW about the load. You can view this information by querying against two system tables:

- system.upload_info

  This table enables administrators to check on upload status in the system. When the system is healthy, this table returns one row for each upload. When the upload data is not consistent across all the hosts in the instance, this table returns a value of false in the CONSISTENT column. In this case, the table may return multiple rows with the same value in the _uploadid column; all of these rows are marked as inconsistent.

- system.raw_upload_info

  This table enables a support person to troubleshoot an inconsistency problem by seeing the raw data distributed across the system

Both of these tables contain a unique identifier for the upload: the value of the _uploadid column. For more information about the _uploadid column, see "Using the UPLOADS Command", on page 132 and "Table Management", on page 120.

These tables store additional information that includes: the minimum and maximum timestamps, the number of lines in the source log data, the number of lines successfully parsed by the PTL, the number of rows loaded into the specified EDW table(s), the PTL and client signatures, and whether the load was successful.

Additionally, the system.upload_info table returns the consistency data and the system.raw_upload_info returns the logical name of the data's source host.

As it begins loading data into the log tables, the EDW inserts a new row in these system tables to track the upload. Initially, the value of the SUCCESSFUL column defaults to false. When the load completes successfully, this column's value changes to true. By default, when you query the data in the log table(s), the EDW returns data only for loads that completed successfully. If you query a log table while it is still receiving data, none of the new data is returned.

IMPORTANT: To view data from a failed load or from an in-progress load, specify the {INCLUDE_BAD_UPLOADS} modifier (enclosed within curly braces) in the FROM clause of your query. If a query and a load occur simultaneously and the query continues after the load completes, you may see all, none, or part of the data from the new load.

For more information, see:

- "system.upload_info", on page 168 and "system.raw_upload_info", on page 169
- Including Rows from Bad Loads in Chapter 10, "Sensage SQL" in the *Reporting Guide*
- "Monitoring for Inconsistent Loads", on page 309

Errors and Return Values

The atload utility returns an exit code of 0 on success, or non-zero exit code and an error condition message upon error. For a full list, see "Errors and Return Values for Data-Store Utilities", on page 133.

Parsing errors may occur if records fail to match the regular expression in the PTL file or if a given field cannot be parsed as the requested data type. You can return these match failures to the client by running atload with the --matchfailures option.

QUERYING DATA

This section describes the `atquery` utility. It contains the following topics:

- "Synopsis", next
- "Description", on page 90
- "Arguments", on page 91
- "Options", on page 91
- "Parameters", on page 95

- “Return Codes And Error Messages”, on page 96
- “Replay Support”, on page 96
- “Scripting Support And Exit Codes”, on page 96
- “Turning .sql Files into Unix Shell Scripts”, on page 97
- “Query Postprocessor”, on page 97

Synopsis

```
atquery [options] <cluster_list> <file_name>.sql [<file_name>.sql […]]

atquery [options] <cluster_list> -
```

NOTE: When you use the second option, in which you enter the query directly from the command line, you must enter CTRL-D on a separate line after the SQL statement to actually run the query.

Description

The `atquery` utility processes:

- Data Manipulation Language (DML) statements, which query (`SELECT`) data in a table and retire (`RETIRE`) data from a table in an EDW instance

  For more information, see Overview of Sensage SQL SELECT Statements in Chapter 10, “Sensage SQL” of the *Reporting Guide* and “Retiring Data”, on page 130.

- Data Definition Language (DDL) statements, which create tables, column filters, and views within an EDW instance

  For more information on the DDL statements that the atquery command can process, see “Defining Data Objects”, on page 141.

The atquery utility supports processing multiple SQL files at once. If you specify the names of SQL files on the command line, they are processed in the order specified. When specifying multiple files, separate each name with a space.

After processing its arguments and checking the SQL files for obvious errors, the querying process proceeds as follows. For each SQL file, atquery:

1 Connects to the specified host and port.

2 Sends the SQL file, wrapped as an XML-RPC "message" (also known as XML data).

3 Listens for errors, progress messages, and result records.

4 On the server, parses the query, sends new queries to each of the hosts in the instance, and combines the results to form the final result.

  NOTE: If the `Primary` directory (which contains the primary copy of log data) is unavailable, the EDW uses the `Secondary` directory (which contains the backup copy of log data). If both of these directories are unavailable, the query fails. If the query uses the `Secondary` directory, processing is slower than when it uses the `Primary`. For more information about these directories, see “EDW Architecture”, on page 24.

INTERNATIONAL SUPPORT FOR QUERYING DATA

The character encoding of <*file_name*>.sql must be UTF-8. ASCII character encoding is a proper subset of UTF-8 character encoding.

For more information on character encoding, see “International Support in the EDW”, on page 159.

Arguments

Argument	Description
`<cluster_list>`	*<cluster_list>* represents a comma-separated list of *<host>:<port>* pairs. The list must be enclosed in quotation marks ("). Specifying a list rather than a single host:port pair allows the utility to run even if the first of the specified EDW nodes is down. Each host:port pair identifies the EDW instance; for example, `edw01:8072`. The command tries each host:port pair in order (left to right) until it finds one that allows a TCP (Transmission Control Protocol) connection. Failure to establish a TCP connection indicates that the EDW host is down. If the connection succeeds but the command fails, the command does not attempt to establish additional connections. Example: `atquery --user=administrator --password=changeme "edw01:8072,edw02:8072,edw03:8072" myfile.sql`
`{ <file_name.sql> [<file_name.sql> […]] | - }`	Specify either one or more files that contain the SQL statements to run (one statement in each file) or specify a dash (-) to read from standard input. **NOTE**: When you use the second option, you are prompted to enter your SQL query. You must enter CTRL-D on a separate line after the SQL statement to actually run the query.

Options

```
--version
```

Displays version of `atquery`.

```
--help, --longhelp
```

Displays online help for command in short form (--help), or with additional detail (--longhelp).

```
--namespace=<namespace>
```

The namespace in which to find the specified table or view. The default namespace is default. To specify the top level, use "" (empty quotation marks).

When you run atquery, you can specify the namespace as a prefix to the table or view name (for example, ns1.ns2.ns3.mytable) or use the --namespace=<*namespace*> flag to specify the namespace.

NOTE:

- When you use the namespace flag, you can specify the namespace once for an entire command; HawkEye AP prepends the specified namespace to each table or view name used in the rest of the request.
- When you run a report from HawkEye AP Console, the console uses the namespace saved in the report definition to supply a value to the --namespace=<*namespace*> flag of the `atquery` utility.

```
--verbose=<n>
```

Where <*n*> represents result set(0), errors(1), warnings(2), progress(3), or everything(4). Default is 2.

Verbose Level	Meaning	Description
	result set only	Displays only the result set of the query. Use the `--format` option to specify the text format of the result set. For more information, see "Scripting Support And Exit Codes", on page 96.
1	error	Displays the result set and only errors; ignores all non-errors
2	warning	Displays the result set and warnings and errors; uses dots (`...`) to indicate progress and exclamation points (`!`) to indicate parsing errors
3	progress	Shows progress in greater detail than level 2
4	everything	Displays all informational messages

```
--override=<name>:<value>
```

Overrides default values for expression macros defined in SQL statements. You can repeat this option to override default values for different expression macros.

For more information, see Expression Macros in Chapter 10, "Sensage SQL" in the *Reporting Guide*.

```
--tableswap=<name>:<newname>
```

Substitutes table names in FROM clauses with another table name. You can repeat this option to substitute different tables in different FROM clauses, such as in sub-SELECTs and subqueries.

When building libraries of queries, it is often helpful to replace the tables in the FROM clause with different tables, chosen at the time the query is run. The --tableswap option enables this functionality, which operates in the same way as the processing directive WITH <*name*> AS TABLE <*newname*>.

For more information, see Table-Name Substitutes in Chapter 10, “Sensage SQL” in the *Reporting Guide*.

-expression=<*SQL_stmt*>
-e=<*SQL_stmt*> (alias)

Executes the given SQL statement. Both of these options allow you to specify an inline SQL statement as a command-line option. You can specify multiple queries this way.

TIP: Alternatively, you can specify a single hyphen at the end of the line preceding the SQL statement, and on a new line, type the `SELECT` statement followed by EOF (`^D`). For example:

```
atquery --user=administrator --pass=pass:s0mep@ss \
localhost:8072 --namespace='system' -
SELECT * FROM users;
```

The statement above returns the same results as:

```
atquery --user=administrator --pass=pass:s0mep@ss localhost:8072   \
--namespace='system' -expression='SELECT * FROM users;'
```

--postproc={<*filename*>|<*perl_code*>}

Optionally post-processes the rows in the final result set of a query. Post-processing logic is written in Perl code, which you supply from a file or the command line. When you provide <*perl_code*>, start the value with an open curly brace ({).

Use postprocessing to change the rows in results sets as they come back from the EDW, before the client application receives them. A postprocessor might do the following:

- Recalculate values in particular column, perhaps through lookup
- Aggregate data returned from the EDW in some special way
- Change the set of columns contained in the result set.

For more information, see “Query Postprocessor”, on page 97.

--format=<*string*>

To format the result rows, <*string*> must be one of the following:

- `max_width` — pipe-delimited that spaces the columns to the maximum-width value
- `tsv` — tab-separated values
- `csv` — comma-separated values
- `psv` — pipe-symbol-separated values
- `dots` — a max_width variant meant for two-column results

The default is `max_width`.

NOTE: Specify only `tsv`, `csv`, or `psv` if you specify `--verbose=0`.

`--timeout=<seconds>`

Specifies how long to let queries run before they time out. The default is `0`, which indicates no timeout.

Automatically cancels a query that is taking too long. This is useful for scripting atquery when you are concerned that a query may "hang" for some reason.

`--[no]raw`

This option forces atquery to print the raw XML load results that the EDW returns, instead of the preformatted results that `atquery` usually shows. This is useful for debugging. The default is *noraw*.

`--[no]dbgprint-request`

Dumps the client-server request, then exits. The default is --nodbgprint-request.

This option forces atquery to print the execution requests that it would have sent to the EDW, and then stops execution rather than proceeding with the query or queries. This is useful for debugging situations and for understanding the EDW API.

`--progress-type=<type>`

For verbose level 3 and above, this option controls what type of progress indicators to return; they include:

- dot — prints only periods, which indicate that the query is running, but does not provide any hints about when it will finish. This indicator is useful when you do not want to see the query details, but want to know that the query is working and has not hung.
- all — the default, this indicator provides instance-wide statistics, which roll up progress indicators sent by each host in the instance.
- unfmt — prints details in a more machine-readable format; for example, suitable for incorporating progress feedback into applications.
- host — prints very verbose per-host details.

The example below illustrates the format of the **all** indicator:

```
- 97.0s elapsed, on select 1/3, pass 2/3, during 1/2, about 32% done  (some at
select 2/3)
-    selecting 1970-01-01 00:00:00 GMT from L0307060456
-   time range 1970-01-01 00:00:00 ------|------------------ 1970-01-01 00:00:03
-      scanned      17M records,   227K rps (  180K rps average)
-   aggregated     591K records,  7,306 rps ( 6,101 rps average) into 6,669 groups
-     RAM free 2,521MB of 4,747MB total (53%), min is perf02: 579MB (36%)
```

`--detailedprogress`

In addition, displays per-query or per-host progress.

`--skip-queues`

Ignores task priority queuing. For more information, see "Task Priority and Queuing", on page 137.

Parameters

Authentication Options

`--user=<username>`

This parameter authenticates the specified *name* as an EDW user. If no user is specified, `atquery` reads the environment variable `ADDAMARK_USER`. If it cannot find this environment variable, it uses the operating system to get the name of the user who is logged in.

NOTE: To log in as the **guest** user, do not specify the `--pass` or `--shared-secret` flags.

`--pass=<pwmethod>`

This parameter provides several methods to specify the password. You can specify the password in clear text, place the password in a file and specify the location of the file, or set it in the `ADDAMARK_PASSWORD` environment variable.

The syntax is:

```
--pass=<password>
--pass=pass:<password>
--pass=file:<filename>
--pass=default
```

When you specify `default` as the password, `atload` looks in the `ADDAMARK_PASSWORD` environment variable. For more information, see "Setting the ADDAMARK_PASSWORD Environment Variable", on page 86.

`--shared-secret=<secret>`

This parameter provides several methods to specify the shared secret. You can specify the secret in clear text, place the secret in a file and specify the location of the file, or set it in the `ADDAMARK_SHARED_SECRET` environment variable. The syntax is:

This parameter provides two methods to specify the shared secret. You can either specify the secret in clear text or place the secret in a file and specify the location of the file. The syntax is:

```
--shared-secret=<secret>
--shared-secret=file:</full_path/filename>
--shared-secret=default
```

When you specify `default` as the password, `atmanage` looks in the `ADDAMARK_SHARED_SECRET` environment variable.

NOTE: The value for `--pass` takes precedence over the value for `--shared-secret`, if both are present.

`--assume-role=role<role_to_be_assumed>`

This parameter causes the operation to be run as if the user had assumed the specified EDW role. The assumed role must have **admin** privileges.

`--skip-queues`

This parameter bypasses normal EDW task queueing. Shared secret authentication is required when you use this parameter.

Return Codes And Error Messages

See "Errors and Return Values for Data-Store Utilities", on page 133.

Replay Support

When diagnosing performance issues, it is often helpful to save the output from the EDW. In particular, `atquery` supports a command (`replay`) that lets you simulate re-running the query that was previously run.

To save the exact bytes sent over the network from the EDW host, run `atquery --raw` and save the results to a file. To replay the results, run `atquery replay <filename>.` Specifying "-" (dash) instead of the filename causes the replay to get data from standard input.

The following example connects the live query to replay:

```
# runs myfile.sql and saves the raw XML messages to savefile.rwt
# and simultaneously visualizes them using "replay"
atquery host:port myfile.sql --raw | tee savefile.rwt | atquery replay -
# replay them (doesn't re-run the query)
atquery replay savefile.rwt
```

NOTE: You will need to `grep` out the "delay" lines to clean up the output.

Scripting Support And Exit Codes

The `atquery` utility returns an exit code of `0` on success, or a non-zero exit code upon error. Use these exit codes to control branching logic in scripts. For example, the following code example controls sending email under various conditions.

```
# these code snippets have been tested with bourne shell (sh), bash,
   #    tcsh and ksh.  They assume that there's an EDW running on port
   #    8072 on the local host, and that there's a table named "syslog"
   #    that has been loaded.
   echo 'select count(*) from syslog during all' > myquery.sql
   echo 'select * from nosuchtable during all' > badquery.sql
   # send an email if the EDW is unavailable, e.g. not running
   atquery nosuchlms:8072 myquery.sql || \
      (mail -s "myquery.sql failed" root < myquery.sql)
   # send an email if there's a syntax error
   atquery localhost:8072 badquery.sql || \
      (mail -s "myquery.sql failed" root < badquery.sql)
   # send an email if/when the query succeeds
   atquery localhost:8072 myquery.sql && \
      (mail -s "myquery.sql succeeded" root < myquery.sql)
   # send an email either way -- sh/bash version
```

```
if ( ./atquery3 localhost:8072 myquery.sql ); then
   mail -s "sql succeeded" asah < /etc/hosts
else
   mail -s "sql failed" asah < /etc/hosts
fi
```

For a full list of errors, see "Errors and Return Values for Data-Store Utilities", on page 133.

In addition to exit codes, the `atquery` utility prints the query result, progress information, and error messages to `stdout`. You control the verbosity of progress information and error messages with the `--verbose` option. The `atquery` utility always prints the result set to `stdout`, regardless of the `--verbose` option.

The results are set off from other text with the following separation lines:

```
==== BEGIN: Results for SQL file
<result_set>
==== END_DATA ====
```

If you specify `--verbose=0`, all messages and separation lines are suppressed, and only the result set is printed. Use the `--format` option to specify the text format of the result set. When `--verbose=0`, valid values for `--format` are `tsv` (tab-separated values), `csv` (comma-separated values), and `psv` (pipe-separated values).

Turning .sql Files into Unix Shell Scripts

For Unix users who would like their SQL files to be runnable, `atquery` knows to ignore the first line of SQL files that start with "`#!`". To support this capability, you must create a wrapper shell script like the following:

```
#!/bin/sh
# this will launch the real atquery program with the given .sql file
# note: please adjust INSTALLNAME, HOST and PORT to reflect your installation
/home/lms/INSTALLNAME/atquery HOST:PORT $*
```

Then, start runnable `.sql` files like the following:

```
#!/usr/local/bin/runatquery
select count(*) from mytable during all
```

NOTE: You must `chmod` both shell scripts, so they become executable; for example, `chmod 755 runatquery`. Also, many shells have a limit on the length of the `#!` line; often, it's limited to 32 characters, and fails silently.

Query Postprocessor

This section describes the client-side `atquery` postprocessor feature. It contains the following topics:

- "What is a Postprocessor?", next
- "Query Postprocessor API", on page 98
- "Query Postprocessor Examples", on page 100

- “Query Postprocessor Sample Script”, on page 105

What is a Postprocessor?

A postprocessor:

- Is a set of functions in Perl.
- Accepts input row/column data returned from the EDW in response to a query.
- Accepts input data that conforms to an "incoming schema" (that is, a set of column names and SQL data types).
- Generates output row/column data, which is the data eventually displayed by the `atquery` program.
- Generates that output data that conforms to an "outgoing schema," which may or may not be the same as the "incoming schema."

A postprocessor is typically used for the two following scenarios:

- To slightly modify one or more column values before return to the client application

  The postprocessor takes incoming rows, makes a slight change to each row, and passes on the values as output rows

- To do specialized aggregation or data formatting

  The postprocessor takes incoming rows and generates outgoing rows that do not directly correspond one-for-one to input rows.

Postprocessors do not chain or cascade. You cannot feed the output of one postprocessor as input to a second postprocessor.

Query Postprocessor API

- “Perl Functions that a Postprocessor Provides”, next
- “Perl Functions that a Postprocessor Calls”, on page 99

Perl Functions that a Postprocessor Provides

The first aspect of the postprocessor API a developer must understand is the set of Perl functions the postprocessor must implement. Some of these functions are optional, as indicated below.

- `postprocInit` — postprocessor initialization [optional function]

  The `atquery` utility calls the `postprocInit` function before any input rows from the EDW results are submitted to the `postprocRow()` function. The `atquery` utility invokes `postprocInit` once before proceeding with results from each individual query, and that a single postprocessor (and its local variables) may be employed for multiple queries.

```
sub postprocInit {
  # The postprocInit function is passed:
  #
  # * a "response" object, representing the response from the EDW --
  #   this "response" handles both the "incoming" and "outgoing" schema and
  #   rows for query postprocessing
  #
  my ($response) = @_;
  # ... Postprocessor initialization code follows -- implementation-specific ...
 }
```

A postprocessor typically uses the initialization function to set up the “outgoing” schema for the postprocessed query, to initialize Perl variables required to perform aggregations, and so on.

- `postprocRow` — a call invoked per row of input arriving from the EDW [required function]

  Do not assume that `postprocRow` is called at least once because, at times, the EDW may return no rows for a given query.

```
sub postprocRow {
  # The postprocRow function is passed:
  #
  # * a "response" object
  # * an "input row" object, which holds the column values for a row of data
  #   incoming from the EDW server
  #
  my ($response, $inputRow) = @_;
  # ... Postprocessor per-row code follows -- implementation-specific ...
 }
```

  A postprocessor typically uses the per-row function to lookup and modify the value of one or more columns for each input row, or to do data aggregation, finding a particular input row's effects on the aggregated values.

- `postprocFinal` — postprocessor finalization [optional function].

  The `atquery` utility calls the `postprocFinal` function after all input rows from the EDW results are submitted to the `postprocRow` function.

```
sub postprocFinal {
  # The postprocFinal function is passed:
  #
  # * a "response" object
  #
  my ($response) = @_;
  # ... Postprocessor finalization code follows -- implementation-specific ...
 }
```

  A postprocessor typically uses the finalization function to put out aggregated values.

  The `atquery` utility does not pass errors directly to any postprocessor function. If a fatal error occurs in transmitting the query to the EDW, while processing the query within the EDW, or while receiving results from the EDW, postprocessing stops and `atquery` handles the error as it would for any other query.

Perl Functions that a Postprocessor Calls

To compute results, a postprocessor can use local Perl variables, subroutines, or other features. At some point, however, the postprocessor must interact with the `$response` and `$inputRow` objects. Those interactions are described below.

- `$response->getIncomingSchema()`

Returns an array containing the column names and SQL data types for the postprocessor's incoming schema. An example return value might be:

```
[ "ts:timestamp", "v1:varchar", "v2:int32", "v3:int64", "v4:bool",
"v5:timestamp" ]
```

NOTE: The returned values can be either in upper or lower case.

- `$response->setMetadata(schema => @outputSchema)`

  Sets the schema for the output displayed by `atquery`.

  Passes in an array of the same form as that returned by the `$response->getIncomingSchema()` function.

- `$response->getSchema()`

  Returns an array containing the column names and SQL data types for the postprocessor's outgoing schema (same format as the `$response->getIncomingSchema() function`).

- `$response->addRowData(rowdata => { "col1" => "value1", "col2" => "value2" })`

  Adds a row of data to the response. The row conforms to the postprocessor's outgoing schema and any number of column names and values can be specified. Any unspecified columns take on the value `""` (empty string), and any extra columns that don't conform to the output schema are ignored.

- `$inputRow->getColumnValue("col1")`

  Returns the value for a particular column of data within the input row.

- `my $hashRef = {}; $response->copyValuesForOutputSchema($inputRow, $hashRef)`

  Convenience function that copies column values from an input row of data to a Perl hash.

  Only those values from the input row whose column names match the outgoing schema are copied to the hash. The hash can subsequently be used (perhaps after further modification) to submit a row of data, with `$response->addRowData(rowdata => $hashRef).`

  NOTE: You are not required to use the function `$response->setMetadata(schema => @outputSchema)` if the outgoing schema matches the incoming schema. The `atquery` utility does this for you implicitly.

Query Postprocessor Examples

- “Example Data Set”, next
- “Simple Lookup Example”, on page 101
- “More complex aggregation example”, on page 102

This topic provides examples of two postprocessor scripts. One example does a simple data lookup to replace values in one column of the incoming data. The other example aggregates and formats data (although the aggregation itself could be done on the EDW, the example uses it for illustrative purposes). The examples cover all postprocessor API functions documented above in Perl Functions that a Postprocessor Calls.

“Query Postprocessor Examples”, on page 100 provides a script that loads the required data and runs the postprocessors. The major steps are outlined below.

Example Data Set

Both postprocessor examples execute against the following data set, which is a simple weblog of presidents shopping for cars:

```
+---------------------------+-----------+-------------------------------------+
|             ts            | user_name |                 url                 |
|        (timestamp)        | (varchar) |              (varchar)              |
+---------------------------+-----------+-------------------------------------*
|2002-02-28T14:23:57.000000Z|gwashington|www.shop.com/home.htm                |
|2002-02-28T14:37:15.000000Z|gwashington|www.shop.com/catalog.htm             |
|2002-02-28T14:39:23.000000Z|gwashington|www.shop.com/prodinfo/infiniti.htm   |
|2002-02-28T14:49:27.000000Z|gwashington|www.shop.com/other_page.htm          |
|2002-02-28T14:29:30.000000Z|jadams     |www.shop.com/home.htm                |
|2002-02-28T14:41:51.000000Z|jadams     |www.shop.com/specials.htm            |
|2002-02-28T14:46:25.000000Z|jadams     |www.shop.com/specials/audi_europe.htm|
|2002-02-28T14:46:29.000000Z|jadams     |www.shop.com/purchase.htm            |
|2002-02-28T14:47:35.000000Z|jadams     |www.shop.com/transaction_complete.htm|
|2002-02-28T14:28:23.000000Z|ztaylor    |www.shop.com/home.htm                |
|2002-02-28T14:44:13.000000Z|ztaylor    |www.shop.com/catalog.htm             |
|2002-02-28T14:46:22.000000Z|ztaylor    |www.shop.com/prodinfo/cadillac.htm   |
|2002-02-28T14:48:31.000000Z|ztaylor    |www.shop.com/purchase.htm            |
|2002-02-28T14:50:01.000000Z|ztaylor    |www.shop.com/catalog.htm             |
+---------------------------+-----------+-------------------------------------+
```

Simple Lookup Example

The URLs are not as descriptive as they could be, so for presentation purposes, the example substitutes the URL with a more descriptive string. The postprocessor script:

- contains only the `postprocRow` function, with no initialization or finalization function
- uses an outgoing schema that is identical to the incoming schema

The postprocessor script shown below uses an internal Perl hash lookup to map URL values to more user friendly page descriptions, and allows for pages not found in the hash.

```
# ---------- FILE:  01_lookup.pproc_in
my %easyUrlMap = (
  "home",                     "Home Page",
  "catalog",                  "Product Catalog",
  "prodinfo/infiniti",        "Car Info Page -- INFINITI",
  "prodinfo/cadillac",        "Car Info Page -- CADILLAC",
  "prodinfo/audi",            "Car Info Page -- AUDI",
  "prodinfo/saturn",          "Car Info Page -- SATURN",
  "specials",                 "Special Offers Central",
  "specials/audi_europe",     "Special Offer--Purchase AUDI in Europe",
  "purchase",                 "CASH REGISTER",
  "transaction_complete",     "-SALE-"
);
sub postprocRow
  my ($response, $inputRow) = @_;
  # Create a copy of the input row, for output
  my $outputRow = {};
  $response->copyValuesForOutputSchema($inputRow, $outputRow);
  # Get the URL, prepare default friendly URL (which isn't very friendly)
  my $rawUrl = $inputRow->getColumnValue("url");
  my $resultUrl = "... $rawUrl ...";
  # Try to find a more friendly URL
  my $url2;
  if (($url2) = $rawUrl =~ /^www.shop.com\/(.*)\.htm$/) {
    my $url3 = $easyUrlMap{$url2};
    if (defined($url3)) {
      $resultUrl = $url3;
    }
  }
  # Replace the URL in the output row
  $outputRow->{"url"} = $resultUrl;
  # Add the modified row to the response
  $response->addRowData(rowdata => $outputRow);
# ---------- END OF FILE:  01_lookup.pproc_in
```

The following command runs the example:

```
   echo "SELECT ts, user_name, url FROM test ORDER BY 2, 1 DURING ALL;" | \
        atquery lmshost:8072 --namespace=myNamespace.pproc_example -
        --postproc=01_lookup.pproc_in
```

The query results are shown below:

```
+---------------------------+-----------+--------------------------------------+
|            ts             | user_name |                   url                |
|        (timestamp)        | (varchar) |                (varchar)             |
+---------------------------+-----------+--------------------------------------+
|2002-02-28T14:23:57.000000Z|gwashington|Home Page                             |
|2002-02-28T14:37:15.000000Z|gwashington|Product Catalog                       |
|2002-02-28T14:39:23.000000Z|gwashington|Car Info Page -- INFINITI             |
|2002-02-28T14:49:27.000000Z|gwashington|... www.shop.com/other_page.htm ...   |
|2002-02-28T14:29:30.000000Z|jadams     |Home Page                             |
|2002-02-28T14:41:51.000000Z|jadams     |Special Offers Central                |
|2002-02-28T14:46:25.000000Z|jadams     |Special Offer--Purchase AUDI in Europe|
|2002-02-28T14:46:29.000000Z|jadams     |CASH REGISTER                         |
|2002-02-28T14:47:35.000000Z|jadams     |-SALE-                                |
|2002-02-28T14:28:23.000000Z|ztaylor    |Home Page                             |
|2002-02-28T14:44:13.000000Z|ztaylor    |Product Catalog                       |
|2002-02-28T14:46:22.000000Z|ztaylor    |Car Info Page -- CADILLAC             |
|2002-02-28T14:48:31.000000Z|ztaylor    |CASH REGISTER                         |
|2002-02-28T14:50:01.000000Z|ztaylor    |Product Catalog                       |
+---------------------------+-----------+--------------------------------------+
```

More complex aggregation example

The second example uses a postprocessor to aggregate some per-page metrics directly from the query results. The metrics are “hits” on any given URL, and “unique users” that saw the given URL. This example also formats the results with more white space than would typically display in query results.

Additionally, this example puts within the query results both the query's incoming schema from the EDW and the output schema the postprocessor defines for the results.

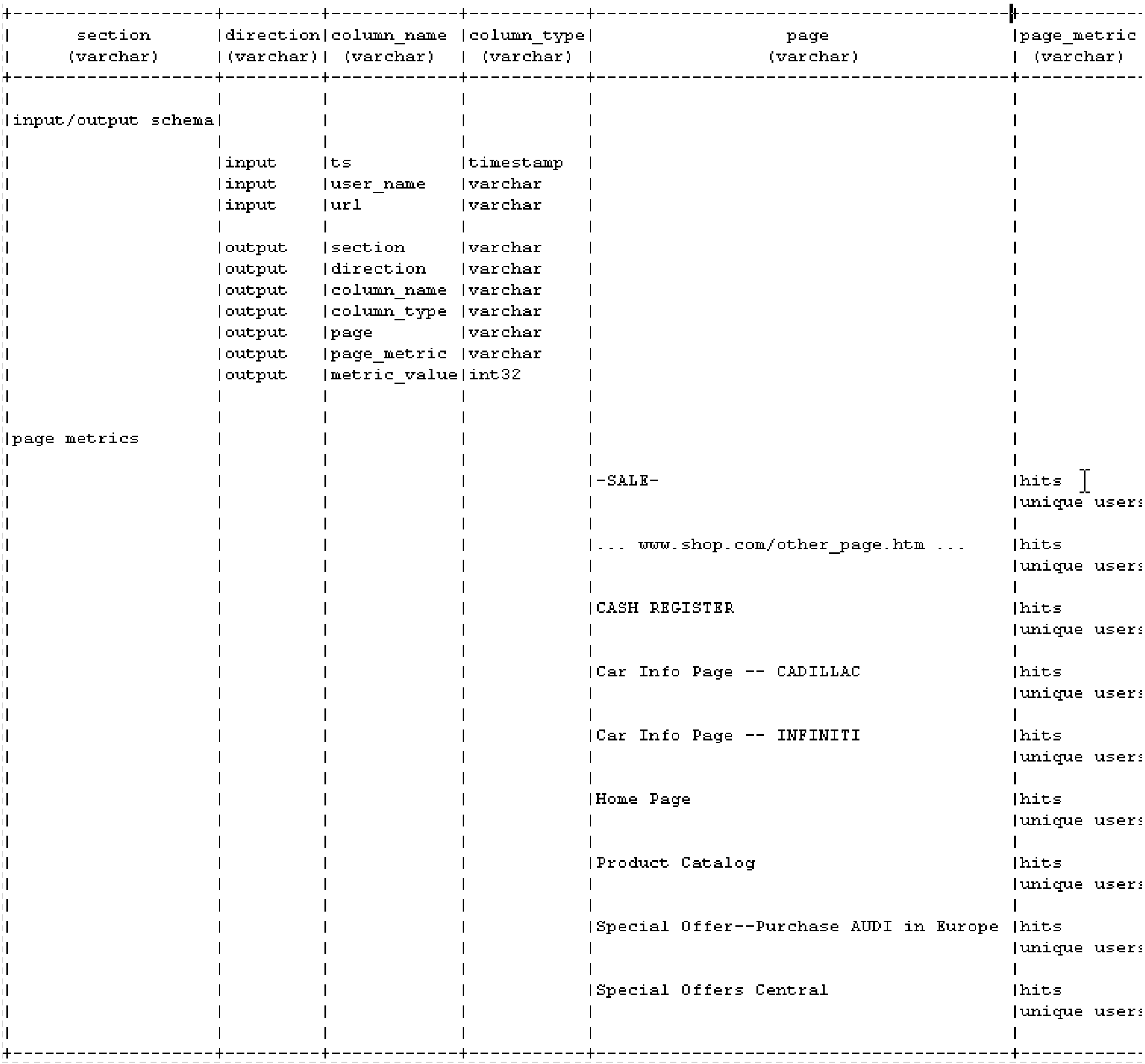
The postprocessor script for this more complex aggregation example:

```
# ---------- FILE:  02_aggregate.pproc_in
my %easyUrlMap = (
  "home",                     "Home Page",
  "catalog",                  "Product Catalog",
  "prodinfo/infiniti",        "Car Info Page -- INFINITI",
  "prodinfo/cadillac",        "Car Info Page -- CADILLAC",
  "prodinfo/audi",            "Car Info Page -- AUDI",
  "prodinfo/saturn",          "Car Info Page -- SATURN",
  "specials",                 "Special Offers Central",
  "specials/audi_europe",     "Special Offer--Purchase AUDI in Europe",
  "purchase",                 "CASH REGISTER",
  "transaction_complete",     "-SALE-"
);
sub mapUrl
  my ($rawUrl) = @_;
  # Prepare default friendly URL (which isn't very friendly)
  my $resultUrl = "... $rawUrl ...";
  # Try to find a more friendly URL
  my $url2;
  if (($url2) = $rawUrl =~ /^www.shop.com\/(.*)\.htm$/) {
    my $url3 = $easyUrlMap{$url2};
    if (defined($url3)) {
      $resultUrl = $url3;
    }
  }
  # Return the mapped URL
  return $resultUrl;
sub pushSchemaInfoIntoTable
  my ($response, $message, $schema) = @_;
  # For each element of the schema, push out the column name/value
  my $schemaElement;
  foreach $schemaElement (@{$schema}) {
    # Find this column's name and type
    my $columnName;
    my $columnType;
    ($columnName, $columnType) = $schemaElement =~ /^(.*):(.*)$/;
    # Prepare a row
    my $rowData = {
      "direction"            => $message,
      "column_name"          => $columnName,
      "column_type"          => $columnType,
    };
    # Push the row
    $response->addRowData(rowdata => $rowData);
  }
my %pageMetrics;
sub postprocInit
  my ($response) = @_;
  # Initialize aggregation metrics -- we MUST do this each time
  # postprocInit is called, because we can have multiple queries
  # running within one 'atquery' session, and we want fresh results
  # each time we run a query.
  %pageMetrics = ();
  # Set the output schema, which will be different from the input schema
  my $outputSchema = [
    "section:varchar",
    "direction:varchar",
    "column_name:varchar",
    "column_type:varchar",
    "page:varchar",
    "page_metric:varchar",
    "metric_value:int32",
  ];
  $response->setMetadata(schema => $outputSchema);
  # To help illustrate input and output schemas, let's print out the
  # input/output schemas in the result set itself
  my $emptyRow = {};
  $response->addRowData(rowdata => $emptyRow);
  my $sectionRow = { section => "input/output schema" };
  $response->addRowData(rowdata => $sectionRow);
  $response->addRowData(rowdata => $emptyRow);
  my @inputSchema = $response->getIncomingSchema();
  my @outputSchema2 = $response->getSchema();
  pushSchemaInfoIntoTable($response, "input", \@inputSchema);
  $response->addRowData(rowdata => $emptyRow);
  pushSchemaInfoIntoTable($response, "output", \@outputSchema2);
  $response->addRowData(rowdata => $emptyRow);
sub postprocRow
  my ($response, $inputRow) = @_;
  # Get relevant fields out of the row
  my $userName = $inputRow->getColumnValue("user_name");
  my $rawUrl = $inputRow->getColumnValue("url");
  # Translate URL to a more friendly value -- we assume
  # the return value is unique
  my $page = &mapUrl($rawUrl);
  # We're doing per-page aggregation, make sure we have space
  # for this page (url)
  if (!defined($pageMetrics{$page})) {
    $pageMetrics{$page} = {
      numHits        => 0,
      uniqueUsers    => {},
    };
  }
  # Now update the per-page aggregates
  $pageMetrics{$page}->{numHits}++;
  $pageMetrics{$page}->{uniqueUsers}->{$userName}++;
sub postprocFinal
  my ($response) = @_;
  # Start a new section
  my $emptyRow = {};
  $response->addRowData(rowdata => $emptyRow);
  my $sectionRow = { section => "page metrics" };
  $response->addRowData(rowdata => $sectionRow);
  $response->addRowData(rowdata => $emptyRow);
  # Present each page and its metrics
  my $page;
  foreach $page (sort(keys(%pageMetrics))) {
    # Add a row for page hits
    my $row = {
      page          => $page,
      page_metric   => "hits",
      metric_value  => $pageMetrics{$page}->{numHits}
    };
    $response->addRowData(rowdata => $row);
    # Add a row for number of unique users
    my @uniqueUsers = keys(%{$pageMetrics{$page}->{uniqueUsers}});
    my $row = {
      page_metric   => "unique users",
      metric_value  => ($#uniqueUsers + 1)
    };
    $response->addRowData(rowdata => $row);
    # Add empty row
    $response->addRowData(rowdata => $emptyRow);
    }
# ---------- END OF FILE:  02_aggregate.pproc_in
```

The following command runs the example:

```
echo "SELECT ts, user_name, url FROM test DURING ALL;" | \
   atquery lmshost:8072 --namespace=myNamespace.pproc_example -
   --postproc=02_aggregate.pproc_in
```

The query results are illustrated below:



```
+-------------------+---------+------------+-----------+----------------------------------------+------------
|      section      |direction|column_name |column_type|                  page                  |page_metric
|     (varchar)     |(varchar)| (varchar)  | (varchar) |               (varchar)                | (varchar)
+-------------------+---------+------------+-----------+----------------------------------------+------------
|                   |         |            |           |                                        |
|input/output schema|         |            |           |                                        |
|                   |         |            |           |                                        |
|                   |input    |ts          |timestamp  |                                        |
|                   |input    |user_name   |varchar    |                                        |
|                   |input    |url         |varchar    |                                        |
|                   |         |            |           |                                        |
|                   |output   |section     |varchar    |                                        |
|                   |output   |direction   |varchar    |                                        |
|                   |output   |column_name |varchar    |                                        |
|                   |output   |column_type |varchar    |                                        |
|                   |output   |page        |varchar    |                                        |
|                   |output   |page_metric |varchar    |                                        |
|                   |output   |metric_value|int32      |                                        |
|                   |         |            |           |                                        |
|                   |         |            |           |                                        |
|page metrics       |         |            |           |                                        |
|                   |         |            |           |                                        |
|                   |         |            |           |-SALE-                                  |hits
|                   |         |            |           |                                        |unique user
|                   |         |            |           |                                        |
|                   |         |            |           |... www.shop.com/other_page.htm ...     |hits
|                   |         |            |           |                                        |unique user
|                   |         |            |           |                                        |
|                   |         |            |           |CASH REGISTER                           |hits
|                   |         |            |           |                                        |unique user
|                   |         |            |           |                                        |
|                   |         |            |           |Car Info Page -- CADILLAC               |hits
|                   |         |            |           |                                        |unique user
|                   |         |            |           |                                        |
|                   |         |            |           |Car Info Page -- INFINITI               |hits
|                   |         |            |           |                                        |unique user
|                   |         |            |           |                                        |
|                   |         |            |           |Home Page                               |hits
|                   |         |            |           |                                        |unique user
|                   |         |            |           |                                        |
|                   |         |            |           |Product Catalog                         |hits
|                   |         |            |           |                                        |unique user
|                   |         |            |           |                                        |
|                   |         |            |           |Special Offer--Purchase AUDI in Europe  |hits
|                   |         |            |           |                                        |unique user
|                   |         |            |           |                                        |
|                   |         |            |           |Special Offers Central                  |hits
|                   |         |            |           |                                        |unique user
|                   |         |            |           |                                        |
+-------------------+---------+------------+-----------+----------------------------------------+------------
```

Query Postprocessor Sample Script

This topic documents a query postprocessor example script that creates and runs multiple postprocessors against an EDW. This file is not itself the postprocessor script, but rather, it does the following:

- loads some data into a table
- queries the un-postprocessed data to verify the information is in the table
- creates a simple postprocessor script that does a simple lookup, and runs that script against the original query
- creates a more complex postprocessor script that does some client-side data aggregation, and runs that script against the original query

For more information, see “Query Postprocessor Examples”, on page 100.

To use the sample query postprocessor script

1 Copy the script file to an executable file on a machine that supports the `sh` or `bash` scripting environment (for example, a Unix machine).

2 Ensure the values of the following variables are set appropriately:

  - `_HOST, LMS_PORT`
  - `LMS_CLIENT_TOOL_DIR`

3 Execute the file against the live EDW server.

4 Examine the client-side files and output returned; refer to the explanation in “Query Postprocessor Examples”, on page 100.

```
#!/bin/bash
#---------------------------------------------------------------------
# pproc_example.bash -- a set of example for Query Postprocessing
#---------------------------------------------------------------------
#---------------------------------------------------------------------
# To run, please set the following values to point to the correct
# EDW instance, and to the directory containing the
# tools 'atload', 'atquery', 'atmanage', and 'atview'.
LMS_HOST=localhost
LMS_PORT=7010
LMS_CLIENT_TOOL_DIR=/home/nwatson/mytools_nfw/eng_nfw/mill/bin
#---------------------------------------------------------------------
# Miscellaneous
SUFFIX="3"
VISIBLE="=+=+=+=+=+=+=+=+=+="
NAMESPACE="myNamespace.pproc_example"
TNAME=test
FULLTNAME="${NAMESPACE}.${TNAME}"
ATVIEW0="${LMS_CLIENT_TOOL_DIR}/atview${SUFFIX}"
ATVIEW="${ATVIEW0} ${LMS_HOST}:${LMS_PORT} --namespace=${NAMESPACE}"
ATMANAGE0="${LMS_CLIENT_TOOL_DIR}/atmanage${SUFFIX}"
ATMANAGE="${ATMANAGE0} ${LMS_HOST}:${LMS_PORT} --namespace=${NAMESPACE}"
ATLOAD0="${LMS_CLIENT_TOOL_DIR}/atload${SUFFIX}"
ATLOAD="${ATLOAD0} ${LMS_HOST}:${LMS_PORT} --namespace=${NAMESPACE}"
ATQUERY0="${LMS_CLIENT_TOOL_DIR}/atquery${SUFFIX}"
ATQUERY="${ATQUERY0} ${LMS_HOST}:${LMS_PORT} --namespace=${NAMESPACE}"
#---------------------------------------------------------------------
# Make sure client-side tools exist
if [ ! -x ${ATVIEW0} -o \
     ! -x ${ATMANAGE0} -o \
     ! -x ${ATLOAD0} -o \
     ! -x ${ATQUERY0} ]; then
  echo
  echo "ERROR:  Client side tools not found!"
  echo
  exit -1
fi
#-----------------------------------------------------------------------
# Make sure EDW (or at least master node) is running
echo "SELECT * FROM properties;" | \
  ${ATQUERY} --namespace=system - > /dev/null 2>/dev/null
if [ $? != 0 ]; then
  echo
  echo "ERROR:  Cannot contact SLS!"
  echo
  exit -1
fi
#-----------------------------------------------------------------------
# Create log data file
echo
echo ${VISIBLE} "Creating log file './logdata.pproc_in'" ${VISIBLE}
/bin/rm -f ./logdata.pproc_in 2>/dev/null
cat > ./logdata.pproc_in << EOF
Feb 28 14:23:57 2002,gwashington,http://www.shop.com/home.htm
Feb 28 14:28:23 2002,ztaylor,http://www.shop.com/home.htm
Feb 28 14:37:15 2002,gwashington,http://www.shop.com/catalog.htm
Feb 28 14:29:30 2002,jadams,http://www.shop.com/home.htm
Feb 28 14:39:23 2002,gwashington,http://www.shop.com/prodinfo/infiniti.htm
Feb 28 14:41:51 2002,jadams,http://www.shop.com/specials.htm
Feb 28 14:44:13 2002,ztaylor,http://www.shop.com/catalog.htm
Feb 28 14:46:22 2002,ztaylor,http://www.shop.com/prodinfo/cadillac.htm
Feb 28 14:46:25 2002,jadams,http://www.shop.com/specials/audi_europe.htm
Feb 28 14:46:29 2002,jadams,http://www.shop.com/purchase.htm
Feb 28 14:47:35 2002,jadams,http://www.shop.com/transaction_complete.htm
Feb 28 14:48:31 2002,ztaylor,http://www.shop.com/purchase.htm
Feb 28 14:49:27 2002,gwashington,http://www.shop.com/other_page.htm
Feb 28 14:50:01 2002,ztaylor,http://www.shop.com/catalog.htm
EOF
#-----------------------------------------------------------------------
# Create PTL script file
echo
echo ${VISIBLE} "Creating PTL file './ptl.pproc_in'" ${VISIBLE}
/bin/rm -f ./ptl.pproc_in 2>/dev/null
cat > ./ptl.pproc_in << EOF
^([^,]*),([^,]*),\s*http://([^,]*)$
TimeStr:VARCHAR,UserName:VARCHAR,Url:VARCHAR
SELECT _timestamp(TimeStr) as ts,
       UserName as user_name,
       Url as url
FROM stdin;
EOF
#-----------------------------------------------------------------------
# Drop example table if present
echo
echo ${VISIBLE} "Dropping table ${FULLTNAME}" ${VISIBLE}
${ATMANAGE} droptbl ${TNAME} > /dev/null 2>/dev/null
#-----------------------------------------------------------------------
# Load data into table
echo
```

```
echo ${VISIBLE} "Loading data into table ${FULLTNAME}" ${VISIBLE}
echo
${ATLOAD} ${TNAME} ./ptl.pproc_in ./logdata.pproc_in
#---------------------------------------------------------------------
# Pull all the raw data
echo
echo ${VISIBLE} "Querying raw data in table ${FULLTNAME}" ${VISIBLE}
echo
echo "SELECT ts, user_name, url FROM ${TNAME} ORDER BY 2, 1 DURING ALL;" |
${ATQUERY} -
#---------------------------------------------------------------------
# Use a simple postproc to replace URL values with more "user-friendly" values
cat > 01_lookup.pproc_in << EOF
# ---------- FILE:  01_lookup.pproc_in
my %easyUrlMap = (
  "home",                     "Home Page",
  "catalog",                  "Product Catalog",
  "prodinfo/infiniti",        "Car Info Page -- INFINITI",
  "prodinfo/cadillac",        "Car Info Page -- CADILLAC",
  "prodinfo/audi",            "Car Info Page -- AUDI",
  "prodinfo/saturn",          "Car Info Page -- SATURN",
  "specials",                 "Special Offers Central",
  "specials/audi_europe",  "Special Offer -- Purchase AUDI in Europe",
  "purchase",                 "CASH REGISTER",
  "transaction_complete",     "-SALE-"
);
sub postprocRow
  my (\$response, \$inputRow) = @_;
  # Create a copy of the input row, for output
  my \$outputRow = {};
  \$response->copyValuesForOutputSchema(\$inputRow, \$outputRow);
  # Get the URL, prepare default friendly URL (which isn't very friendly)
  my \$rawUrl = \$inputRow->getColumnValue("url");
  my \$resultUrl = "... \$rawUrl ...";
  # Try to find a more friendly URL
  my \$url2;
  if ((\$url2) = \$rawUrl =~ /^www.shop.com\\/(.*)\\.htm$/) {
    my \$url3 = \$easyUrlMap{\$url2};
    if (defined(\$url3)) {
      \$resultUrl = \$url3;
    }
  }
  # Replace the URL in the output row
  \$outputRow->{"url"} = \$resultUrl;
  # Add the modified row to the response
  \$response->addRowData(rowdata => \$outputRow);
# ---------- END OF FILE:  01_lookup.pproc_in
EOF
echo
echo ${VISIBLE} "Querying data in table ${FULLTNAME}, replacing URLs" ${VISIBLE}
echo
echo "SELECT ts, user_name, url FROM ${TNAME} ORDER BY 2, 1 DURING ALL;" | \
  ${ATQUERY} --postproc=01_lookup.pproc_in -
#---------------------------------------------------------------------
# Use a more complex postproc to do client-side aggregation/presentation
cat > 02_aggregate.pproc_in << EOF
# ---------- FILE:  02_aggregate.pproc_in
my %easyUrlMap = (
  "home",                     "Home Page",
  "catalog",                    "Product Catalog",
  "prodinfo/infiniti",          "Car Info Page -- INFINITI",
  "prodinfo/cadillac",          "Car Info Page -- CADILLAC",
  "prodinfo/audi",              "Car Info Page -- AUDI",
  "prodinfo/saturn",            "Car Info Page -- SATURN",
  "specials",                   "Special Offers Central",
  "specials/audi_europe",  "Special Offer -- Purchase AUDI in Europe",
  "purchase",                   "CASH REGISTER",
  "transaction_complete",       "-SALE-"
);
sub mapUrl
  my (\$rawUrl) = @_;
  # Prepare default friendly URL (which isn't very friendly)
  my \$resultUrl = "... \$rawUrl ...";
  # Try to find a more friendly URL
  my \$url2;
  if ((\$url2) = \$rawUrl =~ /^www.shop.com\\/(.*)\\.htm$/) {
    my \$url3 = \$easyUrlMap{\$url2};
    if (defined(\$url3)) {
      \$resultUrl = \$url3;
    }
  }
  # Return the mapped URL
  return \$resultUrl;
sub pushSchemaInfoIntoTable
  my (\$response, \$message, \$schema) = @_;
  # For each element of the schema, push out the column name/value
  my \$schemaElement;
  foreach \$schemaElement (@{\$schema}) {
    # Find this column's name and type
    my \$columnName;
    my \$columnType;
    (\$columnName, \$columnType) = \$schemaElement =~ /^(.*):(.*)$/;
    # Prepare a row
    my \$rowData = {
      "direction"           => \$message,
      "column_name"         => \$columnName,
      "column_type"         => \$columnType,
    };
    # Push the row
    \$response->addRowData(rowdata => \$rowData);
  }
my %pageMetrics;
sub postprocInit
  my (\$response) = @_;
  # Initialize aggregation metrics -- we MUST do this each time
  # postprocInit is called, because we can have multiple queries
  # running within one 'atquery' session, and we want fresh results
  # each time we run a query.
  %pageMetrics = ();
  # Set the output schema, which will be different from the input schema
  my \$outputSchema = [
    "section:varchar",
    "direction:varchar",
    "column_name:varchar",
    "column_type:varchar",
    "page:varchar",
    "page_metric:varchar",
    "metric_value:int32",
  ];
  \$response->setMetadata(schema => \$outputSchema);
  # To help illustrate input and output schemas, let's print out the
  # input/output schemas in the result set itself
  my \$emptyRow = {};
  \$response->addRowData(rowdata => \$emptyRow);
  my \$sectionRow = { section => "input/output schema" };
  \$response->addRowData(rowdata => \$sectionRow);
  \$response->addRowData(rowdata => \$emptyRow);
  my @inputSchema = \$response->getIncomingSchema();
  my @outputSchema2 = \$response->getSchema();
  pushSchemaInfoIntoTable(\$response, "input", \\@inputSchema);
  \$response->addRowData(rowdata => \$emptyRow);
  pushSchemaInfoIntoTable(\$response, "output", \\@outputSchema2);
  \$response->addRowData(rowdata => \$emptyRow);
sub postprocRow
  my (\$response, \$inputRow) = @_;
  # Get relevant fields out of the row
  my \$userName = \$inputRow->getColumnValue("user_name");
  my \$rawUrl = \$inputRow->getColumnValue("url");
  # Translate URL to a more friendly value -- we assume
  # the return value is unique
  my \$page = &mapUrl(\$rawUrl);
  # We're doing per-page aggregation, make sure we have space
  # for this page (url)
  if (!defined(\$pageMetrics{\$page})) {
    \$pageMetrics{\$page} = {
      numHits        => 0,
      uniqueUsers    => {},
    };
  }
  # Now update the per-page aggregates
  \$pageMetrics{\$page}->{numHits}++;
  \$pageMetrics{\$page}->{uniqueUsers}->{\$userName}++;
sub postprocFinal
  my (\$response) = @_;
  # Start a new section
  my \$emptyRow = {};
  \$response->addRowData(rowdata => \$emptyRow);
  my \$sectionRow = { section => "page metrics" };
  \$response->addRowData(rowdata => \$sectionRow);
  \$response->addRowData(rowdata => \$emptyRow);
  # Present each page and its metrics
  my \$page;
  foreach \$page (sort(keys(%pageMetrics))) {
    # Add a row for page hits
    my \$row = {
      page          => \$page,
      page_metric   => "hits",
      metric_value  => \$pageMetrics{\$page}->{numHits}
    };
    \$response->addRowData(rowdata => \$row);
    # Add a row for number of unique users
    my @uniqueUsers = keys(%{\$pageMetrics{\$page}->{uniqueUsers}});
    my \$row = {
      page_metric   => "unique users",
      metric_value  => (\$#uniqueUsers + 1)
    };
    \$response->addRowData(rowdata => \$row);
```

```
        # Add empty row
        \$response->addRowData(rowdata => \$emptyRow);
      }
    # ---------- END OF FILE:  02_aggregate.pproc_in
    EOF
    echo
    echo ${VISIBLE} "Querying data in table ${FULLTNAME}, aggregating data"
    ${VISIBLE}
    echo
    echo "SELECT ts, user_name, url FROM ${TNAME} DURING ALL;" | \
      ${ATQUERY} --postproc=02_aggregate.pproc_in -
    #-----------------------------------------------------------------------
    # Done
    echo
    exit 0
    #----------
    # End of file 'pproc_example.bash'
    #----------
```

Managing an EDW Data Store

This section describes the `atmanage` utility. It contains the following topics:

- "Synopsis", next
- "Description", on page 111
- "Options", on page 111
- "Arguments", on page 113
- "Authentication Management", on page 116
- "Table Management", on page 120

Synopsis

```
atmanage [options] <cluster_list> <action>
```

Description

The `atmanage` utility enables management of various aspects of the EDW, including managing users, roles, and permissions; creating, deleting, and renaming tables; and stopping tasks.

Because `atmanage` requests may alter the state of the EDW, use this utility carefully.

To examine the current state of the EDW, use the `atview` command.

Options

Information About atmanage

```
--version
```

Displays version of `atmanage`.

`--help, --longhelp`

Displays online help in short form (`--help`), or with additional detail (`--longhelp`).

Specify a Namespace

`--namespace=<name>`

The table namespace in which to manage items. The default namespace is `default`. To specify the top level, use `""` (empty quotation marks).

NOTE: When you run `atmanage`, you can specify a table name explicitly (for example, `ns1.ns2.ns3.mytable`) or use the `--namespace=<namespace>` flag to specify the namespace. When you use the namespace flag, you can specify the namespace once for an entire command; HawkEye AP prepends the specified namespace to each table name used in the rest of the request.

Authentication Options

`--user=<username>`

This parameter authenticates the specified *name* as an EDW user. If no user is specified, `atmanage` reads the environment variable `ADDAMARK_USER`. If it cannot find this environment variable, `atmanage` uses the operating system to get the name of the user who is logged in.

NOTE: To log in as the `guest` user, do not specify the `--pass` or `--shared-secret` flags.

`--pass=<pwmethod>`

This parameter provides several methods to specify the password. You can specify the password in clear text, place the password in a file and specify the location of the file, or set it in the `ADDAMARK_PASSWORD` environment variable. The syntax is:

```
--pass=<password>
--pass=pass:<password>
--pass=file:<filename>
--pass=default
```

When you specify `default` as the password, `atmanage` looks in the `ADDAMARK_PASSWORD` environment variable. For more information, see “Setting the ADDAMARK_PASSWORD Environment Variable”, on page 86.

`--shared-secret=<secret>`

This parameter provides several methods to specify the shared secret. You can specify the secret in clear text, place the secret in a file and specify the location of the file, or set it in the `ADDAMARK_SHARED_SECRET` environment variable. The syntax is:

This parameter provides two methods to specify the shared secret. You can either specify the secret in clear text or place the secret in a file and specify the location of the file. The syntax is:

```
--shared-secret=<secret>
--shared-secret=file:</full_path/filename>
```

```
--shared-secret=default
```

When you specify `default` as the password, `atmanage` looks in the `ADDAMARK_SHARED_SECRET` environment variable.

NOTE: The value for `--pass` takes precedence over the value for `--shared-secret`, if both are present.

`--assume-role=role<`*`role_to_be_assumed`*`>`

This parameter causes the operation to be run as if the user had assumed the specified EDW role. The assumed role must have `admin` privileges. For more information about managing users and roles, see "Authentication Management", on page 116.

Debugging

```
--[no]dbgprint-request
```

Dumps client-server request, and then exits. The default is `--nodbgprint-request`.

This option forces `atmanage` to print the execution requests that it would have sent to the EDW, and then stops execution rather than proceeding with the action. This is useful for debugging situations and for understanding the EDW API.

```
--[no]raw
```

This options forces `atmanage` to print the raw XML results that the EDW returns, instead of the interpreted results `atmanage` usually shows. The default is `--noraw`.

```
--[no]show-responses
```

When reading the response from the EDW server, `atmanage` filters out certain messages, reformats others, and attempts to present a simpler response suitable for most cases. The `--show-responses` flag forces `atmanage` to show the uninterpreted responses from the EDW in addition to normal output. The default is `--noshow-responses`.

Arguments

The `atmanage` utility takes as its first argument the *`<cluster_list>`* for the EDW that is to be managed. The *`<cluster_list>`* represents a comma-separated list of *`<host>:<port>`* pairs. The list must be enclosed in quotation marks (`"`).

Specifying a list rather than a single host:port pair allows the utility to run even if the first of the specified EDW nodes is down. Each host:port pair identifies the EDW instance; for example, `edw01:8072`.

The command tries each host:port pair in order (left to right) until it finds one that allows a TCP (Transmission Control Protocol) connection. Failure to establish a TCP connection indicates that the EDW host is down. If the connection succeeds but the command fails, the command does not attempt to establish additional connections.

Example:

```
atmanage --user=administrator --password=changeme
   "edw01:8072,edw02:8072,edw03:8072" adduser joe_cool
```

The command then expects an *<action>* argument, which indicates how the program should act on the specified EDW. Depending on the *<action>*, there are a number of additional arguments, as indicated in the table below.

IMPORTANT: These commands must be submitted by a user with administrator permission.

Action/Admin Action	Additional Arguments	Description
Authentication Management		
adduser	*<username> <password>*	Add a user
deleteuser	*<username>*	Delete a user
changepassword	*<username> <new_password>*	Add or change a user password
addrole	*<rolename>*	Add a role
deleterole	*<rolename>*	Delete a role
addpermission	*<permission_name> <type> <admin_value>*	Create a permission that can later be associated with a role; *<type>* is either 'token' or 'property'
deletepermission	*<permission_name>*	Delete a permission object
addrolepermission	*<rolename> <permission_name> <value>*	Associate a permission with a role
	<rolename> sls.namespace <namespace>	Grant a role permission to access a namespace
deleterolepermission	*<rolename> <permission_name> <value>*	Dissociate a permission from a role
	<rolename> sls.namespace *<namespace>*	Revoke a role's permission to access a namespace
adduserrole	*<username> <rolename>*	Associate a user with a role
deleteuserrole	*<username> <rolename>*	Dissociate a user from a role
setuserstate	*<username>* [enable|disable]	Enable or disable a user
setrolestate	*<rolename>* [enable|disable]	Enable or disable a role
Table management		
createtbl	*<table_name> <column1>*:*<type>* [*<column2>*:*<type>*[...]]	Create a table with the given columns **NOTE**: This option enables creation of tables but not views.
renametbl	*<old_tablename> <new_tablename>*	Rename a table and/or move it between namespaces **NOTE**: This option enables renaming and moving tables but not views.
droptbl	*<table_name>*	Delete a table **NOTE**: This option enables deletion of tables but not views.

Action/Admin Action	Additional Arguments	Description
quiesce		Set the data store to read-only mode
resume		Restore the data store to its nominal mode
backuptbl	start <*table_name*>	Backup a table
	commit <*table_name*>	Commit the backup
	rollback <*table_name*>	Rollback a table backup to its original state
restoretbl	start <*table_name*>	Create a sandbox into which you restore a previously backed up table
	commit <*table_name*>	Commit the sandbox into the data store
	rollback <*table_name*>	Rollback a table restore to its original state
archivetbl	<*table_name*> <nsi> [*end_time*] [*start_time*]	Archive the requested (ISO) time range from the requests table to the specified NSI
setnsi	[*nsi*] [*nsa*]	Add or change NSI/NSA association
deletensi	[*nsi*]	Delete an NSI/NSA association
listnsi		List NSI/NSA associations
updateuploadinfo	<*table_name*> <*CSV_file*>	Update upload information for the table from CSV file
adduploadinfo	<*table_name*> <*CSV_file*>	Add upload information for the table from CSV file
deleteuploadinfo	<*table_name*> <*CSV_file*>	Delete upload information for associated IDs
canceltask	<*request_id*>	Kill the given task (auto-detects internal task ID rather than application-defined task ID).
compacttbl	<*table_name*> [<*seconds_to_run*>] [<*enable_compact_archived_data*>]	Compact all fragmented portions of a table. Optionally, can facilitate scheduled compaction in the Loader by specifying: • number of seconds to run; compacting stops when the specified number is exceeded. • whether to compact archived data For more information, see: • “compacttbl”, on page 124 • Defining Loaders in Chapter 3, “Collector Configuration” in the *Event Collection Guide*

Action/Admin Action	Additional Arguments	Description
`force-upgrade`	*`<table_name>`*	Manually upgrade the data store Upgrades the table's storage format to the latest version. **IMPORTANT**: may be very slow, and may cause the table to be unreadable by previous versions of the software. Consult Hexis Cyber Solutions Technical Support before using.

Authentication Management

This topic describes utilities provided by the EDW that enable administrators to manage users, roles, and passwords. For conceptual information, see "Users, Roles, and Permissions", on page 218.

IMPORTANT:

- These commands must be submitted by a user with administrator permission.
- These commands affect only the EDW instance in which they are run; in other words, adding a user to one instance does not add the user to other instances.

This topic describes the following utilities:

- "adduser", next
- "deleteuser", on page 117
- "changepassword", on page 117
- "addrole", on page 117
- "deleterole", on page 117
- "addpermission", on page 117
- "deletepermission", on page 118
- "addrolepermission", on page 118
- "deleterolepermission", on page 118
- "adduserrole", on page 119
- "deleteuserrole", on page 119
- "setuserstate", on page 119
- "setrolestate", on page 119

adduser

```
adduser <username>
```

Adds a new user account to the EDW instance.

The *`username`* argument is a text string that identifies the user. The EDW does not distinguish between uppercase and lowercase letters in the name. Therefore, "JoeCool" is the same as "joecool".

Example:

```
atmanage --user=administrator --pass=pass:s0mep@ss localhost:8072 \
adduser joe_cool
```

deleteuser

```
deleteuser <username>
```

Deletes a user from the EDW instance.

Example:

```
atmanage --user=administrator --pass=pass:s0mep@ss localhost:8072 \
deleteuser joe_cool
```

changepassword

```
changepassword <username> <new_password>
```

Adds or changes a user's password.

Example:

```
atmanage --user=administrator --pass=pass:s0mep@ss localhost:8072 \
changepassword joe_cool j03p@ss
```

addrole

```
addrole <rolename>
```

Adds a role to the EDW instance.

The *rolename* argument is a text string that identifies the role. You use roles to grant specific permissions, such as limiting access to a single namespace. You can associate multiple users and permissions to the role. Examples of a *rolename* are: `analyst`, `security`, and `guest`.

Example:

```
atmanage --user=administrator --pass=pass:s0mep@ss localhost:8072 addrole sls
```

deleterole

```
deleterole <rolename>
```

Deletes a role from the EDW instance, and disassociates all user accounts from that role.

Example:

```
atmanage --user=administrator --pass=pass:s0mep@ss localhost:8072 deleterole sls
```

addpermission

```
addpermission <permission_name> <type> <admin_value>
```

Creates a permission to be associated with a role or roles.

deletepermission

```
deletepermission <name>
```

Deletes a permission object and removes it from all the roles that referenced it.

Example:

```
atmanage --user=administrator --pass=pass:s0mep@ss host03:7979 \
deletepermission securityperm
```

addrolepermission

```
addrolepermission <rolename> {<permission_name> <value>}|{sls.namespace
<namespace>}
```

- Associate a Permission with a Role:
  - The *`rolename`* argument was created using "`addrole <rolename>`".
  - The *`permission_name`* argument was created using "`addpermission <permission_name> ...`".
  - The *`value`* argument is the token or property value specified by the addpermission "type" attribute, created using "`addpermission <permission_name> <type>...`"

  NOTE: All user accounts (as created using "`adduser <username>`") that are associated with a role inherit its permissions.

  Example:

  ```
  atmanage --user=administrator --pass=pass:s0mep@ss localhost:8072 \
  addrolepermission analyst sls.select true
  ```

- Grant a Role Permission to Access a Namespace:
  - The *`rolename`* argument was created using "`addrole <rolename>`".
  - The *`namespace`* argument identifies the namespace to which the role is being granted access permission.

    Example:

    ```
    atmanage --user=administrator --pass=pass:s0mep@ss localhost:8072 \
    addrolepermission analyst sls.namespace finance
    ```

deleterolepermission

```
deleterolepermission <rolename> {<permission_name> <value>}|{sls.namespace
<namespace>}
```

- Revoke a Permission from a Role:

  Removes the specified permission from the role, which removes the permission from all user accounts associated with the role.

Example:

```
atmanage --user=administrator --pass=pass:s0mep@ss localhost:8072 \
deleterolepermission analyst sls.select true
```

- Revoke a Role Permission to Access a Namespace:

  Example:

```
atmanage --user=administrator --pass=pass:s0mep@ss localhost:8072 \
deleterolepermission analyst sls.namespace finance
```

adduserrole

```
adduserrole <username> <rolename>
```

Associates a user with a role.

- The *username* argument is the user account created using "`adduser <username>`".
- The *rolename* argument is the role created using "`addrole <rolename>`".

Example:

```
atmanage --user=administrator --pass=pass:s0mep@ss host03:7979 \
vijay analyst
```

deleteuserrole

```
deleteuserrole <username> <rolename>
```

Dissociates a user from a role.

Example:

```
atmanage --user=administrator --pass=pass:s0mep@ss host03.myco.com:7979 \
deleteuserrole vijay analyst
```

setuserstate

```
setuserstate <username> [enable|disable]
```

Enables or disables a user.

NOTE: For information on permanently deleting a user account, see “deleteuser”, on page 117.

Example:

```
atmanage --user=administrator --pass=pass:s0mep@ss localhost:8072 \
setuserstate vijay enable
```

setrolestate

```
setrolestate <rolename> [enable|disable]
```

Enables or disables a role.

NOTE: For information on permanently deleting a role, see "deleterole", on page 117.

Example:

```
atmanage --user=administrator --pass=pass:s0mep@ss localhost:8072 \
setrolestate analyst enable
```

Table Management

This topic describes utilities provided by the EDW that enable administrators to manage tables and other database objects.

IMPORTANT:

- These commands must be submitted by a user with administrator permission.
- These commands affect only the EDW instance in which they are run; in other words, creating a table in one instance does not create the table in other instances.

This topic describes the following utilities:

- "createtbl", on page 120
- "renametbl", on page 121
- "droptbl", on page 121
- "backuptbl", on page 122
- "restoretbl", on page 122
- "canceltask", on page 122
- "compacttbl", on page 124
- "force-upgrade", on page 124

createtbl

```
createtbl <table_name> <column1>:<type> [<column2>:<type> [...]]
```

The atload utility automatically creates a table at load time if the specified table does not exist. However, there are times you want to create an empty table in a given namespace, with a given name and set of columns. The atmanage utility provides the createtbl action for this purpose. For example, the following command creates a table named quotes in the external namespace and defines six columns:

```
atmanage --namespace=external edwhost:8072 createtbl quotes \
ts:TIMESTAMP col1:VARCHAR col2:INT32 col3:INT64 col4:TIMESTAMP col5:FLOAT
```

NOTE: When you create a table, keep the following in mind:

- You must always define a column named ts, with SQL data type TIMESTAMP.

- The EDW creates automatically an extra column named _uploadid, with SQL data type varchar. The column holds the ID of the upload operation that inserted each row into the table.
- You can define other columns.

renametbl

```
renametbl <old_tablename> <new_table_name>
```

The renametbl action allows you to:

- correct spelling mistakes
- roll out ("swap in") new data sets
- establish or change namespaces
- establish table-naming conventions.

The following example moves the quotes table from the external namespace to the external.websvcs namespace, and changes the table's name from quotes to stock_quote_log:

```
atmanage edwhost:8072 --namespace=external renametbl quotes \
  websvcs.stock_quote_log
```

The following example illustrates two commands. The first creates a table in the default namespace (default). The second moves the table from the default namespace to a namespace named new_namespace:

```
atmanage edwhost:8072 createtbl my_table ts:TIMESTAMP val:VARCHAR
atmanage edwhost:8072 --namespace '' renametbl \
  myNamespace.my_table new_namespace.my_table
```

Before specifying the old and new table names, the example above uses an empty string to identify the top namespace. If this flag were missing from the command, the EDW would assume the old table was myNamespace.myNamespace.my_table and would rename the table to myNamespace.new_namespace.my_table.

droptbl

```
droptbl <table_name>
```

The droptbl action allows you to delete a specified table if none of its data is archived to a nearline storage device. If any of its data has been archived, use the retire command to remove the non-archived data. Only after all of its data has been removed from the nearline storage device can you drop the table. For more information, see "Retiring Data", on page 130.

IMPORTANT:

- Because typical EDW tables are gigantic in size, no undo is possible. Therefore, Hexis Cyber Solutions recommends that you use this action with extreme caution and that you back up your tables first.
- When you issue `droptbl` against a table that has any data under retention on a nearline storage device, the `droptbl` command fails. Its error message provides the date in the future

when the nearline storage device will retire the data and advises you to use `retire` to remove local and non-retained data.

- When you issue droptbl against a table that has data archived to a nearline storage device but the data is not under retention or the retention date has past, the droptbl command fails. Its error message informs you to use retire to remove the archived data before running droptbl.

For more information, see Installing Perl Modules in Chapter 12, “Perl Subroutines” in the Reporting Guide and “Retiring Data”, on page 130.

The following example deletes the table created and renamed in renametbl above:

```
atmanage edwhost:8072 --namespace=external.websvcs droptbl stock_quote_log
```

backuptbl

```
backuptbl [start|commit|rollback] <table_name>
```

The backuptbl action allows you to back up a table. You must use this action in conjunction with a standard backup tool and the atquery utility. For more information, see “Backing Up and Restoring an EDW Table”, on page 139.

This action has three arguments:

- `start`—begins the backup session
- `commit`—commits the changes
- `rollback`—returns the table data to its pre-backup state

restoretbl

```
restoretbl [start|commit|rollback] <table_name>
```

The restoretbl action allows you to restore a table from a backup. You must use this action in conjunction with a standard backup tool. For more information, see “Backing Up and Restoring an EDW Table”, on page 139.

This action has three arguments:

- start—creates a sandbox directory to which the table can be restored; you can use the sandbox to double-check the data before you run a restore on your EDW instance's data store
- commit—commits the sandbox back into the data store
- rollback—returns the table data to its pre-restored state

IMPORTANT: You cannot use the restoretbl option to upgrade a datastore from an older version of HawkEye AP to a newer version. You can only use `restoretbl` to restore tables created using the same version of HawkEye AP.

canceltask

```
canceltask <request_id>
```

The `canceltask` action allows you to cancel a task currently running on the EDW.

Request IDs

Client programs such as atview, atmanage, atquery, and atload interact with the EDW to send requests for various actions (for example, requests to load data into tables or query the data in tables), and wait for results. Requests have an associated request ID, which the client program defines and prints. For example, atload might print the following information in response to running a load:

```
request id: [app=atload,u=harry,h=perf02.hq.myco.com,id=e4f5e41f]
```

To cancel the request's operation in the EDW, take the value that follows id= above and use it in a command like the one below:

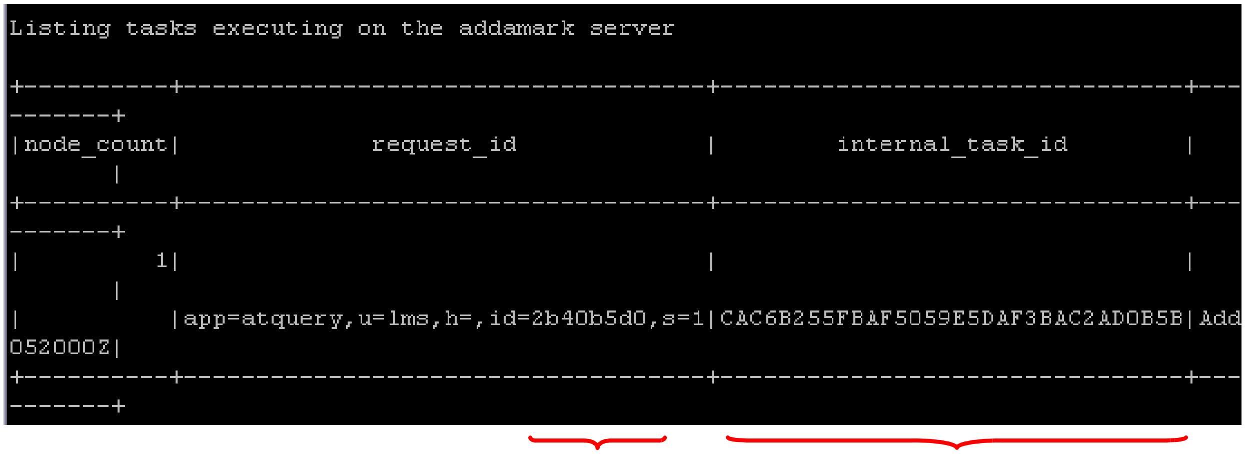
```
atmanage edwhost:8072 canceltask e4f5e41f"
```

Killing a task is generally safe. However, Hexis Cyber Solutions recommends that, if possible, you use the client-side utility that emits the request to perform the kill. For example, press CTRL-C during a load with atload. The atmanage canceltask command was designed for system administrators who detect "runaway" jobs submitted by remote users.

Task IDs

A client program may not define an appropriate, unique, and readily usable request ID for the requests it processes. For any request running on the EDW, there is always a unique task ID that the EDW server itself defines. Each request sent to the EDW can involve multiple processes that run on multiple hosts in the instance. The EDW treats all these processes as a single "task".

If a client program does not define a usable request ID, an administrator can use the atmanage utility to cancel a task by specifying this task ID. This option ensures that, no matter what shortcomings a client program may have, there is always a way to cancel a running task. To find the internal task IDs, run "atview tasks" to list the tasks. The graphic below illustrates running this command and its output.



```
Listing tasks executing on the addamark server

+----------+------------------------------------+--------------------------------+---
-------+
|node_count|             request_id             |        internal_task_id        |
       |
+----------+------------------------------------+--------------------------------+---
-------+
|         1|                                    |                                |
       |
|          |app=atquery,u=lms,h=,id=2b40b5d0,s=1|CAC6B255FBAF5059E5DAF3BAC2AD0B5B|Add
052000Z|
+----------+------------------------------------+--------------------------------+---
-------+
```

Request ID *Task ID*

Locate the internal_task_id for the desired request and specify its value as the *`request_id`* when you run `canceltask`.

For example:

```
   atmanage edwhost:8072 canceltask CAC6B255FBAF505935DAF3BAC2AD0B5B
```

NOTE:

- Because the "atview tasks" command illustrated above returns the request ID as well as the task ID, you can use either value to cancel the task.
- You can use the "atview tasks" command to obtain the request ID and task ID of tasks run from the HawkEye AP Console.

compacttbl

```
compacttbl <table_name> [<num_seconds>] [<enable_compact_archived_data>]
```

By default, the EDW loads new data into separately indexed fragments to avoid the overhead associated with combining data into more efficient files for long-term archival. If you have many loads in the same time range (especially small loads under 10,000 records), query performance and/or compression can suffer.

Optionally you can use this utility to compact archived data as well as local data. To do so, set *<enable_compact_archived_data>* to 1. You can also set the number of seconds compacting occurs before it stops. The following example compacts local and archived data for myTable; compacting stops after 10 minutes (600 seconds).

```
compacttbl myTable 600 1
```

IMPORTANT: This command cannot compact any node that stores data archived to a nearline storage device. Only after the retention period has ended can the nearline storage data be compacted. For more information, see:

- “Installing Perl Modules”, on page 415
- Compact in Chapter 3, “Collector Configuration” in the *Event Collection Guide*

IMPORTANT: Consult Hexis Cyber Solutions Technical Support before using this command, because it can be very slow and often yields little improvement.

force-upgrade

```
force-upgrade <table_name>
```

To avoid modifying terabytes of data during upgrades, the EDW only upgrades the file format of its data files as new data is loaded. Over multiple upgrades, this can result in a data store with many versions of the files. In some rare cases, newer versions of the data store provide better performance or other features, and it is advantageous to manually upgrade production tables to the latest version.

IMPORTANT: Consult Hexis Cyber Solutions Technical Support before using this command, because it can be very slow and results in the table being unreadable by older versions of the EDW.

Errors and Return Values

The atmanage utility prints an exit code of 0 on success, or a non-zero exit code and an error condition message upon error. For a full list, see “Errors and Return Values for Data-Store Utilities”, on page 133.

Examining the State of an EDW Data Store

This section describes the `atview` utility. It contains the following topics:

- “Synopsis”, next
- “Description”, on page 125
- “Options”, on page 125
- “Arguments”, on page 128
- “Usage and Examples”, on page 128
- “Errors and Return Values”, on page 130

Synopsis

```
atview [options] <cluster_list> <item> [<item> […]]
```

Description

The atview utility examines the state of an EDW instance, such as its metadata or status.

NOTE: Requests from atview do not alter the state of the EDW. For information on changing an EDW instance, see “Managing an EDW Data Store”, on page 111.

Options

Information About atview

`--version`

Displays version of `atview`.

`--help, --longhelp`

Displays online help for command in short form (`--help`), or with additional detail (`--longhelp`).

`--timestamps`

This parameter specifies that the output includes the minimum and maximum timestamp values of the data in each table. If the table is empty, the display shows "no records". When views are displayed in the output, those rows do not include the minimum and maximum timestamp values. The command executes more quickly when run without this parameter.

Namespace Identification

```
--namespace=<name>
```

Identifies the namespace from which to query information. The default is default. To specify the top level, use "" (empty quotation marks).

If you run the atview command without specifying a table name, the output includes all tables in the specified namespace. For example, if you specify --namespace=hq.west, the command returns the requested information for every table in the hq.west namespace.

If you specify a table when you run the atview command, the output includes the requested information only for the specified table. For example, if you specify --namespace=hq.west columns mytbl, the output is specific to hq.west.mytbl.

NOTE: When you run atview, you can specify a table name explicitly (for example, ns1.ns2.ns3.mytable) or use the --namespace=<*namespace*> flag to specify the namespace. When you use the namespace flag, you can specify the namespace once for an entire command; HawkEye AP prepends the specified namespace to each table name used in the rest of the request.

The command will display an 'error' column and include error messages in the column when there are inconsistencies across hosts in the cluster. For example, error type TABLE_OPEN_FAILURE displays in the 'error' column when the owner of a view is deleted from the system and the view no longer carries the permissions associated with the former owner.

Authentication Options

```
--user=<username>
```

This parameter authenticates the specified *name* as an EDW user. If no user is specified, atview reads the environment variable `ADDAMARK_USER`. If it cannot find this environment variable, it uses the operating system to get the name of the user who is logged in.

NOTE: To log in as the **guest** user, do not specify the `--pass` or `--shared-secret` flags.

```
--pass=<pwmethod>
```

This parameter provides several methods to specify the password. You can specify the password in clear text, place the password in a file and specify the location of the file, or set it in the `ADDAMARK_PASSWORD` environment variable. The syntax is:

```
--pass=<password>
--pass=pass:<password>
--pass=file:<filename>
--pass=default
```

When you specify default as the password, atview looks in the ADDAMARK_PASSWORD environment variable. For more information, see "Setting the ADDAMARK_PASSWORD Environment Variable", on page 86.

```
--shared-secret=<secret>
```

This parameter provides several methods to specify the shared secret. You can specify the secret in clear text, place the secret in a file and specify the location of the file, or set it in the ADDAMARK_SHARED_SECRET environment variable.

This parameter provides two methods to specify the shared secret. You can either specify the secret in clear text or place the secret in a file and specify the location of the file. The syntax is:

```
--shared-secret=<secret>
--shared-secret=file:</full_path/filename>
--shared-secret=default
```

When you specify `default` as the password, `atmanage` looks in the `ADDAMARK_SHARED_SECRET` environment variable.

NOTE: The value for `--pass` takes precedence over the value for `--shared-secret`, if both are present.

`--assume-role=role<role_to_be_assumed>`

This parameter causes the operation to be run as if the user had assumed the specified EDW role. The assumed role must have `admin` privileges. For more information about managing users and roles, see “Authentication Management”, on page 116.

`--timestamps`

This parameter specifies that the output includes the minimum and maximum timestamp values of the data in each table. If the table is empty, the display shows "no records". When views are displayed in the output, those rows do not include the minimum and maximum timestamp values.

Debugging

`--[no]dbgprint-request`

Dumps client-server request, and then exits. The default is `--nodbgprint-request`.

This option forces atview to print the execution requests that it would have sent to the EDW, and then stops execution rather than proceeding with the action. This is useful for debugging situations and for understanding the EDW API.

`--[no]raw`

This options forces atview to print the raw XML results that the EDW returns, instead of the interpreted results `atview` usually shows. The default is `--noraw`.

`--[no]show-responses`

When reading the response from the EDW server, atview filters out certain messages, reformats others, and attempts to present a simpler response suitable for most cases. The --show-responses flag forces `atview` to show the uninterpreted responses from the EDW in addition to normal output. The default is `--noshow-responses`.

Arguments

The atview utility takes as its first argument the <*cluster_list*> for the EDW that is to be examined. The <*cluster_list*> represents a comma-separated list of <*host*>:<*port*> pairs. The list must be enclosed in quotation marks (").

Specifying a list rather than a single host:port pair allows the utility to run even if the first of the specified EDW nodes is down. Each host:port pair identifies the EDW instance; for example, edw01:8072.

The command tries each host:port pair in order (left to right) until it finds one that allows a TCP (Transmission Control Protocol) connection. Failure to establish a TCP connection indicates that the EDW host is down. If the connection succeeds but the command fails, the command does not attempt to establish additional connections.

Example:

```
atview --user=administrator --password=changeme
   "edw01:8072,edw02:8072,edw03:8072" --namespace=s0meNS tables
```

The command then expects an <*item*> argument, which indicates the EDW object, state, or task to examine. Depending on the <*item*>, there are a number of additional arguments, as indicated in the table below.

Item	Additional Arguments	Description
tables		view tables and views in a namespace
namespaces		view namespaces (within a namespace)
columns		view columns for all tables in a namespace
columns	<*table_name*>	view columns for one table
tasks		show tasks currently running across the instance
disk		show disk capacity, use, and free space
tabledisk	<*table_name*>	show disk use for the specified table, not including disk used for directories

Usage and Examples

tables: Viewing Tables and Views in a Namespace

Lists the tables and views in a given namespace. If you specify the `--timestamps` option, the output includes the start and end timestamps, which are useful to create `DURING` clauses.

Example:

```
  atview --user=administrator --pass=s0mep@ss lmshost:8072 \
  --namespace=external.websvcs --timestamps tables
```

NOTE: Views are listed always with `"(no records)"` in the columns for minimum and maximum timestamps. The command will display an ’error’ column and include error messages in the column when there are inconsistencies across hosts in the cluster. Otherwise the ’error’ column is empty.

```
+------------- +--------------+--------------+-----------------
|namespace|table_name| min_time | max_time |error|
+------------+---------------+--------------+----------------+
|default |events |(no records) |(no records)| |
|default |sys_alerts |(no records)|(no records)| |
+-------------+--------------+---------------+---------------+-----+
```

namespaces: Viewing Namespaces Within a Namespace

Lists the namespaces within a given namespace. The output includes each namespace, the number of tables within it, and the cumulative number of tables for all namespaces within the namespace.

Example:

```
atview --user=administrator --pass=pass:s0mep@ss lmshost:8072 \
  --namespace=external.websvcs namespaces
```

columns: Viewing Columns for All Tables in a Namespace

Lists the columns for every table in the given namespace, including each column's data type, and the start and end timestamps for the table. Timestamps are stored in the **ts** column.

Example:

```
atview --user=administrator --pass=pass:s0mep@ss lmshost:8072 \
  --namespace=external.websvcs columns
```

columns <***table_name***>: Viewing Columns for One Table

Lists the columns for one table.

Example:

```
atview  --user=administrator --pass=pass:s0mep@ss lmshost:8072 \
  --namespace=external.websvcs columns stock_quote_log
```

tasks: Showing Tasks Currently Running Across the EDW instance

Lists tasks currently running in the EDW instance.

Example:

```
atview --user=administrator --pass=pass:s0mep@ss lmshost:8072 tasks
```

disk: Showing Disk Capacity, Used Disk Space, and Free Disk Space

Lists the total disk capacity, amount in use, and the amount of free space across the instance. The amount of free disk is a conservative, weighted estimate, which attempts to account for the uneven filling of disk space as new log records get loaded.

Example:

```
atview --user=administrator --pass=pass:s0mep@ss lmshost:8072 disk
```

tabledisk <***table_name***>: Showing Disk Use for the Specified Table

Lists the amount of disk space consumed for the given table, across the EDW instance, not including disk space consumed by directories.

Example:

```
atview --user=administrator --pass=pass:s0mep@ss lmshost:8072 \
  --namespace=external.websvcs tabledisk stock_quote_log
```

Errors and Return Values

See “Errors and Return Values for Data-Store Utilities”, on page 133.

Retiring Data

This section contains the following topics:

- “Synopsis”, next
- “Description”, on page 130
- “Options”, on page 131
- “Parameters”, on page 131
- “Arguments”, on page 131
- “Using the FORCE Keyword”, on page 132
- “Specifying the Data for Deletion”, on page 132
- “Permissions Required to Retire Data”, on page 133
- “Interrupting Deletions”, on page 133
- “Errors and Return Values for Data-Store Utilities”, on page 133

Synopsis

```
RETIRE FROM <table_name> <command> [FORCE]
```

Description

The retire command is a Sensage SQL extension rather than a command-line tool and, therefore, is unlike the utilities documented in this chapter. You do not run retire from the command line as you do the utilities. Instead, you use the atquery utility to run it.

The retire command replaces the deprecated atretire utility as the tool for retiring data from an EDW instance. The retire command provides significantly enhanced performance over atretire.

Moreover, the syntax of the retire command is easier and more comprehensive than that of the atretire command. While the atretire command required you to specify the cutoff time in the predefined syntax of a timestamp, the retire command allows you to specify the cutoff time as any Sensage SQL expression that evaluates to a timestamp data type.

The retire command permanently deletes data from the specified table in the EDW instance. Use this command to:

- Reclaim Space

  Although the combination of EDW compression and ever-increasing disk capacity has reduced the urgency of deleting older data, there are still applications that require this regular maintenance. You can use `retire` to delete old data, retiring on a range of upload IDs or timestamps.

  IMPORTANT: Retiring data from an EDW table removes only data that is *not* stored under retention on a nearline storage device. The command skips archived data that is under retention and returns a message that provides the date in the future when the nearline storage device will retire the data. That date is the earliest you can retire the data. The command does remove archived data that is not under retention or its retention period has passed. For more information, see Chapter 9: Archiving to Nearline Storage.

- Undo Duplicate Loads

  Occasionally, the same log file is accidentally loaded twice. You can use `retire` to remove one of the copies. You delete based on upload ID.

- Undo Loads Containing Bad Data

  Occasionally, the source data itself is incorrect, and you want to fix the data in the long-term archives. If the problem is limited to one log file, Hexis Cyber Solutions recommends backing out this one file, fixing the source data, then reloading. However, if the corruption occurred over the course of weeks, it may be preferable to use retire to delete selected records.

Options

See the options documented for atquery. For more information, see “Querying Data”, on page 89.

Parameters

See the parameters for `atquery`. For more information, see “Querying Data”, on page 89.

Arguments

Argument	Additional Arguments	Description
<tablename>		target table name
<command>	UPLOADS <upload_id> [, <upload_id>[...]][DURING <timestamp>, <timestamp>] Deletes the data associated with specified load(s)— specified by upload ID.BEFORE <timestamp_expression> Deletesdatawhosetimestamp('ts')value is prior to the specified timestamp expression.	One of two deletion commands to perform on the specified table For more information, see “Specifying the Data for Deletion”, next.

Using the FORCE Keyword

The FORCE keyword causes the RETIRE command to remove corrupted data. If you run RETIRE without this keyword and the EDW detects any corruption in the data to be retired, it removes only the non-corrupted data specified by the command and returns an error about the corrupted data.

Hexis Cyber Solutions recommends that you run RETIRE without the FORCE keyword when you first retire old or bad data. Running without forcing the retirement allows the EDW to inform you of corrupted data and provides the opportunity for you to examine that data. After you have taken the necessary steps to prevent similar corruption in the future, run RETIRE with the FORCE keyword to remove the corrupted data.

Specifying the Data for Deletion

There are two ways to specify which data to delete.

- “Using the UPLOADS Command”, next
- “Using the "BEFORE <timestamp_expression>" Command”, on page 133

Using the UPLOADS Command

To permanently delete the records associated with a given set of upload IDs, provide a comma-separated list of upload IDs. The syntax is:

```
UPLOADS <upload_id> [, <upload_id>[...]][DURING <timestamp>, <timestamp>]
```

where the DURING causes the RETIRE command to delete only the data in the specified time and date range. You can choose the optional DURING clause when deleting an upload ID that encompasses a very wide time and date range. Using the DURING clause can help avoid possible negative impacts on performance by breaking the deletion job into a series of smaller operations addressing shorter time ranges within the larger min and max timestamp of the load.

Example:

```
atquery localhost:8072 -e RETIRE FROM syslog
 UPLOADS ’31AEAF94551AA4366519F4)EB3F9A4FA’, ’F8OC8F2003D655A8444A8B4EF172EC7a’
```

Example With `DURING clause`:

```
atquery localhost: 8072 -e "RETIRE FROM syslog
UPLOADS ’31AEAF94551AA4366519F4)EB3F9A4FA’, ’F8OC8F2003D655A8444A8B4EF172EC7a’
DURING time(’FEB 01 00:00:00 2011’), time(’May 31 23:60:00 2011’)"
OR DURING time (’Mar 01 00:00:00 2011’), time (’May 31 23:60:00 2011’)
```

Example With `FORCE` Keyword:

```
atquery localhost:8072 -e "RETIRE FROM syslog
 UPLOADS '31AEAF94551AA4366519F40EB3F9A4FA', 'F80C8F2003D655A8444A8B4EF172EC7A'
 FORCE"
```

Using the "BEFORE <timestamp_expression>" Command

To permanently delete the records whose date and time precede a specified timestamp expression, follow the `BEFORE` key word with a timestamp expression. The syntax is:

```
BEFORE <timestamp_expression>
```

where *<timestamp_expression>* can use any Sensage SQL expression that evaluates to a timestamp data type. For example:

```
atquery localhost:8072 -e "RETIRE FROM syslog
   BEFORE time('Aug 26 11:02:34 2005')"
```

Example With `FORCE` Keyword:

```
atquery localhost:8072 -e "RETIRE FROM syslog
   BEFORE time('Aug 26 11:02:34 2005') FORCE"
```

For more information, see Time Functions in Chapter 11, “SQL Functions” in the *Reporting Guide*.

Permissions Required to Retire Data

To run this command, you must have access to the namespace where the table exists and your user role must have sls.(EDW) retire permission. For more information, see Users, Roles, and Permissions in Chapter 8, “Administering Users and Authentication”.

IMPORTANT: Hexis Cyber Solutions recommends that you give sls.retire permission to only a few trusted users.

Interrupting Deletions

The retire command retires all specified data or none of the specified data. Therefore, if a retire operation is interrupted, the EDW rolls back all changes made by the retire request.

ERRORS AND RETURN VALUES FOR DATA-STORE UTILITIES

Error Code or Return Value	Command	Description
0	all	Success. Command completed successfully or `--dbgprint-request` or `--version` or `--raw` was also specified. For "`atview tasks`", at least one task was found running.
2	`atmanage droptbl`	The table did not exist.
	`atview tasks`	No relevant tasks found.
	`atview [namespaces,tables,columns]`	No namespaces, tables, or columns/types found.
	`atview tabledisk`	Specified table does not exist.

Error Code or Return Value	Command	Description
`3`	`atmanage`	Could not fulfill command against the table.
	`atview`	Some problem retrieving the specified information (tasks, disk space information, meta-data).
`10`	all except `atload`	Interrupt (for example, `SIGTERM`, `SIGINT`) caught once, and then caught again before the program finishes recognizing the first interrupt.
`11`	all	The `--help` or `--longhelp` options are specified, or there are command-line parsing problems, or any other condition that leads to the program displaying its usage information.
`12`	`atload`	Load failure due to interrupt or failed load.
	`atquery`	Errors happen during the query, and no interrupts happen; or `--raw` is specified and the program was not able to connect to the specified port on the specified host.
	`atmanage`	The upgrade process was interrupted (twice), not giving full opportunity for this utility to clean up any ongoing upgrade it had started.
`13`	`atmanage, atview`	`--raw` is specified.
	`atquery`	Query interrupted (and cancellation ran to completion).
`14`	`atquery`	Problems with `--postproc`.
`15`	`atquery`	Interrupt happens while query is not running.
`20`	all	There are problems with the supplied command line, consistency and compatibility in the command line arguments, or problems with the specified files (for `atload`, PTL files; for `atquery`, SQL or Perl file).
`21`	`atmanage force-upgrade`	The upgrade was cancelled before any effects took place on the table.
`22`	`atmanage force-upgrade`	The upgrade was cancelled, but only after some effect took place on the table.
`23`	`atmanage force-upgrade`	The upgrade failed before any effects took place on the table.
`24`	`atmanage force-upgrade`	The upgrade failed, but only after some effects possibly took place on the table.
`98`	all	Problems connecting to the EDW server.
`99`	all	Unhandled internal errors.

Example Event-Log Data

Your HawkEye AP solution includes example log events from a variety of log-source types. The examples include the source log events, corresponding PTL files for loading the events into the EDW, and corresponding `.sql` files with queries that can be run against the loaded events. The event data comes from real production systems; to protect privacy, some values have been substituted in a 1-for-1 mapping to preserve statistical relevance.

This section includes these topics:

- "Tuxedo Data", on page 135
- "Websrv Data", on page 135
- "Windows Data", on page 136

NOTE: HawkEye AP compression technology works more efficiently than `gzip` for very large volumes of data. Due to file-system overhead, benefits are not seen generally below about 10-50MB of log data.

Tuxedo Data

The example source log file contains Tuxedo "`STDERR`" log events. You can find the log and related files in this location:

```
<SenSage_Home>/latest/example_logs/tuxedo
```

records	500,000
size	36,368,971 (uncompressed) / 2,692,070 (gzip -9 ==> 13.5:1)
start time	997080780 (2001/08/05-11:53:00 PDT 2001/08/06-06:53:00 GMT)
end time	997085724 (2001/08/06-01:15:24 PDT 2001/08/06-08:15:24 GMT)
bytes/record	72.74
fields	6
bytes/field	12.12
HawkEye AP size	683,264 bytes (53:1)

Websrv Data

The example source log file contains Web site events recorded by a Microsoft IIS web server. You can find the log and related files in this location:

```
<Sensage_Home>/latest/example_logs/websrv
```

records	60,191
min timestamp	Wed Jan 30 07:44:37 2002 GMT (1012376677)
max timestamp	Sun Mar 17 18:08:45 2002 GMT (1016388525
columns	12
data files	`websrv_log.gz` (495,653 bytes) `websrv_log_100.gz` (1734 bytes) `websrv_log2_100.gz` (1260 bytes)
parse failures	88 records in `websrv_log.gz`

Windows Data

The example source log file contains events recorded by Windows systems and collected by the HawkEye Retriever (Windows). You can find the log and related files in this location:

```
<Sensage_Home>/latest/example_logs/sensage_win_evt
```

records	29,940
size	8,040,491 (uncompressed) / 218,638 (gzip-9 ==> 37:1)
start time	1161808842 (2006-11-25 20:40:42 GMT)
end time	1164652992 (2006-11-27 18:43:12 GMT)
bytes/record	268.56
fields	15
bytes/field	17.90
HawkEye AP size	3,036,256 bytes (2.6:1)

HawkEye AP compression technology works more efficiently than gzip for very large volumes of data. Due to file-system overhead, benefits are not seen generally below about 10-50MB of log data.

CHAPTER 4

Administering an EDW Instance

This chapter contains the following sections:

TASK PRIORITY AND QUEUING

This section contains the following subsections:

Also see: “Querying Data”, on page 89.

How the Event Data Warehouse (EDW) Queues Tasks

The EDW has three queues: one for Normal Priority processes, one for Critical Priority processes, and one for Urgent Priority processes. The first two of these queues are defined in:

```
<Sensage_Home>/latest/etc/sls/instance/<instance_name>/athttpd.conf
```

as:

- `normalrunqueuequota` (default `3`)
- `criticalrunqueuequota` (default `10`)

The third process, UrgentRunQueueQuota, cannot be configured. This process has an infinite depth, which means it can run an infinite number of jobs. The EDW immediately runs everything on this queue)

The command "atmanage canceltask" has urgent priority by default, all other tasks have normal priority. Therefore the --skip-queues option is not necessary for cancels to work on a loaded system.

The Normal and Critical queues independently allow a set number of requests to run simultaneously. (This number can be changed for these queues, see “Setting the Number of Tasks Allowed in a Queue”, on page 139).

If there are more requests of a given priority than can be run simultaneously, the later requests will be blocked until an earlier request completes. Blocked tasks are marked as EVENT_WAIT in the state column of the system.task_list table.

Monitoring Task Priority

The EDW returns progress indicators for requests waiting in the queue.

Use the "atview tasks" command to see task priority. For example:

```
atview --user=administrator --pass=pass:s0mep@ss lmshost:8072 tasks
```

To view more specific information about task priority, query the system.task_list table with the --skip-queues flag to push your query to the top of the tasks. For example:

```
atquery --skip-queues --user=administrator \
--shared-secret=file:shared_secret.asc "edw01:8072,edw02:8072,edw03:8072"\
--namespace='system' -expression='SELECT * FROM task_list;'
```

IMPORTANT: The --skip-queues flag works only with a shared-secret, not a password. See “Authentication Options”, on page 85.

The system.task_list table contains the following information about running tasks:

Column Name	Data Type	Description
row_type	varchar	Type of information being returned: NODE or TASK
node	varchar	The host that is being returned
port	int32	The port number (installation)
_systaskid	varchar	A 32-character hex string denoting the internal task ID
method_name	varchar	The name of the internal method being run
_extid	varchar	The "external" ID to which the client program given this task
at_top	bool	If true, the process running as the "master" for this operation
start_time	varchar	The start time of this task
finish_time	timestamp	The end time of this task (flushes every few minutes)
state	timestamp	The state of the task, such as RUNNING. The wait/blocked state for task priority queuing is displayed as EVENT_WAIT.

Cancelling Tasks

To cancel a task, use "`atmanage canceltask <request_id>`" as shown in the following example:

```
atmanage --user=administrator --pass=pass:s0mep@ss \
lmshost:8072 canceltask 673FA7200BC6F109955E227201BCEA73255
```

To find the internal task IDs, run "`atview tasks`".

Setting the Number of Tasks Allowed in a Queue

Each host in an EDW instance maintains its own EDW local task queue, which has a default queue depth of 3 normal tasks and 10 critical tasks. There is no default size for the Urgent queue.

To change the number of tasks allowed in the Normal and Critical queues, use the "`clsetup add|edit sls`" flags:

```
--normal-runqueue-quota
--critical-runqueue-quota
```

These flags reset the default to the new limit you choose for all hosts in the EDW instance. Queue limits cannot be changed on a per-host basis.

For example, to change the normal queue limit from `3` to `6` across all hosts:

```
clsetup edit sls "edw01:8072,edw02:8072,edw03:8072" --normal-runqueue-quota=6
```

Backup and Restore

This section describes the backup and restore process.

For information on recovering failed EDW node, see “Recovering a Failed EDW Node”, on page 175.

Backing Up and Restoring an EDW Table

This topic provides an overview of the backup and restore process for EDW tables. Hexis Cyber Solutions strongly recommends that you perform these processes only in consultation with Technical Support.

Overview

Backup and restore operations occur at the table level and are performed using the `atmanage` and `atquery` commands. The two sections below document the operations these commands perform on a table in the datastore.

BACKUP OVERVIEW

- User submits the following command, which sets the table to read-only (quiesce) and verifies the table's integrity; if the table is already read-only, saves the state:

```
atmanage --user=administrator --pass=<password> \
  --namespace=<namespace> <cluster_list> backuptbl start <tablename>
```

- User submits `atquery` against virtual table (`<tablename>.storage.dsminfo`)
- EDW uses returned data from `dsminfo` to pass to file-system backup tool
- User submits the following command, which returns the table to its original read-only / read-write state before the "`atmanage backuptbl start`" command was submitted; if the table was already in a read-only state, the following command retains that read-only setting:

```
atmanage --user=administrator --pass=<password> \
  --namespace=<namespace> <cluster_list> backuptbl commit <tablename>"
```

- User submits the following command, which returns the read-only / read-write state to previous setting; there is nothing to roll back because your backup tool performs the actual backup:

```
atmanage --user=administrator --pass=<password> \
  --namespace=<namespace> <cluster_list> backuptbl rollback <tablename>"
```

NOTE: In addition to backing up your tables, you should backup your LDAP server. For more information, see "Backing Up Your LDAP Server", on page 141.

RESTORE OVERVIEW

- User submits the following command, which returns a sandbox (directory) on each host of the EDW instance to which a `dsroot` tree is to be restored:

```
atmanage --user=administrator --pass=<password> \
  --namespace=<namespace> <cluster_list> restoretbl start <tablename>"
```

- User submits the following command, which verifies the sandbox on each host, places the sandbox into the tree on each host, and returns the read-only / read-write state to previous setting:

```
atmanage --user=administrator --pass=<password> --namespace=<namespace> \
  <cluster_list> restoretbl commit <tablename>
```

- User submits the following command, which removes the sandbox and returns the read-only / read-write state to previous setting:

```
atmanage --user=administrator --pass=<password> \
  --namespace=<namespace> <cluster_list> restoretbl rollback <tablename>"
```

NOTE:

- Incremental backups are by date and may encompass more than you would expect due to tree reorganization.

- In addition to restoring your tables, you must restore your LDAP server. For more information, see “Restoring Your LDAP Server”, on page 141.

Backing Up Your LDAP Server

To back up your LDAP server, run the following commands as the lms user on the host running the LDAP Server:

```
<Sensage_Home>/latest/etc/init.d/sensage_atslapd stop
<Sensage_Home>/latest/sbin/slapcat -f <Sensage_Home>/latest/etc/atslapd/
slapd.conf | grep -v ^creat | grep -v ^modif \
> <Sensage_Home>/latest/data/backups/backup.ldif
<Sensage_Home>/latest/etc/init.d/sensage_atslapd start
```

Restoring Your LDAP Server

To restore your LDAP server, run the following commands as the lms user on the host running the LDAP Server:

```
<Sensage_Home>/latest/etc/init.d/sensage_atslapd stop
tar czvfC <Sensage_Home>/latest/data/backups/slapd_backup.tgz <Sensage_Home>/
latest/data atslapd
/bin/rm -f <Sensage_Home>/latest/data/atslapd/*
<Sensage_Home>/latest/sbin/slapadd -f <Sensage_Home>/latest/etc/atslapd/
slapd.conf \
-l <Sensage_Home>/latest/data/backups/backup.ldif
<Sensage_Home>/latest/etc/init.d/sensage_atslapd start
```

Backing Up Your PostgreSQL Server

To back up your PostgreSQL server, run the following commands as the lms user on the host running the PostgreSQL Server:

```
<Sensage_Home>/latest/bin/pg_dump -c -U lms -d controller -f
<Sensage_Home>/latest/data/backups/pgsql_backup
```

Restoring Your PostgreSQL Server

To restore your PostgreSQL server, run the following commands as the lms user on the host running the PostgreSQL Server:

```
<Sensage_Home>/latest/bin/psql -U lms -d controller < <Sensage_Home>/latest/
data/backups/pgsql_backup
```

DEFINING DATA OBJECTS

In addition to using the `atquery` utility to query the database, you can also use it to create, rename, and drop database tables, column filters, and views. You can either specify these actions in the `.sql` files that `atquery` processes or you can use the dash option to specify the data manipulation statement directly from the command line.

You can use `atquery` to create and manipulate the following database objects in an EDW instance:

- **Tables**—tabular objects that store loaded log events

For more information, see “Defining Tables with Sensage SQL (SSQL)”, next.

- **Column Filters**—special objects attached to table columns to improve query performance

  For more information, see “Defining Column Filters with Sensage SQL”, on page 144.

- **Views**—tabular objects through which log events in tables can be viewed in different ways

  For more information, see “Defining Views with Sensage SQL”, on page 149.

Defining Tables with Sensage SQL (SSQL)

Tables store log events that have been loaded into in an EDW instance.

Syntax for Creating Tables

```
CREATE TABLE [<namespace>.]<table_name>
  ( ts timestamp[, <column_declaration>[, <column_declaration>[...]] )
;
```

SPECIFYING NAMES FOR TABLES AND COLUMNS

Follow these rules for the names of tables and columns:

- The first character must be an ASCII underscore (_) or an ASCII English A-z.
- Remaining characters can be any of the above, plus the ASCII numerals 0-9.
- The name can be from 1 through approximately 32 Kb characters in length.

DECLARING THE COLUMNS IN TABLES

Enclose the list of columns you declare for a table in parentheses. Separate column declarations with commas. There is virtually no limit to the number of columns that you can declare for a table.

Column declarations have the following form:

```
<column_name> <data_type>
```

The list of columns must declare a column named "ts", and its data type must be "timestamp". Other column declarations can have any name that follows the naming rules listed above, and its data type can be any of the data types supported by Sensage SQL.

For more information on data types supported by Sensage SQL, see Data Types in Chapter 10, “Sensage SQL” in the *Reporting Guide*.

Automatic Columns when Creating Tables

The EDW creates automatically a special column named _uploadid, with SQL data type varchar. The column holds the ID of the upload operation that inserted each row into the table.

For more information about the _uploadid column, see “Tracking Uploads in the EDW”, on page 88 and “Using the UPLOADS Command”, on page 132.

Creating Tables in Specific Namespaces

Tables in an EDW instance are located within groups called *namespaces*. If you do not specify a namespace when you create a table, it is located in the default namespace established for the EDW instance. If you specify the namespace at the time you create the table, the table is created within the specified namespace. If the specified namespace does not exist, it is created. For information on how to specify a namespace, see “Specify a Namespace”, on page 112.

IMPORTANT: If you create tables in different namespaces that store data from the same type of log source, Hexis Cyber Solutions recommends that you name the tables identically.

When you create a table, you can specify the namespace explicitly in the `CREATE TABLE` statement, or you can specify it with the `--namespace` option of the `atquery` command. For example, assume you have a `.sql` file named `myTable.sql` that contains the following statement:

```
CREATE TABLE myNamespace.myTable
  (ts           timestamp,
   HostName     varchar,
   ProgName     varchar,
   Message      varchar
  )
;
```

Assume you run the following atquery command:

```
atquery --namespace=firewalls host02:8072 myTable.sql
```

A table named `myTable` will be created in the namespace `firewalls.myNamespace`.

Permissions for Creating Tables

To create a table, you must have `sls.create` permission in the EDW instance, and you must have `sls.namespace` permission on the namespace where you want the table to be created.

Renaming Tables

Use the `atmanage` command to change the names of tables.

For more information on how to rename a table, see “Renaming Tables”, on page 157.

Syntax for Dropping Tables

```
DROP TABLE [<namespace>.]<table_name>
;
```

When you drop a table, you must identify fully the namespace in which the table is located. You can specify the namespace explicitly in the DROP TABLE statement, or you can specify it with the `--namespace` option of the `atquery` command.

IMPORTANT: When you a drop a table, views and reports that depend on the table become invalid. In addition, the log events loaded in the table are discarded from the EDW instance.

PERMISSIONS FOR DROPPING TABLES

To drop tables, you must have `sls.drop` permission in the EDW instance, and you must have `sls.namespace` permission on the namespace where the table you want to drop is located.

Defining Column Filters with Sensage SQL

Column filters let you influence the execution plans for sparse-row queries. In the EDW, create filters on columns instead of adding indexes. Column filters enhance performance for queries that use the:

- equals (`=` or `==`) or not equals (`<>` or `!=`) operators
- `_strleft()`, `_strright()`, `_strmiddle()`, and `_substr()` functions

NOTE: A column filter enhances performance only when the matched string is a constant.

Syntax for Creating Column Filters

The syntax below creates filters on column(s) in the specified table. If you specify column(s) in the command, a filter is created for each specified column.

```
CREATE FILTERS [<filter_name>] ON [<namespace>.]<table_name>
   { (<column_name> [<partial_filter>])[, (<column_name> [<partial_filter>])
   [...]] | ALL }
;
```

NOTE:

- The table you specify with `<table_name>` and any columns you specify with `<column_name>` must exist. Specify `ALL` to create column filters on all the columns in the table, with the exception of the `ts` column. An error occurs if you specify `ts` as a `<column_name>`.
- Existing filters remain in place.

When you create column filters on a table, you must identify fully the namespace in which the table is located. You can specify the namespace explicitly in the CREATE FILTERS ON statement, or you can specify it with the `--namespace` option of the `atquery` command.

IMPORTANT:

- All filters have names. If you do *not* specify a filter name in the `create filters` statement, the filter is named identically to the column it filters. If you *do* specify a filter name in the `create filters` statement, you can filter on only one column. The rules for naming a filter are the same as those for naming tables and columns; for more information, see "Specifying Names for Tables and Columns", on page 142.
- Each filter definition on a table must be unique to that table.

SYNTAX FOR DEFINING PARTIAL FILTERS

Partial filters enable filtering on text in a specified part of a varchar column. These filters enhance performance when:

- a query uses any of the following HawkEye AP string functions: `_strleft()`, `_strmiddle()`, `_strright()` or `_substr()`

- the matched string is a constant
- the length of the matched string is greater than or equal to the count or length specified in the definition of the column filter

NOTE: Additionally, the `LEADING`, `TRAILING`, and `ANY SUBSTRING` filters enhance performance when a query specifies equality on the named column.

The syntax for *<partial_filter>* is:

```
LEADING <count> | TRAILING <count> | SUBSTRING <offset>, <length> | ANY SUBSTRING
<length>
```

When you specify a partial filter with the forms `LEADING`, `TRAILING,` or `SUBSTRING`, the SQL engine applies the corresponding string function to the column value when it constructs the filter on a varchar column. A query applies the same function to the predicate string that is matched to use the filter.

NOTE: The examples in the next three sections assume a table named `cdr` in a namespace also named `cdr`

Example of the LEADING Filter

For a leading filter, the filter is constructed on the first *<count>* characters in the column.

```
atquery --namespace=cdr localhost:8072 -e 'create filters caller_leading ON cdr
(calling_number LEADING 3);'
```

For this filter, assume a calling number of `1234567890`. If a query run against this table includes either of the following predicates, the query takes advantage of this column filter:

```
where calling_number = "1234567890"
```

or

```
where _strleft(calling_number,3)="123"
```

Example of the TRAILING Filter

For a trailing filter, the filter is constructed on the last *<count>* characters in the column.

```
atquery --namespace=cdr localhost:8072 -e 'create filters callee_trailing ON cdr
(called_number TRAILING 4);'
```

For this filter, assume a called number of `9876543210`. If a query run against this table includes either of the following predicates, the query takes advantage of this column filter:

```
where called_number = "9876543210"
```

or

```
where _strright(called_number,5)="43210"
```

Example of the SUBSTRING Filter

For a substring filter, the filter is constructed on the substring that starts at the specified *<offset>* and extends the specified *<length>*.

```
atquery --namespace=cdr localhost:8072 -e 'create filters callee_substr ON cdr
(called_number SUBSTRING 1,4);'
```

For this filter, again assume a called number of 9876543210. If a query run against this table includes either of the following predicates, the query takes advantage of this column filter:

```
where _substr(called_number,1,4)="8765"
```

or

```
where _strmiddle(called_number,1,4)="8765"
```

NOTE:

- When searching for a string on the left, the first character is in the zero (0) position.
- The syntax requires a comma to separate the offset and length values specified for the SUBSTRING filter. However, as illustrated below in System Tables for Column Filters, the comma does not display in the filter definition.

Example of the ANY SUBSTRING Filter

For an ANY SUBSTRING filter, the filter is constructed on all substrings in a varchar column of the specified *<length>*.

```
atquery --namespace=cdr localhost:8072 -e 'create filters callee_anysubstr ON
cdr (called_number ANY SUBSTRING 8);'
```

For this filter, again assume a calling number of 1234567890. If a query run against this table includes any of the following predicates, the query takes advantage of this column filter:

```
  where _strmiddle(calling_number,0,8)=="12345678"
```

or where the number of characters in the predicate is greater than or equal to the number defined for the filter:

```
  _strmiddle(calling_number,0,9)=="123456789"
```

or where the predicate uses the `_substr()` function:

```
  _substr(calling_number,0,9)=="123456789"
```

or where the predicate function includes only the first two arguments:

```
  _substr(calling_number,1)=="234567890"
```

IMPORTANT: Specify a length filter cautiously because it tends to be much larger than the other filter types. You can estimate the relative size of the filter if you understand the data. For example, assume that you know that most values in the column are 10 characters long and you create a filter of length 7. The typical number of filter values will be 10 minus 7, or 3 filter values for each column value. In this case, the filter can be expected to be three times the size of the other filter types. On the other hand, if you know that most values in the column are `30` characters long and you create a filter of length `10`, you can expect the filter to be twenty times the size of other filter types.

SYSTEM TABLES FOR COLUMN FILTERS

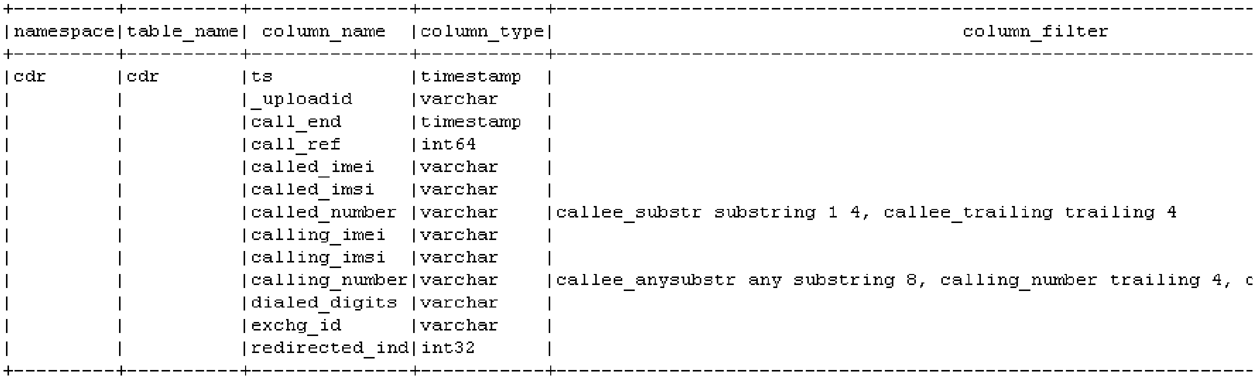
The `column_filter` column in `<namespace>.storage.metadata` system tables identifies all filters applied to all columns in a given table.

To view the column filters in the `cdr.cdr` table used in the examples in Syntax for Defining Partial Filters above, you can run the following command:

```
atview --namespace=cdr localhost:8072 columns cdr
```

Figure 4-1 and Figure 4-2 illustrate the output.

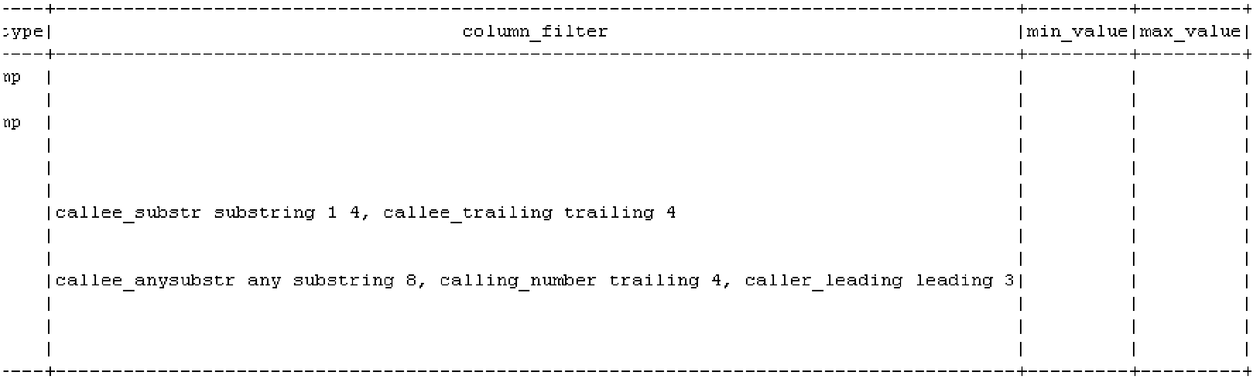
Figure 4-1: Left Side of Output



```
+---------+----------+--------------+-----------+---------------------------------------------------------------
|namespace|table_name| column_name  |column_type|                                                     column_filter
+---------+----------+--------------+-----------+---------------------------------------------------------------
|cdr      |cdr       |ts            |timestamp  |
|         |          |_uploadid     |varchar    |
|         |          |call_end      |timestamp  |
|         |          |call_ref      |int64      |
|         |          |called_imei   |varchar    |
|         |          |called_imsi   |varchar    |
|         |          |called_number |varchar    |callee_substr substring 1 4, callee_trailing trailing 4
|         |          |calling_imei  |varchar    |
|         |          |calling_imsi  |varchar    |
|         |          |calling_number|varchar    |callee_anysubstr any substring 8, calling_number trailing 4, c
|         |          |dialed_digits |varchar    |
|         |          |exchg_id      |varchar    |
|         |          |redirected_ind|int32      |
+---------+----------+--------------+-----------+---------------------------------------------------------------
```

NOTE: The user created the second filter for `calling_number` without providing a filter name. The name defaulted to the name of the filtered column.

Figure 4-2: Right Side of Output



```
----+-------------------------------------------------------------------------------------+---------+---------+
:ype|                                    column_filter                                    |min_value|max_value|
----+-------------------------------------------------------------------------------------+---------+---------+
mp  |                                                                                     |         |         |
    |                                                                                     |         |         |
mp  |                                                                                     |         |         |
    |                                                                                     |         |         |
    |                                                                                     |         |         |
    |                                                                                     |         |         |
    |callee_substr substring 1 4, callee_trailing trailing 4                              |         |         |
    |                                                                                     |         |         |
    |                                                                                     |         |         |
    |callee_anysubstr any substring 8, calling_number trailing 4, caller_leading leading 3|         |         |
    |                                                                                     |         |         |
    |                                                                                     |         |         |
    |                                                                                     |         |         |
----+-------------------------------------------------------------------------------------+---------+---------+
```

NOTES ABOUT CREATING PARTIAL FILTERS

This section covers the following notes:

- "Multiple Filters on the Same Column", next
- "Keywords", on page 148

Multiple Filters on the Same Column

Several filters can be created on the same column, as illustrated above in System Tables for Column Filters. When a query is run against a column with multiple filters, the SQL engine chooses the best filter that is fully contained by the string.

For example, assume that a column has three `LEADING` filters: `LEADING 8`, `LEADING 11`, and `LEADING 15`. In a query that is matching a 10-character string, the SQL engine uses the `LEADING 8` filter to evaluate the first 8 characters of the string.

Keywords

The following words have been added to the list of SQL keywords:

- `ANY`
- `LEADING`
- `SUBSTRING`
- `TRAILING`

None of these words can be used for table or column names.

If existing customers have tables or columns that use these words they will have to change them.

Permissions for Creating Column Filters

To create column filters on a table, you must have `sls.create` permission in the EDW instance, and you must have `sls.namespace` permission on the namespace where the table is located.

Syntax for Dropping Column Filters

The syntax for dropping column filters differs depending on whether you name the filter in the command.

Dropping a Named Filter

```
DROP FILTERS [<filter_name>] ON [<namespace>.]<table_name>
;
```

Dropping Filter(s) by Column Name

```
DROP FILTERS ON [<namespace>.]<table_name>
  { (<column_name>)[, (<column_name>)[...]] | ALL }
;
```

NOTE:

- To drop a column filter, specify either the filter name or the column(s) that have filters you want to drop.
- When you list columns explicitly, only the columns you specify lose their filters; filters on other columns remain in place.
- Specify `ALL` to drop all column filters in the table.

- When you drop column filters from a table, you must identify fully the namespace in which the table is located. You can specify the namespace explicitly in the DROP FILTERS ON statement, or you can specify it with the `--namespace` option of the `atquery` command.

Permissions for Dropping Column Filters

To drop column filters from a table, you must have `sls.drop` permission in the EDW instance, and you must have `sls.namespace` permission on the namespace where the table is located.

Defining Views with Sensage SQL

Views let people see log events in tables in different ways. Views do not store log events themselves; they show selected columns and rows from other tables and views. You can use SELECT statements to query views in the same way you query tables.

This section describes these topics:

- "Syntax for Creating Views", next
- "Declaring the Columns in a View", on page 150
- "Selecting a Subset of Table Columns or Rows for a View", on page 152
- "Specifying and Querying Against Time Ranges in a View", on page 153
- "Using Processing Directives when Creating a View", on page 153
- "Declaring a Union of Tables in a View", on page 153
- "Syntax for Renaming Views", on page 154
- "Syntax for Dropping Views", on page 155
- "Retrieving Information About a View", on page 155

NOTE: The term "table" means generally "tables and views" throughout HawkEye AP documentation. Where the term applies to tables only, the restriction is mentioned specifically.

Syntax for Creating Views

```
CREATE [OR REPLACE] VIEW [<namespace>.]<view_name> [ ( <column_list> ) ] AS
  <select_subclause>
;
```

The names of views follow the rules for the names of tables. For more information, see "Specifying Names for Tables and Columns", on page 142.

Creating or Replacing a View

If you create a view that already exists, an error occurs. Sometimes you want to create a view even if one by that name already exists. Use the `OR REPLACE` keyword phrase in conjunction with the `CREATE` keyword to cause the EDW to create a new view if one does not yet exist or to ignore a duplicate view error and replace the current view with a new definition. For an example, see "Declaring Union Views with _tablematch()", on page 154.

Creating Views in Specific Namespaces

Views in an EDW instance are located within EDW (SLS) groups called *namespaces*. If you do not specify a namespace when you create a view, it is located in the default namespace established for the EDW instance. At the time you create the view, you can specify the namespace in which you want to create it; if the namespace does not exist, it is created. For information on how to specify a namespace, see "Specify a Namespace", on page 112.

When you create a view, you can specify different namespaces for the view and the underlying table. In other words, you can create the view in one namespace based on a table in a different namespace. You can specify the namespace(s) explicitly in the `CREATE VIEW` statement or you can use the `--namespace` option of the `atquery` command or you can use a combination of these two options. If both table and view are in the same namespace, you can use the `--namespace` option of the `atquery` command to specify a shared namespace. If they are in different namespaces, you must explicitly specify the table's namespace in the `CREATE VIEW` statement.

IMPORTANT: If you create views in different namespaces that represent data from the same type of log source, Hexis Cyber Solutions recommends that you name the views identically.

The following `CREATE VIEW` statement illustrates creation of a view whose namespace is different from the view's underlying table:

```
CREATE VIEW newNamespace.myView AS
  SELECT *
    FROM existingNamespace.myTable
    DURING ALL
;
```

Assume the `CREATE VIEW` statement is contained in a file called `myView.sql` and you run the following atquery command to create the view:

```
atquery --namespace=firewalls host02:8072 myView.sql
```

A view named `myView` will be created in the namespace `firewalls.newNamespace`, which selects data from a table named `myTable`, which is located in the `firewalls.existingNamespace` namespace. In this case, the view and its underlying table share the same parent namespace but are located in different subordinate namespaces below the parent. If `firewalls.newNamespace` did not exist before you ran the `CREATE VIEW` statement, it is created by the statement.

Assume next that you run the following query to retrieve data from the view:

```
atquery --namespace=firewalls.newNamespace host02:8072 -e "select count(*) from
myView"
```

The query above selects and aggregates data from `firewalls.newNamespace.myView`. However, `firewalls.`**`newNamespace`**`.myView` pulls the data from `firewalls.`**`existingNamespace`**`.myTable`.

Permissions for Creating Views

To create views, you must have `sls.create` permission in the EDW instance, and you must have `sls.namespace` permission on the namespaces where you want the view to be created and where its underlying table is located.

Declaring the Columns in a View

The columns in a view are declared by the columns in the target clause of `<`*`select_subclause`*`>`. In the target clause you can:

- specify all columns from the table implicitly with an asterisk (`*`)

- specify columns from table explicitly with a column list

The data types of columns in the view are the same as the corresponding columns in the table.

Specifying View Columns Implicitly with an Asterisk

When you specify view columns with an asterisk (*) in the SELECT subclause, the view has the same set of columns as the underlying table. Their names and data types are identical to those in the table.

For example, assume a table named `myTable` has columns `ts`, `ClientDNS`, `Method`, and `URL`. The following CREATE VIEW statement creates a view named `myView` with the same set of columns.

```
CREATE VIEW myView AS
  SELECT *
    FROM MyTable
    DURING ALL
;
```

When the schema of the underlying table changes, the columns in a view that you declared with an asterisk in the target clause always conform to the underlying table.

Specifying View Columns Explicitly with a Column List

When you specify view columns with an explicit column list, the view has only the listed columns from the underlying table; other columns in the table are excluded from the view. The data types of view columns are identical to the corresponding columns in the table, but you can rename the view columns with the `AS` modifier keyword.

IMPORTANT:

- You should include the `ts` column in *every* view definition. A view that does not include this column would return the entire table. Most likely, such a view would fill your disk with unwanted information.
- When the specified view columns are identical to the table columns, the `AS` keyword is optional. When a specified view column is an expression, the `AS` keyword is required. You can use any valid SELECT statement expression to create a view.

For example, assume a table named `myTable` that has columns `ts`, `ClientDNS`, `Method`, and `URL`. The following CREATE VIEW statement creates a view named `myView` with the same set of columns, but applies the `_strlink` function to the `URL` column. Because the view columns include an expression, that column must be renamed. Because the `ClientDNS` column is named identically in the view as in the table, the renaming of this column is optional.

```
CREATE VIEW myView AS
  SELECT ts, ClientDNS AS Client, Method, _strlink(URL, ’Click Here’) AS
      anchor_tag
    FROM MyTable
    DURING ALL
;
```

NOTE:

- When you specify a column in the target clause as a computed expression, the data type of the column in the view is based on the data type of the computation.

- If additional columns are added to the underlying table, the additional table columns remain excluded from the view. If a column included in the view is removed from the underlying table, a query-time error states that the column cannot be found.
- When you query on a view in HawkEye AP Console always include the DURING ALL clause at the end of the main select. If you do not, the EDW will scan all data in the underlying tables.

RENAMING COLUMNS IN A VIEW

As illustrated above in Specifying View Columns Explicitly with a Column List, you can give the columns in the view different names than they have in the underlying table from which the view selects its data. The new names follow the same rules that apply to table and columns names. For more information, see “Specifying Names for Tables and Columns”, on page 142.

You can rename the view columns either with the AS keyword modifier or with a column list. In either case, you must explicitly list the view columns in the target clause of the SELECT subclause.

The example above uses the AS modifier keyword to rename some of the columns in the view. Alternately, you can specify a *<column_list>* with the column names you want in the view. The number and order of columns in *<column_list>* must match the number and order of columns in the target clause of the SELECT subclause. Enclose the list of column names that you declare for a view in parentheses. Separate column names with commas.

For example, the following statement creates a view named myView with the columns from myTable, and renames them in the view with a column list:

```
CREATE VIEW myView (Timestamp, Client, Method, URL) AS
  SELECT ts, ClientDNS, Method, Url
    FROM MyTable
    DURING ALL
;
```

IMPORTANT: You should not use a *<column_list>* to rename view columns if you use an asterisk (*) in the target clause of the SELECT subclause.

If you rename the same column in both the column list and with the AS modifier keyword in the target clause of the SELECT subclause, the name in the column list takes precedence.

Selecting a Subset of Table Columns or Rows for a View

Generally, you create views that expose only subsets of columns or rows from their underlying tables. You limit the columns in the target clause of the SELECT subclause. You limit the rows in the WHERE clause of the SELECT subclause.

For example, assume you have a table named syslog that stores log events from log sources that use the syslog protocol. The table contains log events from a variety of programs, including SSH and HTTP. This table has 7 columns: ts, Hostname, Progname, ProcessID, Message, IP, and upload_id. You want to create a view that exposes only three of these columns. Additionally, you want to limit the view to expose only those rows that show the syslog events from SSH.

The following statement creates a view from the syslog table that exposes only a subset of the columns and rows.

```
CREATE VIEW syslog_ssh (Timestamp, Host, Message) AS
  SELECT ts, Hostname, Message
  FROM syslog
  WHERE _strstr(_strlowercase(Progname), "ssh") > -1
  DURING ALL
;
```

The WHERE clause declares that the view shows only rows from the `syslog` table that contain `"ssh"` in the `Progname`. The view does not include the `Progname` column, because by its definition the view includes log events from a single program.

For another example, assume you have a table with all the web server log events for a particular region. You could create separate views that select rows only for one office in the region. If you create each office view in a separate namespace, you can give the security teams for each office access to their rows without exposing the rows from other offices.

Specifying and Querying Against Time Ranges in a View

Every view must conclude with the DURING clause. A SELECT statement that specifies a view in the FROM clause requires a DURING ALL clause if it is run from HawkEye AP Console.

IMPORTANT: When you create a view, ensure that the DURING clause behaves as you expect. A badly formed DURING clause can cause the query to return incorrect or no results. For more information, see Subqueries and Views and the DURING Clause in Chapter 10, “Sensage SQL” in the *Reporting Guide*.

Using Processing Directives when Creating a View

You can include processing directives, including the declarations for subqueries and Perl subroutines, anywhere within SELECT subclause of CREATE VIEW statements. For example:

```
CREATE VIEW myView AS
  WITH $table_name as "myNameSpace.myTable"
  SELECT *
    FROM $table_name
    DURING ALL
;
```

Subqueries and Perl subroutines are compiled by the EDW at the time you create the view. Subqueries and Perl subroutines execute at the time people query the view.

For more information, see Processing Directives in Chapter 10, “Sensage SQL” in the *Reporting Guide*.

Declaring a Union of Tables in a View

You can declare a view that selects its columns and rows from a union of tables. As with all UNION queries, the SQL query engine places the following restrictions on the UNION ALL clause:

- All the subselects within the SELECT statement must return the same number of target columns.
- Each target column must have the same data type in each subselect, respectively.

Declaring Union Views with the UNION ALL Clause

The following statement uses the UNION ALL clause to create a view of the union of the tables `syslog_a`, `syslog_b`, and `syslog_c`:

```
CREATE VIEW myUnionView AS
  SELECT ts, Hostname, Progname, ProcessID, Message, IP
    FROM syslog_a
    DURING ALL

  UNION ALL

  SELECT ts, Hostname, Progname, ProcessID, Message, IP
    FROM syslog_b
    DURING ALL

  UNION ALL

  SELECT ts, Hostname, Progname, ProcessID, Message, IP
    FROM syslog_c
    DURING ALL
;
```

For more information, see UNION ALL Clauses in Chapter 10, “Sensage SQL” in the *Reporting Guide*.

Declaring Union Views with _tablematch()

Assume that the namespace `myNameSpace` contains the tables `syslog_a`, `syslog_b`, and `syslog_c`. The following statement uses the `_tablematch()` function to create or replace a view of the same union as the previous example:

```
CREATE OR REPLACE VIEW myUnionView AS
  SELECT ts, Hostname, Progname, ProcessID, Message, IP
    FROM @_tablematch( "syslog_.*", "myNameSpace" )
    DURING ALL
;
```

The tables included in the union are computed by `_tablematch()` when people run queries against the view, not when you create the view. If you add table `syslog_d` to `myNameSpace` after you create the view, the view exposes rows from all tables that begin with `"syslog_"`.

For more information, see _tablematch() in Chapter 11, “SQL Functions” in the *Reporting Guide*.

Syntax for Renaming Views

```
RENAME VIEW [<old_namespace>.]<old_view_name> TO
    [<new_namespace>.]<new_view_name>
;
```

IMPORTANT: When you a rename a view, other views and reports that depend on it may no longer produce results. The query engine does not generate an error at the time of the change, but does generate an error when a query is run against the renamed view.

Permissions for Renaming Views

To rename views, you must have `sls.rename` permission in the EDW instance, and you must have `sls.namespace` permission on the namespace where the view you want to rename is located. If renaming the view results in moving it from one namespace to another, you must have `sls.namespace` permission on the target namespace, too.

Syntax for Dropping Views

```
DROP VIEW [<namespace>.]<view_name>
;
```

When you drop a view, you must identify fully the namespace in which the view is located. You can specify the namespace explicitly in the DROP VIEW statement, or you can specify it with the `--namespace` option of the `atquery` command.

IMPORTANT:

- When you a drop a view, other views and reports that depend on it may no longer produce results. The query engine does not generate an error at the time of the change, but does generate an error when a query is run against the dropped view.
- Dropping a view does not discard any log events from the EDW instance.

Permissions for Dropping Views

To drop views, you must have `sls.drop` permission in the EDW instance, and you must have `sls.namespace` permission on the namespace where the view you want to drop is located.

Syntax for Transferring Ownership of a View from One User to Another

```
GRANT OWNER ON <view_name> TO <user_name>
```

When a user change roles or is deleted from the system -- for example, because the user moves to another department or leaves the company - all VIEWs owned by the user are disabled and unusable, unless you transfer ownership to another user. A VIEW gets its permissions from its owner. Thus, after ownership is transferred, *view_name* gets its permissions from new owner *user_name*. A VIEW can have only one user; when you transfer ownership to another user, the first user loses ownership. You must have sls.admin permissions to execute this command.

Retrieving Information About a View

You can query the EDW to get the following information about each view in your EDW instance:

- query definition
- namespace
- creator

To get this information, you must query the `storage.raw_metadata` system table. Querying system tables is documented in "System Tables", on page 163.

The `storage.raw_metadata` system table contains the same columns as the `storage.metadata` system table, with the addition of the `data` column. The `data` column stores metadata only for those row types that require it. Each view will have at least three rows with values in this column: for the query, the namespace, and the creator. For a multi-host EDW instance, this table returns these three rows for each host in the instance. For information about the `storage.metadata` table, see "<namespace>.storage.metadata", on page 170.

When you query the `storage.raw_metadata` table for view information, the most relevant columns are `row_type`, `table_name`, and `data`. The following query, which was run from atquery, retrieves only these three columns and only those rows with text in the `data` column:

```
select row_type, table_name, data from storage.raw_metadata where data != ""
```

The example output below illustrates view information for a view named `PubsTest`.

```
| Results for SQL file >(standard input)< |
+--------------+----------+----------------------------------------------------+
| row_type |table_name| data
|
| (varchar) |(varchar) | (varchar)
|
+--------------+----------+----------------------------------------------------+
|VIEW_QUERY    |PubsTest  |CREATE VIEW Pubs.PubsTest (Timestamp, Client, Method,
URL) AS\n SELECT ts, ClientDNS, Method, Url\n FROM  websrv DURING ALL\n |

|VIEW_NAMESPACE|PubsTest |myNamespace
|
|VIEW_CREATOR |PubsTest |administrator
|
|+--------------+----------+----------------------------------------------------+
```

NOTE: When you rename a view, you change only its name and not the statement that defines it. Therefore, if you query the `storage.raw_metadata` system table on a view that has been renamed, the `table_name` column in the query results displays the new name, but the view-creation statement in the `data` column still specifies the name used when the view was created. You can compare the name of the view in the table_name column with the name specified in the `CREATE VIEW` statement to determine whether a view has been renamed.

The example output below illustrates view information for a renamed view. The output displays information about the `PubsView` view. As shown in the `data` column, this view was originally named `PubsTest`.

```
| Results for SQL file >(standard input)< |
+--------------+----------+----------------------------------------------------+
| row_type |table_name| data
|
| (varchar) |(varchar) | (varchar)
|
+--------------+----------+----------------------------------------------------+
|VIEW_QUERY    |PubsView |CREATE VIEW Pubs.PubsTest (Timestamp, Client, Method,
URL) AS\n SELECT ts, ClientDNS, Method, Url\n FROM  websrv DURING ALL\n |

|VIEW_NAMESPACE|PubsView|myNamespace
|
|VIEW_CREATOR |PubsView|administrator
|
|+--------------+----------+----------------------------------------------------+
```

Listing, Deleting, and Renaming Tables

In this topic:

- "Listing Tables", next

- "Deleting Tables", on page 157
- "Renaming Tables", on page 157

IMPORTANT: Before performing these activities, make sure you back up your data store. See "Backing Up and Restoring an EDW Table", on page 139 for details.

Listing Tables

To list the tables in a given installation, use

```
atview <cluster_list> tables --namespace= --user=administrator --pass=<password>
```

The `--namespace=` argument tells `atview` to list all tables in all namespaces, including the default namespace, `default`.

Also see "Examining the State of an EDW Data Store", on page 125.

Deleting Tables

To delete tables, use

```
atmanage <cluster_list> droptbl <tablename>
```

Once deleted, a table's data is permanently removed.

Also see "Defining Views with Sensage SQL", on page 149.

Renaming Tables

To rename a table, use

```
atmanage <cluster_list> renametbl <old_tablename> <new_tablename>
```

Also see "Defining Views with Sensage SQL", on page 149.

MONITORING AN EDW INSTANCE

The following topics give you general guidelines for monitoring your EDW cluster:

- "Monitoring CPU and Memory Usage", next
- "Verify Hosts Are Up and Running", on page 159

NOTE: Many of the messages reported by the EDW are informational and require no action.

Monitoring CPU and Memory Usage

CPU usage on each machine is a good indicator of the health of the cluster. Use `cltop` to monitor CPU usage. In particular:

- Check that the average CPU is below 50% during loads or queries. Don’t worry if you see *only* high CPU usage—this is due to the EDW process of compressing and decompressing the log data.

- Check for high CPU usage *and* no running tasks. This is an indication of poor CPU usage.

Memory usage is another excellent indicator on the state of your cluster. However, the `top` command will not correctly report the amount of free RAM. Instead, use the `free` command to return a more accurate number for available memory. To see if the cluster is running out of memory, examine the progress indicators returned from a query. If a progress indicator is consistently visible, with no more than a one-second pause, this typically means the cluster has sufficient memory.

NOTE: You must examine each host separately to see if any one host is running low on memory and slowing the cluster down.

```
# free
             total       used       free     shared    buffers     cached
Mem:       2074928    1784060     290868          0     164632    1256520
-/+ buffers/cache:     362908    1712020
Swap:      2031608        148    2031460

cltop Pubs_Instance
cltop - cltop 4.1, change #42464;
Initializing (ssh) root@piltdown.hq.myco.com
Initializing (ssh) root@poweredge04.hq.myco.com
Initializing (ssh) root@proliant05.hq.myco.com
* piltdown.hq:-Done- | poweredge04.hq:-Done- | proliant05.hq:-Done-
All done.
NODE NODE ..........CPU usage......... .............. RAM usage ............
NUM ROLES idle busy (user+system) total free used shared cache
NODE PROCESS VMEM RAM SHMEM PROC CPU INSTALL
NUM ID (MB) (MB) (MB) STAT used TIME PORT START NAME ACTION OTHER
04 28154 608.0 605.0 8.3 R N 45.1% 0:28 8072 10:19 1178817571 run 1 /opt/sensage/
latest/etc/sls/instance/Pubs_Instance/sensage_sls_Pubs_Instance r=ETL%2DEngine
e=RS%5FRUN t=BA07101495ACF887728D635C89885C69
05 21401 141.0 141.0 8.3 R N 24.9% 0:29 8072 10:21 1178817711 run 0 /opt/sensage/
latest/etc/sls/instance/Pubs_Instance/sensage_sls_Pubs_Instance r=ETL%2DEngine
e=RS%5FRUN t=BA07101495ACF887728D635C89885C69
04 28150 217.0 216.0 7.0 R N 7.7% 0:08 8072 10:19 1178817571 run 1 /opt/sensage/
latest/etc/sls/instance/Pubs_Instance/sensage_sls_Pubs_Instance
r=Addamark%2ELoad%2EFarm t=BA07101495ACF887728D635C89885C69
04 28143 20.2 14.0 9.2 S 2.4% 0:02 14 Dec31 14 14 0 /opt/sensage/latest/java/bin/
java -cp /opt/sensage/latest/../3.7/lib/java/cli/cli-3.7.jar
com.sensage.cli.Ssiload --namespace=blue poweredge04:8072 blue /tmp/
blue_websense.ptl /tmp/SG_WA_Sensage__100922070000.log
04 28149 14.1 13.0 6.7 S N 0.9% 0:00 8072 10:19 1178817571 run 0 /opt/sensage/
latest/etc/sls/instance/Pubs_Instance/sensage_sls_Pubs_Instance r=StdinContent
t=BA07101495ACF887728D635C89885C69
04 28148 14.0 12.0 6.7 S N 0.4% 0:00 8072 10:19 1178817571 run 0 /opt/sensage/
latest/etc/sls/instance/Pubs_Instance/sensage_sls_Pubs_Instance r=StdinTrans
t=BA07101495ACF887728D635C89885C69
04 28145 25.6 25.0 4.9 S 0.4% 0:00 8072 May 1178578523 init 1 /opt/sensage/
latest/etc/sls/instance/Pubs_Instance/sensage_sls_Pubs_Instance
04 28146 16.6 16.0 8.4 S N 0.4% 0:00 8072 10:19 1178817571execute 1 /opt/sensage/
latest/etc/sls/instance/Pubs_Instance/sensage_sls_Pubs_Instance
r=Addamark%2ELoad e=RS%5FRUN t=BA07101495ACF887728D635C89885C69
```

```
04 28144 20.2 14.0 9.2 S 0.4% 0:00 14 Dec31 14 14 0 /opt/sensage/latest/java/bin/
java -cp /opt/sensage/latest/../3.7/lib/java/cli/cli-3.7.jar
com.sensage.cli.Ssiload --namespace=blue poweredge04:8072 blue /tmp/
blue_websense.ptl /tmp/SG_WA_Sensage__100922070000.log
KEY:
- PROC STAT:
D uninterruptible sleep (usually IO)
R runnable (on run queue)
S sleeping
T traced or stopped
Z a defunct ('zombie') process
W has no resident pages
< high-priority process
N low-priority task
L has pages locked into memory
```

Verify Hosts Are Up and Running

To see if all the hosts in your EDW instance are up and running, submit an `atquery` command to display the `cluster_properties` table in the `system` namespace of your EDW:

```
atquery [machine name]:[port] -e "select * from system.cluster_properties"
--namespace= | grep NODE_INFO
```

The `-e` option tells `atquery` to read the SQL statement that follows it.

In the results, look for the `NODE_INFO` lines. Each host in your EDW instance should have an entry. For example, if you have a 10 hosts in your EDW instance, you should expect 10 records back from this query.

MONITORING DISK USAGE

You can monitor disk usage in the EDW by running one of the EDW System reports, available as part of the Foundation Analytics report package. For more information, see following reports in the *Analytics Guide.*

- EDW System Disk Space Usage Alerts in Chapter 4, “Foundation Analytics Report Package”
- EDW System Disk Usage Per Node in Chapter 4, “Foundation Analytics Report Package”
- EDW Percent Disk Space Usage in Chapter 4, “Foundation Analytics Report Package”

You run these reports in HawkEye AP Console. For more information, see Chapter 3: Running, Viewing, and Managing Reports in the *Reporting Guide.*

International Support in the EDW

The EDW supports the storage of *international characters*, which are characters other than those in the U.S. English alphabet and some of its punctuation marks. For example, the Spanish word “España” contains the international character “ñ”. The EDW stores all text values with UTF-8 character encoding to ensure that international characters are stored and queried successfully.

This section describes these topics:

- “Character Sets”, next

- “Character Encoding Schemes”, on page 160
- “Character Encoding and Computer Fonts”, on page 161

Character Sets

A *character set* is a fixed set of characters and symbols stored in computer systems and transmitted across computer networks. The fixed set of characters is sometimes referred to as the *character repertoire* of a character set. For example, the character repertoire of the ASCII character set is the 26 upper- and lower-case letters of U.S. English, the numerals 0-9, some punctuation marks, and special symbols.

To facilitate storage and transmission, each member of a character repertoire is assigned a unique decimal number. This number is sometimes referred to as the *code position* or *code point* for a character. For example, the code point for the letter “A” in the ASCII character set is 65. The code point for “B” is 66, and so on.

Character sets are often represented in tables. Thus, a code point is the position of a character within a characer-set table. The following table shows a portion of the ASCII character-set table.

Code Point	Character Name	Graphic Representation
62	Greater-than Symbol	
63	Question Mark	
64	Commercial At	
65	Upper-case “A”	
66	Upper-case “B”	
67	Upper-case “C”	
68	Upper-case “D”	
69	Upper-case “E”	

Character Encoding Schemes

A *character encoding scheme* is the method by which the characters in a particular character set are represented with bit patterns for storage in computers and transmission across computer networks. A single character set can be represented by different encoding schemes with different bit patterns.

Encoding schemes are often characterized by the way in which their bit patterns represent characters:

- **Single-byte encoding schemes**—each character in the character set is represent by a unique pattern of bits within a single byte. The character sets of single-byte encoding schemes are limited to 255 unique characters and symbols, excluding bytes with all bits set to zero. ISO 8859-1 is an example of a single-byte encoding scheme.
- **Double-byte encoding schemes**—each character in the character set is represent by a unique pattern of bits in a two-byte “word.” The character sets of double-byte encoding schemes are limited to roughly 65,000 unique characters and symbols. Unicode character

encoding is an example of a double-byte encoding scheme. Unfortunately, as an encoding scheme, it cannot represent all the characters in the ever-expanding Unicode character set.

- **Variable-width encoding schemes**—each character in the character set is represent by a unique pattern of bits in a sequence of bytes. The character sets of variable-width encoding schemes are essentially unlimited in terms of the number of unique characters and symbols that can be represented. Some characters are represented by single-byte bit patterns, some by two-byte bit patterns, and so on. UTF-8 character encoding is an example of a variable-width encoding scheme that can represent all the characters in the Unicode character set and any additional characters that are added in the future.

Encoding Schemes Used by the EDW

There are too many encoding standards in use to enumerate them all. The ones that follow are relevant for understanding how the EDW stores, processes, and exports international character data.

Encoding Standard	Encoding Characteristics
ASCII / ISO 646	7-bit encoding, with 128 code points; a fixed-width encoding scheme with characters represented in computer systems by single, 8-bit bytes. IS0 646 encoding can represent the upper- and lower-case letters of the English alphabet, as well as a few English punctuation marks and special symbols. **NOTE:** In 8-bit byte streams, ASCII encoding generally sets high-order bits to 0. An ASCII byte stream is indistinguishable from an UTF-8 byte stream.
ISO 8859-1	8-bit encoding, with 255 code points; a fixed-width encoding scheme with characters represented by single, 8-bit bytes. ISO 8859-1 encoding can represent the Latin alphabet and many Western European variants. It cannot represent international characters from other languages, such as Cyrillic languages, Arabic, or Japanese. **NOTE:** When an ISO 8859-1 byte stream encodes only English text, the stream is indistinguishable from an ASCII or a UTF-8 byte stream.
Unicode / ISO 10646	A 16-bit encoding, with tens of thousands of code points; a fixed-width encoding scheme with characters represented by single, 16-bit bytes. ISO 10646 can represent characters from many European and non-European languages but has some limitations with Asian languages; although it has roughly 60,000 code points, the Unicode character set has 90,000 characters in its repertoire that need representation in computers.
UTF-8	Variable-width encoding scheme for Unicode, where the high-order bits of the first bytes of characters indicate how many of the bytes that follow are part of the same character. For example, if the high-order bit of the first byte in a character is 0, none of the following bytes are part of the character—it a single-byte character. **NOTE:** When an UTF-8 byte stream encodes only English text, the stream is indistinguishable from an ASCII byte stream.

The EDW stores all character data with UTF-8 character encoding. For character data in the EDW that contains only English text, the character encoding is identical to ASCII encoding.

Character Encoding and Computer Fonts

Character encoding determines how computers store human-readable text internally and transmit it across computer networks. At some point, people want to read the text that computers store and transmit. *Computer fonts* are typefaces that determine how computer screens and printers display human-readable text from the machine-readable text inside computers.

For example, the first letter in the English alphabet is the letter “A”. Depending on the computer font used to display an “A”, it may appear as A.

Computer fonts are designed to display specific character sets or subsets of a character set. For example, most computer fonts can display the ASCII character set. In contrast, there are no computer fonts that can display all the characters in the Unicode character set. With Unicode, fonts are generally designed for a linguistic subset, such as Japanese.

Sometimes a block of text contains characters that a computer font is not designed to display. For example, there might be some Japanese characters in a block of English text. Generally, if a computer font cannot display a particular character, it displays a placeholder character instead, such as hollow rectangle. If you see these placeholder characters, you need to install or activate a different font. Consult the documentation for you computer or printer to learn how.

System Tables

This section contains the following topics:

Overview

You can query the EDW to get information about your HawkEye AP system, such as the state of an EDW instance or all defined roles and which users have been assigned to them. When you query for this information, the EDW dynamically formats the result data as a standard table.

Because these tables return system information, HawkEye AP calls them *system tables*. Because the data returned is not saved on disk but is dynamically retrieved and formatted as a standard table, HawkEye AP also refers to them as *virtual tables*. The virtual nature of these tables is apparent when you query the EDW for all tables in your instance by running the `atview` command; for example:

```
atview --user=administrator --pass=<passwd> --namespace='' <cluster_list> tables
```

The above command does not return any system tables.

Because the system tables are virtual, you cannot insert data directly into them. For example, you cannot run `atload` to load a new user into `system.users`.

To explore the state of your system, run the `atview` command; for more information, see “Examining the State of an EDW Data Store”, on page 125 and “Monitoring an EDW Instance”, on page 157.

To explore authorization and authentication data, run the `atquery` command against specific system tables; for more information, see “Listing Users, Roles and Permissions”, on page 240.

This topic documents the schema of each system table.

IMPORTANT:

- To enable the EDW to integrate with an external authentication authority (such as Active Directory), several system tables required additional columns. To prevent incompatibility with existing scripts that reference these tables, HawkEye AP did not add columns to the existing tables. Instead, HawkEye AP created a new expanded version of these tables. The name of each expanded table ends in "2".
- The values in authentication tables are retrieved from the external authentication authority. You do not directly enter these values into the EDW by running atmanage.

system.properties

This table provides configuration information about a specific host in your EDW instance. For example, run atquery against this table to return the version of HawkEye AP running and the date that HawkEye AP version was installed.

Column Name	Data Type	Description
row_type	varchar	type of data represented by this row
basic_property_name	varchar	property name
basic_property_value	varchar	property value

system.cluster_properties

This table provides configuration information about an entire EDW instance. For example, run atquery against this table to return the name, host (node), port, and block size of every host in your EDW instance.

Column Name	Data Type	Description
row_type	varchar	type of data represented by this row, for example. columns or tables
node_index	int32	identifies the host in the cluster
dsm_sib_idx	int32	identifies sibling relationships between hosts
cluster_name	varchar	the EDW instance name
node_name	varchar	the host "name" (typically the same as the DNS host name, but not required to be)
node	varchar	the DNS host name for a host, can vary if the self-reported name differs from the name that other hosts use to address it
port	int32	port number the host is listening on
device	varchar	device number, for accurately detecting disk-sharing, and avoid misreporting free space
block_size	int32	size (in bytes) of a block on this device
num_blocks	int64	total capacity (in blocks) of this device

Column Name	Data Type	Description
num_bytes	int64	total capacity (in bytes) of this device
count	int32	number of the instances of this type of item

system.task_list

This table provides information about tasks running in your EDW instance. For example, run `atquery` against this table to return the start time of running tasks.

Column Name	Data Type	Description
row_type	varchar	type of information being returned: NODE or TASK
node	varchar	host the record relates to
port	int32	port of the host the record relates to
_systaskid	varchar	internal ID that identifies the task in the system
method_name	varchar	name of the internal method being run
runqueue	varchar	priority level of the queue that is running
purpose	varchar	additional task information (where known)
_extid	varchar	external ID that the client program has given this task
at_top	bool	if true, this is the process running as the "master" for this operation
start_time	varchar	time the task started
finish_time	timestamp	time the task ended (flushes every few minutes)
state	timestamp	current state, for example, `RUNNING`

system.users

This table provides the unique identifier of each user in your EDW instance. It also returns each user's status (enabled or disabled).

Column Name	Data Type	Description
uid	varchar	unique user identifier of every user in the system, including administrator, guest, system, and individual users
isEnabled	boolean	indicates whether the user has been enabled

system.users2

This table provides information about the users in your EDW instance. The source of the information is an external authentication authority.

Column Name	Data Type	Description
name	varchar	login name
local_id	varchar	user identifier that is unique within a specific EDW instance
guid	varchar	user identifier that is unique across all EDW instances
isEnabled	boolean	indicates whether the user has been enabled to use the system
real_name	varchar	user's full name
mail	varchar	user's email address
pager	varchar	user's pager number
description	varchar	text description of the user

system.userroles

This table returns user/role relationships: every role in your EDW instance and the users assigned to each role. Roles with multiple users display multiple times in the role column (as many rows as the number of users assigned to them).

Column Name	Data Type	Description
role	varchar	name of every role in the system, including administrator, guest, system, and roles created for individual users
uid	varchar	unique identifier of each user assigned to a specific role

system.userroles2

This table returns user/role relationships: every role in your EDW instance, the users assigned to each role, and the group identifiers of each role and user. The source of the information is an external authentication authority.

Column Name	Data Type	Description
role_name	varchar	name of every role in the system, including administrator, guest, system, and roles created for individual users
role_id	varchar	role identifier that is unique within a specific EDW instance
role_guid	varchar	role identifier that is unique across all EDW instances
user_name	varchar	login name
user_id	varchar	user identifier that is unique within a specific EDW instance
user_guid	varchar	user identifier that is unique across all EDW instances

system.roles

This table provides the unique identifier of each role in your EDW instance. It also returns each role's status (enabled or disabled).

Column Name	Data Type	Description
role	varchar	unique identifier of every role in the system, including administrator, guest, system.
isEnabled	boolean	indicates whether the role has been enabled

system.roles2

This table returns every role in your EDW instance and its local and global identifiers. It also returns each role's status (enabled or disabled). The source of the information is an external authentication authority.

Column Name	Data Type	Description
name	varchar	name of every role in the system, including administrator, guest, system, and roles created for individual users
local_id	varchar	role identifier that is unique within a specific EDW instance
guid	varchar	role identifier that is unique across all EDW instances
isEnabled	boolean	indicates whether the role has been enabled

system.permissions

This table returns every permission that can be granted to a role in your EDW instance.

Column Name	Data Type	Description
permission	varchar	name of every permission in the system
mixMethod	boolean-or string-set	indicates whether the permission has been enabled

system.rolepermissions

This table returns role/permission relationships: every role in your EDW instance and the permissions assigned to each role. Roles with multiple permissions display multiple times in the role column (as many rows as the number of permissions assigned to them). The table also indicates whether each permission has been enabled for each role.

Column Name	Data Type	Description
role	varchar	name of every role in the system
permission	varchar	name of every permission in the system
value	varchar	this column has internal significance only

system.rolepermissions2

This table returns role/permission relationships: every role in your EDW instance and the permissions assigned to each role. Roles with multiple permissions display multiple times in the role column (as many rows as the number of permissions assigned to them). The table also returns the globally unique role identifier and indicates whether each permission has been enabled for each role. The source of the information is an external authentication authority.

Column Name	Data Type	Description
role	varchar	name of every role in the system, including administrator, guest, system, and roles created for individual users
role_id	varchar	role identifier that is unique within a specific EDW instance
role_guid	varchar	role identifier that is unique across all EDW instances
permission	varchar	name of every permission in the system
value	varchar	this column has internal significance only

system.upload_info

This table allows administrators to check on upload status in the system. When the system is healthy, this table returns one row for each upload (using its unique identifier, **uploadid**). When the upload data is not consistent across all the hosts in the instance, this table returns rows with a **consistent** value of `false`*.* In this case, the table may return multiple rows for a given **uploadid** (all of which are marked as inconsistent).

Column Name	Data Type	Description
uploadid	varchar	The unique identifier for a particular load
original_tablename	varchar	The fully qualified name of the table that was the target of this load at the time of the load operation
ptl_signature	varchar	The digital signature of the PTL Information used for this load
started	timestamp	The time at which the load was started
completed	timestamp	The time at which the load completed
user	varchar	The username of the person who performed the load
min_ts	timestamp	The minimum timestamp in the loaded data
max_ts	timestamp	The maximum timestamp in the loaded data
line_count	int64	The number of lines in the source log data
parse_count	int64	The number of lines that were successfully parsed by the PTL's regular expressions (regexes)
load_count	int64	The number of rows that were loaded into the table
successful	boolean	Indicates whether the load completed successfully
client_signature	varchar	The client-side signature that was generated for this load
client_signature_method	varchar	Indicates how the client generated the signature (a value of `PRIVATE` indicates it is client-specific)

Column Name	Data Type	Description
client_blob	varchar	The additional data that the client wanted associated with this load
current_tablename	varchar	The fully qualified name of the current name of the table that received this load
consistent	boolean	Indicates whether information about this uplaod was consistent across the cluster

system.raw_upload_info

This table allows a support person to see the raw data that is distributed across the system in order to troubleshoot an 'inconsistency' problem in the `upload_info` data.

Column Name	Data Type	Description
uploadid	varchar	The unique identifier for a particular load
original_tablename	varchar	The fully qualified name of the table that was the target of this load at the time of the load operation
ptl_signature	varchar	The digital signature (MD5) of the PTL Information used for this load
started	timestamp	The time at which the load was started
completed	timestamp	The time at which the load completed
user	varchar	The username of the person who performed the load
min_ts	timestamp	The minimum timestamp in the loaded data
max_ts	timestamp	The maximum timestamp in the loaded data
line_count	int64	The number of lines in the source log data
parse_count	int64	The number of lines that were successfully parsed by the PTL's regular expressions (regexes)
load_count	int64	The number of rows that were loaded into the table
successful	boolean	Indicates whether the load completed successfully
client_signature	varchar	The client-side signature that was generated for this load
client_signature_method	varchar	Indicates how the client generated the signature (a value of `PRIVATE` indicates it is client-specific)
client_blob	varchar	The additional data that the client wanted associated with this load
partition	varchar	Specifies either the Primary or Secondary directory
current_tablename	varchar	The fully qualified name of the current name of the table that received this load
node	varchar	The logical name of the host that this row of data came from

<*namespace*>.storage.metadata

This table provides information about the tables and columns in the specified namespace. If you specify an empty namespace ("") when you run `atquery` against this table, the table returns all tables and columns throughout your EDW instance.

Column Name	Data Type	Description
row_type	varchar	type of data represented by this row. 'COLUMN_INFO'=
cluster_name	varchar	the installation name
namespace	varchar	the complete namespace name for this table
table_name	varchar	the table name
column_name	varchar	the column name
column_type	varchar	the data type for this column
primary_key	boolean	is this the primary key? typically, only the 'ts' (timestamp) column is a key
min_value	varchar	the minimum value for this column (currently, only available for the 'ts' column)
max_value	varchar	the maximum value for this column (currently, only available for the 'ts' column)
error_type	varchar	message(s) in case there are inconsistencies across hosts in the cluster
column_filter	varchar	the name of each filter on the column, separating multiple filters with commas

<*namespace*>.storage.raw_metadata

This table provides configuration information about an entire EDW instance. For example, you can query this table to return the name, host (node), port and relationship of every host in your EDW instance.

Column Name	Data Type	Description
row_type	varchar	type of data represented by this row. 'COLUMN_INFO'=
node_index	int32r	identifies the host in the cluster
dsm_sib_idx	int32	identifies sibling relationships between hosts
cluster_name	varchar	the EDW instance name
node_name	varchar	the host "name" (typically the same as the DNS host name, but not required to be)
node	varchar	the DNS host name for a host, can vary if the self-reported name differs from the name that other hosts use to address it
port	int32	port number the host is listening on.
partition	varchar	Specifies either the Primary or Secondary directory

Column Name	Data Type	Description
namespace	varchar	the complete namespace name for this table
table_name	varchar	the table name
column_name	varchar	the name of each filter on the column, separating multiple filters with commas
column_type	varchar	the column name
primary key	boolean	False because there is no TS
column filter	varchar	the name of each filter on the column, separating multiple filters with commas
error_type	varchar	message(s) in case there are inconsistencies across hosts in the cluster

<*namespace*>.storage.raw_metadata_with_ts

This table provides configuration information about an entire EDW instance. For example, you can query this table to return the name, host (node), port and relationship of every host in your EDW instance.

Column Name	Data Type	Description
row_type	varchar	type of data represented by this row. 'COLUMN_INFO'=
node_index	int32r	identifies the host in the cluster
dsm_sib_idx	int32	identifies sibling relationships between hosts
cluster_name	varchar	the EDW instance name
node_name	varchar	the host "name" (typically the same as the DNS host name, but not required to be)
node	varchar	the DNS host name for a host, can vary if the self-reported name differs from the name that other hosts use to address it
port	int32	port number the host is listening on.
partition	varchar	Specifies either the Primary or Secondary directory
namespace	varchar	the complete namespace name for this table
table_name	varchar	the table name
column_name	varchar	the name of each filter on the column, separating multiple filters with commas
column_type	varchar	the column name
primary key	boolean	T if TS value is present, F if TS value is not present.
column filter	varchar	the name of each filter on the column, separating multiple filters with commas
min_value	varchar	the minimum value for this column
max_value	varchar	the maximum value for this column
error_type	varchar	message(s) in case there are inconsistencies across hosts in the cluster

<namespace>.<tablename>.storage.metadata

For each table, there exists a virtual table named `<tablename>.storage.metadata` which contains information about the columns in this table. Prefix the table with a namespace to query a table from a specific namespace other than the default.

Column Name	Data Type	Description
row_type	varchar	type of data represented by this row. 'COLUMN_INFO'=
cluster_name	varchar	the EDW instance name
namespace	varchar	the complete namespace name for this table
table_name	varchar	the table name
column_name	varchar	the column name
column_type	varchar	the data type for this column
primary_key	boolean	indicates whether this the primary key; typically, only the 'ts' (timestamp) column is a key
min_value	varchar	the minimum value for this column (currently, only available for the 'ts' column)
max_value	varchar	the maximum value for this column (currently, only available for the 'ts' column)
error_type	varchar	message(s) in case there are inconsistencies across hosts in the cluster

MODIFY TABLE SCHEMA - ALTERTBL

Use `altertbl` to modify the schema of an existing table. The previous command `clchgschema,` is deprecated. The new command enables you to execute the following:

- Add new empty columns
- Remove existing columns
- Apply a perl function to an input column that creates one or more output columns.

Synopsis

<sensage bin>/unsupported/altertbl <table name> <change-file> <sls config file> [executable name]

Description

```
<table name>
```

The fully qualified table name. A table named test in the default.test namespace, for example, would be specified as default.test.test.1

```
<change file>
```

A text file containing a list of instructions for `altertbl`. The files must be readable by the sensage user.

```
<sls config file>
```

The full path to the athttpd.conf file. If your EDW instance were named testsls, for example, a path could be similar to the following:

```
/opt/sensage/latest/etc/sls/instance/testsls/athttp.conf
```

```
[executable file]
```

A program that takes a single argument and returns a comma separated list of output values. This is required if your change file includes the OUTPUT directive.

The `altertbl` command must be run on each host and must run when the EDW is down, because it modifies table-level schema files.

IMPORTANT: You must run `altertbl` as sensage user, not as root. Running `altertbl` as root causes the datastore to be modified with root ownership. If you run as root user by mistake, execute `chown <sensageuser>:<sensageuser>`over the dsroot to reclaim all the new column files.

Adding or Dropping Columns

Following is the command to add an empty column to the datastore:

```
ADD <column name> BOOL|INT32|INT64|FLOAT|TIMESTAMP|VARCHAR
```

Note that when you add a column to an existing table using ADD, existing rows of data have an empty value where they intersect the new column.

To drop an existing column, add the following command to the change file.

```
DROP <column name>
```

Example

Add an empty `VARCHAR` column named `newcolumn,` and drop an existing column named `deleteme,` by adding the following to your change file:

```
ADD newcolumn VARCHAR
DROP deleteme
```

Populating New Columns using a Function

You have the ability to add columns that are pre-populated with data, as follows:

1 Create a command line executable (in any language) that takes a single piece of data as input and puts out a comma-separated list as output.

2 Execute this function against every row in the table to produce new columns. In the change file you must specify one input column and some number of output columns as follows:

```
INPUT <column name>
```

```
  OUTPUT <column name> <type>
```

Example

You have an existing column named PHONE that contains the complete phone number in the form *nnn-nnn-nnnn* and you want to create a new column, AREA CODE, that contains the first three digits of the phone number, you can create a Perl script that accepts a phone number and generates the three-digit area code. Add the following to the change file:

```
INPUT PHONE
OUTPUT AREACODE int32
```

Now execute the altertbl command specifying the Perl script as the optional executable. The new column is added, populated with the correct area code in each row.

NearLine Storage: Considerations

When adding empty columns to a table archived in NearLine Storage, consider the following:

Empty columns you add are stored locally, regardless of where the original leaf node is archived. As empty columns, their footprint on disk is not large.

Columns added using the INPUT or OUTPUT tags, the new, populated columns are stored locally. They cannot be archived because the leaf node is already archived.

If you delete columns, the altertbl command does not remove those columns from NearLine storage, although the data store does not continue to reference the column. You won't be able to query the data, but it is stored in that location until the leaf node is retired.

CHAPTER 5

Recovering a Failed EDW Node

This section describes the process for handling a failed EDW node and includes the following topics:

- “Overview”, next
- “Restoring an EDW Node Back into a Cluster”, on page 177
- “Sample cluster.xml File for 3-Node HawkEye AP”, on page 182

OVERVIEW

If a EDW node in your HawkEye AP instance fails, you can restore functionality using the procedures in this section. When one of the EDW nodes fails, you can configure your HawkEye AP instance to operate in a degraded mode, without the failed EDW node. In this mode of operation, you can continue to load data into the SLS and run queries against the data, but you can not perform meta-data operations such as adding, or dropping tables or views. After configuring this limited mode, you can then restore the failed EDW node to restore the HawkEye AP instance to full functionality. Alternately, if an EDW node fails, you can stop operations on the entire HawkEye AP instance and then restore or replace the failed EDWnode. Because each EDW node has a duplicate copy of its data store located on another node, no data is lost.

NOTE: A small amount of data may have been loaded to the destination table. If it has, all data uniquely identified by the load’s **upload_id** is marked as unsuccessful. Use `atretire` to remove this data; for more information, see “Retiring Data”, on page 130.

CONFIGURING THE EDW TO OPERATE WITH A FAILED EDW NODE

To configure a HawkEye AP instance to operate with a failed EDW node

1 Login to the hosts where the Console Manager is located as `root` using `ssh` and execute the following command to stop your HawkEye AP instance:

```
clsetup stop sensage
```

2 Determine which EDW node is referenced by the hosts where the Console Manager is located. The hosts where the Console Manager is located always connects first to one of the EDW nodes in the cluster, usually SLS01.

a Login to the hosts where the Console Manager is located as `root`, using `ssh`.

b Open the following file in a text editor:

```
<Sensage_Home>/latest/etc/sysconfig/sensage
```

c Locate the following property: `SENSAGE_SLS_HOST`, for example:

```
SENSAGE_SLS_HOST=sls01.mydomain
```

The argument of the property indicates which EDW node is referenced by the hosts where the Console Manager is located (`SLS01`, in the above example). If this is the EDW node that has failed, continue with Step d. If the node that has failed is not the node referenced by the hosts where the Console Manager is located, skip to Step 6 on page 176.

d Change the argument to reference a different EDW node in your cluster, for example:

```
SENSAGE_SLS_HOST=sls03.mydomain
```

e Save the file and exit the editor.

3 Open the following file for editing: `<Sensage_Home>/latest/etc/collector/config.xml`

a Use a search and replace function to replace *all* (there will likely be more than 20) instances of `edw01.mydomain` with the new hostname.

For example, replace this line:

```
<SLSInstance>edw01.mydomain:8072</SLSInstance>
```

with this line:

```
<SLSInstance>edw03.mydomain:8072</SLSInstance>
```

b Save the file.

4 Open the following file for editing: `<Sensage_Home>/latest/etc/controller/slsconfig.prop`

a Use a search and replace function to replace *all* instances of `sls01.mydomain` with the new host name.

b Save the file.

5 Switch to the following directory

```
cd <Sensage_Home>/latest/etc/sls/instance/<mysls>/
```

6 Edit the `cluster.xml` file located in this directory to mark the offline EDW node as "`down`".

To mark an EDW node as "`down`", change the value of the node's status attribute from "`active`" to "`down`". In the example text below, the highlighted status value for SLS02 has already been changed.

```
<node urlspace="/cgi-app/xmlrpc" status="down"
name="EDW02.hq.myco.com:7efe68af5f5c60e34f781203b030d7d5" port="8072"
host="EDW02.hq.myco.com">
<role name="Master"/>
<role name="QP"/>
<role name="DSM"><sibling
name="EDW03.hq.myco.com:7efe68af5f5c60e34f781203b030d7d5"/>
</role>
<role name="Node"/>
</node>
```

See "Sample cluster.xml File for 3-Node HawkEye AP", on page 182 for a sample of the `cluster.xml` file for a 3-Node HawkEye AP instance.

7 Use the `clsync` utility to synchronize the `cluster.xml` file to the other nodes. See “Copying Files and Directories to Each Host (clsync)”, on page 69. Specify all nodes *except the* hosts where the Console Manager is located *and the down node* with the `--hosts` option. This step informs all other hosts in the SLS that the node is “down”. You must run the clsync utility as the `lms` user.

For example, execute the following commands:

a `su lms`

b `cd <Sensage_Home>/latest/etc/sls/instance/<mysls>/`

c `clsync --hosts=edw01,edw03,edw04,edw05 cluster.xml`

d `exit`

IMPORTANT: You must run the `clsync` utility from the directory where the `cluster.xml` file is located.

8 Restart the hosts where the Console Manager is located by executing the following command:

```
service sensage restart
```

9 Replace the down node with a new or reconfigured node, see “Restoring an EDW Node Back into a Cluster”, next.

Restoring an EDW Node Back into a Cluster

The procedure below requires copying primary and secondary data stores of good EDW nodes to a new EDW node. Copying the correct data stores requires understanding the layout of EDW nodes in an EDW instance. This section describes the logical arrangement of the data stores as they are configured in a new system. Since this configuration may have been changed, you need to find out the exact locations of a node’s secondary data store by examining the `cluster.xml` file on the hosts where the Console Manager is located.

To find out the location of each EDW node’s secondary data store

1 Log into any EDW node as `root`, using `ssh`.

2 View the contents of the following file: `<Sensage_Home>/latest/etc/sls/instance/<mysls>/cluster.xml`. A sample version of this file is displayed in Figure 5-1 below.

Figure 5-1: Sample cluster.xml File

```
<clusters>
<cluster name="mysls" uuid="b31bd06ca6663e44">
<node urlspace="/cgi-app/xmlrpc" status="active"
  name="edw01.mydomain:b31bd06ca6663e44" port="8072"
  host="edw01.mydomain">
  <role name="Master"/>
  <role name="QP"/>
  <role name="DSM"><sibling name="edw02.mydomain:b31bd06ca6663e44"/>
  </role>
  <role name="Node"/>
</node>
```

```
<node urlspace="/cgi-app/xmlrpc" status="active"
  name="edw02.mydomain:b31bd06ca6663e44" port="8072"
  host="edw02.mydomain">
  <role name="Master"/>
  <role name="QP"/>
  <role name="DSM"><sibling name="edw03.mydomain:b31bd06ca6663e44"/>
  </role>
  <role name="Node"/>
</node>

<node urlspace="/cgi-app/xmlrpc" status="active"
  name="edw03.mydomain:b31bd06ca6663e44" port="8072"
  host="edw03.mydomain">
  <role name="Master"/>
  <role name="QP"/>
  <role name="DSM"><sibling name="edw01.mydomain:b31bd06ca6663e44"/>
</role>
  <role name="Node"/>
  </node>

</cluster>
</clusters>
```

3 The XML in this file contains a `<node>` element for each EDW node in your cluster. The `name` attribute of this element indicates the EDW node. Within the `<node>` element, there is also a `<sibling>` element. The `name` attribute of the `<sibling>` element indicates where the secondary data store for this node is located. For example, in the "Sample cluster.xml File for 3-Node HawkEye AP", on page 182, The 3 EDW nodes have the following siblings:

EDW Node	Sibling Node
EDW01	EDW02
EDW02	EDW03
EDW03	EDW01

From the above information, you can determine the following layout of primary and secondary data stores:

	EDW01	EDW02	EDW03
Primary Data store	EDW01	EDW02	EDW03
Secondary Data store	EDW02	EDW03	EDW01

Assume the instance with a down EDW node contains five nodes. Figure 5-2 illustrates these nodes and the data stores they contain.

Figure 5-2: Five-node EDW **instance with one down EDW node**

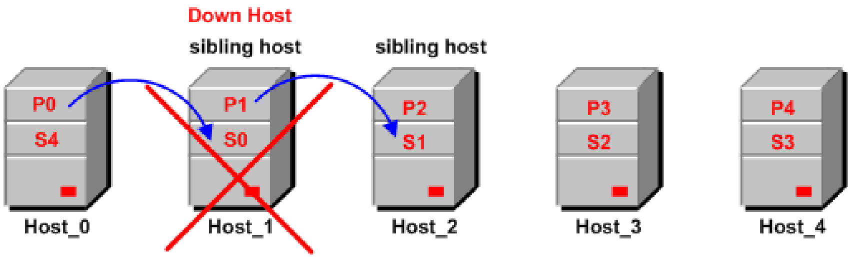


Figure 5-2 illustrates **EDW02** as the down node. The down node stores the secondary data store (**S0**) for **EDW01**. **EDW03** stores the secondary data store (**S1**) for the down node. Restoring the data requires creating a new SLS node and copying the data stores to it, as illustrated in Figure 5-3.

Figure 5-3: Five-node EDW instance with one new EDW node

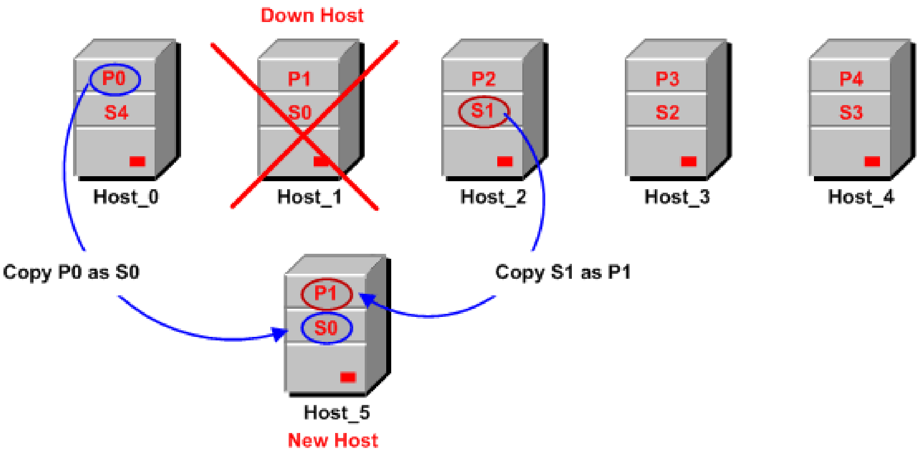


Figure 5-3 illustrates copying:

- Primary data store (**P0**) from **EDW01** to **EDW02**, where it is stored as the secondary data store (**S0**) for **EDW01**.
- Secondary data store (**S1**) from **EDW03** to **EDW02**, where it is stored as the primary data store (**P1**) for **EDW02**.

IMPORTANT: As soon as you lose an EDW node, make backup copies of the data that had been stored on the down EDW node. In other words, make backup copies of the Primary and Secondary data stores from the down node's sibling nodes.

To restore an SLS node to a Cluster

IMPORTANT: Hexis Cyber Solutions recommends that you contact Hexis Cyber Sol;utions Technical Support before attempting the following procedure.

The procedure below uses EDW node names from the illustrations above. Change the `--hosts` parameter to match your environment.

1 Select a new host to replace the down host and then install and configure the EDW on it:

```
install --hosts=Host_5 --prefix=<Sensage_Home>
```

2 Create HawkEye user and run `clsetup`:

```
useradd lms; \
clsetup configure sls --hosts=Host_5 --ldap-instances=<ldap_host from existing
installation>"
```

IMPORTANT: As root, run the `install` script that appears in the top-level of the directory where you extracted the HawkEye AP tar file. For detailed information on installing and configuring HawkEye AP, see the *Installation, Configuration, and Upgrade Guide*.

3 Stop HawkEye AP of existing deployment:

```
clsetup stop sensage
```

4 Copy the instance files from the correct hosts to the new host (leaving out dsroot); the example below uses the host names illustrated above:

```
clssh --in-dir=<Sensage_Home>/latest/etc/sls/instance/<instance_name>\
--hosts=Host_0"\
clsync --hosts=Host_5\
--exclude=<instance_name>/dsroot/Primary-<dir_version#>.d\
<Sensage_Home>/latest/etc/sls/instance/<instance_name>\
<Sensage_Home>/latest/etc/init_d/sensage_sls_<instance_name>"
```

5 Fix `dsroot` symbolic link in *<Sensage_Home>*/latest/etc/sls/instance/*<instance_name>* =directory

a Determine where dsroot points:

```
ls -l dsroot
```

Assume the above command returns:

```
/local/sensage/latest/data/sls/mysls/dsroot
```

b Create new directory for dsroot and for temp:

```
mkdir -p /local/sensage/latest/data/sls/mysls/dsroot
mkdir -p /local/sensage/latest/data/sls/mysls/temp
```

c Preserve the ownership and permissions that enable write access to the EDW user and group (typically `lms`); for example:

```
cd /local/sensage/latest/data/sls/mysls
chmod 770 dsroot
chown <sensage_user>:<sensage_user> dsroot
chmod <sensage_user>:root temp
chmod 770 temp
cd /local/sensage/latest/etc/sls/instance
chmod 700 <instance_name>
```

```
chown <sensage_user>:<sensage_user> <instance_name>
```

NOTE: You can determine appropriate settings by examining settings for corresponding directories on another host in your EDW instance.

6 Copy the instance files from the correct hosts to the new host (leaving out `dsroot`); the example uses the host names illustrated above:

```
clssh --in-dir=<HawkEye AP_Home>/latest/etc/sls/instance/<instance_name>\
--hosts=Host_0,Host_2,Host_3,Host_4,Host_5\
"<HawkEye AP_Home>/latest/bin/atperl -i orig -p -e ’s/down/active/g;s/Host_1/
Host_5/g’ cluster.xml"
```

7 Synchronize the primary and secondary data stores from the correct hosts to the new host; the example uses the host names illustrated above:

a Change to `dsroot` on the new host:

```
ssh Host_5
cd /<path_to_dsroot_directory>
```

b Run the following command to copy the primary data store to the new host as its secondary data store:

```
rsync -e ssh -rpogv
root@Host_0:/<path_to_dsroot_directory>/Primary-<dir_version#>.d/*
Secondary-<dir_version#>.d
```

c Run the following command to copy the secondary data store to the new host as its primary data store:

```
rsync -e ssh -rpogv
root@Host_2:/<path_to_dsroot_directory>/Secondary-<dir_version#>.d/*
Primary-<dir_version#>.d
```

d Run the following command to copy the `NODE.dat` file to the new host:

```
rsync -e ssh -rpogv
root@Host_2:/<path_to_dsroot_directory>/NODE.dat .
```

e Edit the `NODE.dat` file to correct, if necessary, the directory version number specified for the primary and secondary data stores. For example, if your primary directory is `Primary-12.d` and the `NODE.dat` file specifies `Primary-10.d`, correct the version number in the `NODE.dat` file.

8 If you have configured nearline storage, copy the following file from any other EDW node to the node you are restoring:

```
<Sensage_Home>/<path to data directory>/nss_idlistfile.dat
```

NOTE: The path to the data directory was configured during installation. Use that path to find the
`nss-idlistfile.dat` file.

9 Update the `cluster.xml` file on the old instance and the new host, replacing `down` with `active` and the name of the offline host with the name of the new host:

```
clssh --in-dir=<<Sensage_Home>_Home>/latest/etc/sls/instance/<instance_name> \
  --hosts=Host_0,Host_2,Host_3,Host_4,Host_5 \
  "atperl -i.orig -p -e 's/down/active/g;s/Host_1/Host_5/g' cluster.xml"
```

10If the Collector has changed the extension of a load file to `.noload`, change the extension to `.log` to restart loading.

11Restart HawkEye AP using the following command:

```
clsetup start sensage
```

SAMPLE CLUSTER.XML FILE FOR 3-NODE HAWKEYE AP

```
<clusters>
<cluster name="mysls" uuid="b31bd06ca6663e44">
<node urlspace="/cgi-app/xmlrpc" status="active"
name="edw01.mydomain:b31bd06ca666
3e44" port="8072" host="edw01.mydomain">
<role name="Master"/>
<role name="QP"/>
<role name="DSM"><sibling name="edw02.mydomain:b31bd06ca6663e44"/>
</role>
<role name="Node"/>
</node>
<node urlspace="/cgi-app/xmlrpc" status="active"
name="edw02.mydomain:b31bd06ca666
3e44" port="8072" host="edw02.mydomain">
<role name="Master"/>
<role name="QP"/>
<role name="DSM"><sibling name="edw03.mydomain:b31bd06ca6663e44"/>
</role>
<role name="Node"/>
</node>
<node urlspace="/cgi-app/xmlrpc" status="active"
name="edw03.mydomain:b31bd06ca666
3e44" port="8072" host="edw03.mydomain">
<role name="Master"/>
<role name="QP"/>
<role name="DSM"><sibling name="edw01.mydomain:b31bd06ca6663e44"/>
</role>
<role name="Node"/>
</node>
</cluster>
</clusters>
```

CHAPTER 6

Administering HawkEye AP Console and the Application Manager

This chapter discusses system administrator tasks for managing HawkEye AP Console and contains the following sections:

- "Accessing HawkEye AP Console", on page 183
- "Administering HawkEye AP Console and the Application Manager", on page 183
- "HawkEye AP Console Log Files", on page 197

ACCESSING HAWKEYE AP CONSOLE

This section describes the following topics:

- "Overview", next
- "Logging into HawkEye AP Console", on page 183
- "Determining the URL for HawkEye AP Console", on page 184

Overview

When you log into HawkEye AP Console, you are also logging into the Event Data Warehouse (EDW) instance that stores your event-log data. The modes, data, and objects that display in the console are determined by the roles to which the user has been assigned and the permissions they authorize. For more information, see "Role-Based Access to Functionality in the HawkEye AP Console", on page 224.

Logging into HawkEye AP Console

You can use either the HTTP or HTTPS protocol to open the HawkEye AP Welcome page, from which you can open HawkEye AP Console. HTTP sends communication in the clear; HTTPS uses Secure-Socket Layer (SSL) to send encrypted communication.

The web address for opening the welcome page in a browser depends on which protocol the Application Manager was configured to support, and whether a standard or non-standard port was assigned.

Generally, the web address for the welcome page has the following form:

```
http://<host_name>[:<port_number>]
```

—or —

```
https://<host_name>[:<port_number>]
```

For *<host>*, use the fully qualified name of the host where Application Manager is installed. Use the fully qualified domain of the host to avoid certificate mismatches when you start the console.

For `<port_number>`, use the port number specified during configuration of the Application Manager.

Determining the URL for HawkEye AP Console

If you do not remember the host or port, you can obtain this information by logging onto one of your HawkEye AP hosts. See Setting Up SSH Trust in Chapter 1, “Installing HawkEye AP” in the *Installation, Configuration, and Upgrade Guide*. System administrators should provide this URL to their users.

Administering HawkEye AP Console and the Application Manager

This section describes the following topics:

- “Logging into HawkEye AP Console”, next
- “Investigating Application Manager Problems and Events in Log Files”, on page 184
- “Starting, Stopping, and Restarting Application Manager”, on page 189
- “Creating Schedules to Run Reports and Dashboards”, on page 189
- “Managing Report Caches”, on page 190
- “Troubleshooting Installations of HawkEye AP Console”, on page 195

Configuring Logging for HawkEye AP Console

HawkEye AP provides an auditing mechanism that logs activities performed by the Application Manager. The Application Manager also logs error and transaction information. Logging is pre-configured and uses the Apache log4j logging services. If you need to change this configuration, for instance, to change the logging level, you should be familiar with log4j configuration. For more information, see:

- http://logging.apache.org/log4j/index.html
- http://en.wikipedia.org/wiki/Log4j

The log4j configuration file is at the following location in your HawkEye AP installation:

```
<Sensage_Home>/latest/jboss/server/default/conf/jboss-log4j.xml
```

After making changes to this file, restart the Application Manager using the following command:

```
  <Sensage_Home>/latest/etc/init.d/sensage_appserver restart
```

Investigating Application Manager Problems and Events in Log Files

The Application Manager writes log data to two different files:

- `app_manager.log`–contains all log information. This file is useful for investigating HawkEye AP Console problems and exceptions.

- `app_manager_audit.log`—contains only the subset of log data for auditable events, such as user logins and login failures, and report creation, modification, and deletion. This file provides an audit trail and enables reporting on this data. For more information, see "Investigating Audit Trails", next.

The Application Manager writes both files in the following directory:

```
/<Sensage_Home>/latest/var/log/controller/
```

The Application Manager also sends the `app_manager_audit` data to syslog-ng, which writes the data to the following directory:

```
/<Sensage_Home>/incoming/syslog-ng/sensage_applicationManagerAudit_syslogng/
```

Syslog-ng writes this audit data to a file named `sensage_applicationManagerAudit_syslogng.log`. The Collector uses a file system retriever to load the data from this audit log file into the EDW.

In the `/<Sensage_Home>/latest/var/log/controller/` directory, the Application Manager maintains a separate log file for each day of operation. When the first event of a new day is processed, (that is, the first event after midnight) or when the Application Manager is stopped and restarted, the latest log is rolled over, dated, and compressed.

- The compressed file created at midnight identifies the date represented by the file:

  ```
  app_manager.log.<yyyy-mm-dd>.gz
  ```

  ```
  app_manager_audit.log.<yyyy-mm-dd>.gz.
  ```

- The compressed file created by Console-Manager restart identifies the date, time, and GMT offset represented by the file:

  ```
  app_manager.log.<yyyy-mm-ddThr-mm-ss-GMToffset>.gz
  ```

  ```
  app_manager_audit.log.<yyyy-mm-ddThr-mm-ss-GMToffset>.gz.
  ```

Only the active logs (`app_manager.log` and `app_manager_audit.log`), the ones to which entries are currently being written, have no date suffix. The Application Manager keeps each dated log file for 168 hours before it deletes the file.

For example, the log directory could contain the following files:

- active logs:

  ```
  app_manager.log
  ```

  ```
  app_manager_audit.log
  ```

- logs rolled over at midnight:

  ```
  app_manager.log.2009-08-17.gz
  ```

  ```
  app_manager_audit.log.2009-08-17.gz
  ```

- logs rolled over when the Application Manager was restarted:

```
app_manager.log.2009-08-17T15:51:39-0700.gz

app_manager_audit.log.2009-08-17T18:25:22-0700.gz
```

Investigating Audit Trails

As mentioned above, the Application Manager maintains `app_manager_audit.log`. If you install the Foundation Analytics Report Package, you can run any of the Internal System Health and Status Module reports to view events contained in this log file. See Internal System Monitoring in Chapter 4, “Foundation Analytics Report Package” in the *Analytics Guide*.

Although you can open this file directly to examine the audit logs, the meaning of the information requires interpretation. Each audit entry consists of name/value pairs separated by tabs. The table below describes the value pairs included in every audit log.

Key	Description
LOG_LEVEL	This value is always "`AUDIT`"
APP_VERSION	Current version
SESSION_ID	ID of the user session
STATUS	"`ENTERED`", "`FAILED`", or `SUCCEEDED`"
DATE_TIME	The time of the request in GMT
TIMEZONE	The time zone of above date time (server time zone)
USER_ID	ID of the user who is logged in
SOURCE_IP	IP address of the request
THREAD	ID of the thread handling the request
SERVICE	The name of the service, which is one of the following: • Dashboard Service • Report Service • Journal Service • Schedule Service • Security Service • Session Service • User Service **NOTE**: Each service displays with the method called. The method is specific to the service type. for example: `SERVICE=DashboardService.getDashboardItems`

The table below describes the optional value pairs that depend on the activity and status of the log entry.

Key	Description
OBJECT_ID_1	ID of the object acted on, such as a report or dashboard. This field is used for read, update, delete, and run activities.
OBJECT_NAME_1	The name of the object; used for create activities where no ID has been assigned yet
OBJECT_TYPE_1	The type of object acted upon

Key	Description
OBJECT_ID_2	ID of a secondary object
OBJECT_NAME_2	The name of a secondary object
OBJECT_TYPE_2	The type of a secondary object
OBJECT_ID_3	ID of a secondary object
OBJECT_NAME_3	The name of a secondary object
OBJECT_TYPE_3	The type of a secondary object
OBJECT_ID_4	ID of a secondary object
OBJECT_NAME_4	The name of a secondary object
OBJECT_TYPE_4	The type of object acted upon
OBJECT_ID_5	ID of a secondary object
OBJECT_NAME_5	The name of a secondary object
OBJECT_TYPE_5	The type of a secondary object
PARAMS	A string consisting of name/value for each parameter, delimited by " `~|~` ". Parameters of type list are of the form listnameIndex=value; for example: `userrole_1=administrator  ~|~  userrole_2=guest` **NOTE**: This value contains all important parameters but not all information for all parameters.
ERROR_CODE	The error code upon failure
ERROR_TEXT	The error message in English of the error code

Example of Simple Log Entry

```
2009-12-14 13:27:20,113 AUDIT app_manager[http-0.0.0.0-31001-Processor73]
(com.sensage.middletier.audit.AuditEvent)     OBJECT_ID_1=/%089ea66451775e90
SESSION_ID=U72b1033096bdf9c897524ceef89849f6    STATUS=SUCCEEDED
OBJECT_NAME_1=McAfee NSP Top 20 Most Common Events THREAD=http-0.0.0.0-
31001-Processor73   USER_ID=administrator   TIME_ZONE=US/Pacific
OBJECT_TYPE_1=com.sensage.rpc.databean.report.ReportBean        DATE_TIME=2009-
12-14T21:27:20.113Z      SOURCE_IP=10.0.2.224
SERVICE=ReportService.createReportDefinition
```

The above log entry illustrates successful creation of a report definition named “`McAfee NSP Top 20 Most Common Events`“:

```
SERVICE=ReportService.createReportDefinition

OBJECT_NAME_1=McAfee NSP Top 20 Most Common Events

STATUS=SUCCEEDED
```

The report definition was created by the administrator user on March 30, 2009 at 6 PM:

```
USER_ID=administrator

DATE_TIME=2009-12-14T21:27:20.113Z
```

Example of Log Entry with a Parameter

```
2009-12-09 15:39:06,767 AUDIT app_manager[http-0.0.0.0-31001-Processor85]
(com.sensage.middletier.audit.AuditEvent)
SESSION_ID=Ua5da027a1558769bab76586d66c9fca0    STATUS=ENTERED  THREAD=http-
0.0.0.0-31001-Processor85   USER_ID=administrator   PARAMS=WHERE_CLAUSE=where
dashboarditem.name = '<root-dashboard-folder>' ~|~
objectType=com.sensage.rpc.databean.ParamBean ~|~ dirty=true ~|~
name={objectType=nullObject} ~|~ id={objectType=nullObject}       TIME_ZONE=US/
Pacific    DATE_TIME=2009-12-09T23:39:06.767Z      SOURCE_IP=10.0.2.177
SERVICE=DashboardService.getDashboardItems
```

The above log entry illustrates locating the root dashboard folder for a dashboard that has changed:

```
SERVICE=DashboardService.getDashboardItems

PARAMS=WHERE_CLAUSE=where dashboarditem.name = '<root-dashboard-folder>' ...
```

NOTE: The ellipses (...) indicate omitted text.

This log entry illustrates a request that has been logged rather than an operation that has completed:

```
STATUS=ENTERED
```

Example of Log Entries with a Login and Logout

```
2009-12-09 13:33:55,046 AUDIT app_manager[http-0.0.0.0-31001-Processor119]
(com.sensage.middletier.audit.AuditEvent)    OBJECT_ID_1=administrator
STATUS=ENTERED  THREAD=http-0.0.0.0-31001-Processor119  TIME_ZONE=US/Pacific
DATE_TIME=2009-12-14T21:33:55.614Z        SOURCE_IP=10.0.2.177
SERVICE=SessionService.authenticate

2009-12-14 15:38:53,614 AUDIT app_manager[http-0.0.0.0-31001-Processor127]
(com.sensage.middletier.audit.AuditEvent)
SESSION_ID=U72b1033096bdf9c897524ceef89849f6    STATUS=ENTERED  THREAD=http-
0.0.0.0-31001-Processor127  USER_ID=administrator   TIME_ZONE=US/Pacific
DATE_TIME=2009-12-14T23:38:53.046Z         SOURCE_IP=10.0.2.224
SERVICE=SessionService.logout
```

The above log entry illustrates locating login and logout activity. Both use the `SessionService` value of the `SERVICE` key. For logout, the method is `logout`. For login, the method is `authenticate` rather than `login`. Users are able to log in only after they pass authentication.

Schedule Failures

NOTE: When a schedule fails, the Application Manager logs a message to the `app_manager_audit.log` file (documented in "Investigating Audit Trails", on page 186). However this audit trail file does not include the reason for failure. To determine the cause of failure, search in the Application Manager's `error.log` or `activity.log` files documented in this section.

ANALYTICS REPORTS OF CONSOLE LOG FILES

The logged information tracks when each activity began and whether it was successful. The Analytics Foundation Package provides reports that query these logs for commonly needed information. For information about these reports, see *Chapter 4: Foundation Analytics Report Package* in the *Analytics Guide*. For information about creating a report, see Chapter 3: Running, Viewing, and Managing Reports in the *Reporting Guide*.

STARTING, STOPPING, AND RESTARTING APPLICATION MANAGER

You cannot perform any of the following three functions in HawkEye AP Console. You must login as root on the head node in your EDW instance.

IMPORTANT: If the HawkEye AP Application Manager is not running when a schedule runs, the scheduled item runs when the Application Manager is restarted. If this is not desired behavior, disable the schedules before stopping the Application Manager. See Chapter 7: Editing and Deleting Schedules in the *Reporting Guide*.

To start the Application Manager

As root, enter the following command.

```
<Sensage_Home>/latest/etc/init.d/sensage_appserver start
```

To stop the Application Manager

As root, enter the following command.

```
<Sensage_Home>/latest/etc/init.d/sensage_appserver stop
```

To restart the Application Manager

As root and, enter the following command.

```
<Sensage_Home>/latest/etc/init.d/sensage_appserver restart
```

CREATING SCHEDULES TO RUN REPORTS AND DASHBOARDS

Creating and managing schedules is an ongoing administration task. You can only perform this task in HawkEye AP Console. For more information, see Chapter 7: Creating and Editing Schedules in the *Reporting Guide*.

Setting the Maximum Rows Returned

You can set a maximum in the `controller.prop` file for the number of report rows that can be returned by any query, as follows:

1 Open `controller.prop` in a text editor:

<Sensage_Home>/etc/controller/controller.prop

2 Set the maximum number of rows to be returned at the `maxQueryResultsRows` variable. For example:

```
maxQueryResultsRows=10000
```

The default is 100,000 rows returned. Setting the value at 0 allows an unlimited number of rows to be returned.

Managing Report Caches

Sizing the Report Caches

The Application Manager manages report *caches* that are created when HawkEye AP runs a report against event data in the EDW. Each cache contains all the rows returned by each run of a report. The Application Manager can easily handle caches containing up to one million average-sized rows but larger report caches may experience performance degradation when sorting and filtering the cached data. You can optimize the performance of cached reports by creating and scheduling reports that return fewer rows. To return fewer rows, you may wish to change the frequency of the report run or change the report's selection criteria to limit the number of rows returned by the report.

Note the following guidelines when estimating the size of report caches:

- In general, caches where the size of the content is greater than 1/2 of the host machine's physical RAM will encounter severe performance issues or may fail to load.
- Data sets with many small fields will require more RAM than those having a small number of large fields.

Pruning Report Cache

The Application Manager saves "cached" versions of the output of every report that has been run. Users can delete these cache entries manually using HawkEye AP Console (see Viewing Report Results and Managing Report Cache Entries in Chapter 3, "Running, Viewing, and Managing Reports" in the *Reporting Guide*), or an administrator can specify a cache "pruning" task that runs at regular intervals and deletes reports based on their age. This section describes how to configure this cache pruning task.

To specify automatic report cache entry pruning

1 Open the following file in a text editor:

```
<Sensage_Home>/etc/controller/controller.prop
```

2 Edit the following properties, as described in the table below.

Property	Description	Default Value
`queryResultPruneDays`	Number of days to retain report cache entries. A value of 0 means that report caches are retained indefinitely.	0

Property	Description	Default Value
`queryResultPrunePruneFrequency`	Number of times per day to run the cache pruning task. Valid values are `0,1,2,3,4,6,8,12,24` A value of 0 means that the cache pruning task never runs.	0
`queryResultPrunePruneOffset`	Number of minutes after midnight when the first cache pruning task runs. A value of 0 means that the cache pruning task begins at midnight.	0

3 Save the file and exit the editor.

4 Restart the Application Manager using the following command:

```
<Sensage_Home>/latest/etc/init.d/sensage_appserver restart
```

For example, the following set of properties specifies that the cache pruning task runs twice per day and prunes only caches older than 1 year old. The first run of the task begins at 12:05 AM.

```
queryResultPruneDays=365
queryResultPruneFrequency=2
queryResultPruneOffset=5
```

Configuring the Encoding Used for Scheduled Reports Exported in CSV Format

To configure the encoding used for scheduled reports that specify CSV as their output format, set the `ReportEncoding` property in the `controller.prop` file. For example:

```
ReportEncoding=UTF8
```

Set this property to one of the basic or extended Java encodings shown in the two tables below. For a complete description of these encodings, see http://java.sun.com/j2se/1.5.0/docs/guide/intl/encoding.doc.html.

basic Java Encodings Supported for CSV Export

Encoding
ISO8859_1
ISO8859_2
ISO8859_4
ISO8859_5
ISO8859_7
ISO8859_9
ISO8859_13
ISO8859_15
KOI8_R
ASCII
UTF8
UTF-16
UnicodeBigUnmarked
UnicodeLittleUnmarked
Cp1250
Cp1251
Cp1252
Cp1253
Cp1254
Cp1257
UnicodeBig
UnicodeLittle

Extended Java Encodings Supported for CSV Export

Extended Encoding
Big5
Big5_HKSCS
EUC_JP
EUC_KR
GB18030
EUC_CN
GBK
Cp838
Cp858
Cp1140
Cp1141
Cp1142
Cp1143

Extended Encoding
Cp1144
Cp1145
Cp1146
Cp1147
Cp1148
Cp1149
Cp037
Cp1026
Cp1047
Cp273
Cp277
Cp278
Cp280
Cp284
Cp285
Cp297
Cp420
Cp424
Cp437
Cp500
Cp775
Cp850
Cp852
Cp855
Cp857
Cp860
Cp861
Cp862
Cp863
Cp864
Cp865
Cp866
Cp868
Cp869
Cp870
Cp871
Cp918
ISO2022CN
ISO2022JP
ISO2022KR

Extended Encoding
ISO8859_3
ISO8859_6
ISO8859_8
SJIS
TIS620
Cp1255
Cp1256
Cp1258
MS932
Big5_Solaris
EUC_JP_LINUX
EUC_TW
EUC_JP_Solaris
Cp1006
Cp1025
Cp1046
Cp1097
Cp1098
Cp1112
Cp1122
Cp1123
Cp1124
Cp1381
Cp1383
Cp33722
Cp737
Cp856
Cp874
Cp875
Cp921
Cp922
Cp930
Cp933
Cp935
Cp937
Cp939
Cp942
Cp942C
Cp943
Cp943C

Extended Encoding
Cp948
Cp949
Cp949C
Cp950
Cp964
Cp970
ISCII91
ISO2022_CN_CNS
ISO2022_CN_GB
x-iso-8859-11
JISAutoDetect
x-Johab
MacArabic
MacCentralEurope
MacCroatian
MacCyrillic
MacDingbat
MacGreek
MacHebrew
MacIceland
MacRoman
MacRomania
MacSymbol
MacThai
MacTurkish
MacUkraine
MS950_HKSCS
MS936
PCK
MS874
MS949
MS950

TROUBLESHOOTING INSTALLATIONS OF HAWKEYE AP CONSOLE

Perform these actions to help troubleshoot problems with installations of HawkEye AP Console:

- “Examine the Log Files”, next
- “Double-Check Basics”, on page 196
- “Clear the Java Web Start Cache”, on page 196

- "HawkEye AP Console Log Files", on page 197

Examine the Log Files

Monitor the following log files created by the Application Manager to look for unexpected behavior. For more information, see "Configuring Logging for HawkEye AP Console", on page 184.

Double-Check Basics

- If you are installing an upgrade, did you remove all old, temporary, staging directories?
- Did you install the correct upgrade packages? If you install each package using wild cards to replace the version names, you could be installing the wrong files and the installation can fail without error.

  To list all RPMs on your system, run:

  ```
  rpm -qa | grep ^lms-
  ```

  All of the HawkEye AP packages begin with "`lms`".

Clear the Java Web Start Cache

Strange behavior with HawkEye AP Console on a particular workstation can result from an outdated cache in Java Web Start. When you clear the cache, Java Web Start removes the JNLP file that was downloaded from the Application Manager, and it removes the desktop and start menu shortcuts. After you clear the cache, you must install HawkEye AP Console again.

To clear the cache in Java Web Start

1 Open the **Java Web Start** console:

On Windows workstations

a Select **Run** from the Start menu.

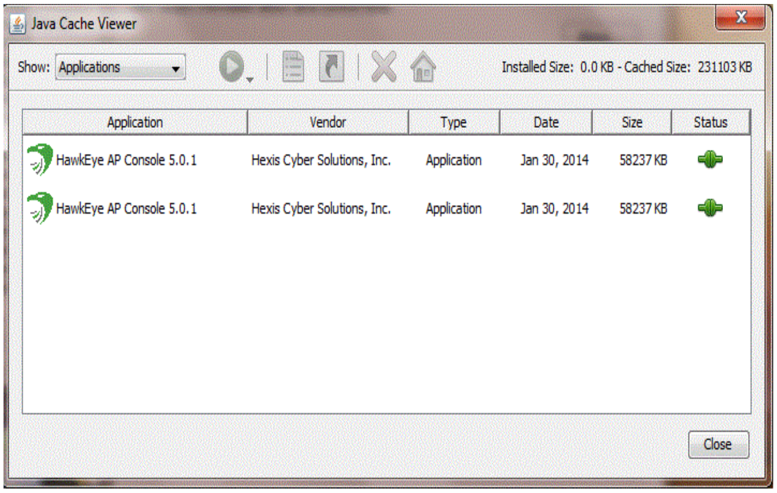
b In the Run dialog box, type "`javaws -viewer`" and click **OK**.

TIP: If the `javaws` executable is not in the PATH environment variable, run the following command to locate it:

```
  cd /usr/java ; find ./ -type f -name javaws -print
```

The Java Cache Viewer displays, as shown in Figure 6-1, below.

Figure 6-1: Java Web Start Cache Viewer



Java Cache Viewer

Show: Applications

Installed Size: 0.0 KB - Cached Size: 231103 KB

Application	Vendor	Type	Date	Size	Status
HawkEye AP Console 5.0.1	Hexis Cyber Solutions, Inc.	Application	Jan 30, 2014	58237 KB	✚
HawkEye AP Console 5.0.1	Hexis Cyber Solutions, Inc.	Application	Jan 30, 2014	58237 KB	✚

2 Select each application labeled **HawkEye AP Console 5.0.1** and delete it by clicking the red X button in the tool bar.

3 Click **Close**.

The Java Cache Viewer closes.

4 If the Java Control Panel window is also open, click OK in the Java Control Panel window.

The Java Control Panel closes.

5 Install HawkEye AP Console again.

For instructions, see HawkEye AP Console in Chapter 1, “Installing HawkEye AP” in the *Installation, Configuration, and Upgrade Guide*.

HawkEye AP Console Log Files

HawkEye AP Console maintains a log file, called `console.log`, on each workstation where it runs. The file contains error and other messages. On Windows XP systems, this file is located in the following directory:

```
C:\Program Files\SenSage\SenSage Console 4.5\console.log
```

CHAPTER 7

Administering the Collector

This chapter contains these sections:

PREPARING LOG DATA FOR LOADING

The Collector does not rotate logs. Therefore, you need to provide your own means for rotating logs on your log hosts in a way that is compatible with the Collector. Specifically:

- provide a log file that is not actively being written to
- ensure that the logs maintain unique names
- include an archive sub-directory for SFTP, FTP, and HTTP

The Collector package includes a script called `log-rotate.exe` (for Windows) or `log-rotate` (for Linux) that provides an example of how you can write a script to achieve this task. Run the script with no options to see its usage.

To use the script, you must specify:

- a sourceFile
- a targetDir
- whether you want hourly or daily naming

The script creates an `archive/` directory (under the targetDir), then compresses and renames the files.

The script also has an option that enables you to specify a hierarchy of directories for archived sub-directories. Because the Collector typically does not use this option, you should turn it off.

Starting, Stopping, and Restarting the Collector

Use the following command to start, stop and restart the Collector. The syntax is:

```
clsetup [<general_options>] { start | stop | restart } rt collector \
[--host=<host>]
```

IMPORTANT:

- The `stop` argument forces shutdown of the Collector. To cause HawkEye AP to attempt to complete current operations before shutting down the Collector, run the following command:

  ```
  <Sensage_Home>/latest/bin/collectd graceful
  ```

  NOTE: You must run this command from the host on which the Collector is running.

- The `restart` argument causes HawkEye AP to attempt to complete current operations before shutting down the Collector and then restarts the Collector.
- If you configured the Collector to run on more than one host, you are running more than one Collector. To operate on a specific Collector, specify its host in the `--host` flag. To operate on all Collectors, omit the `--host` flag. For more information, see Configuring the Collector Component: Overview in Chapter 2, “Configuring HawkEye AP” in the *Installation, Configuration, and Upgrade Guide*.
- Although the Collector is a component and is not a module of the Real-Time component, you must include “`rt`” in the `clsetup` command to control the Collector.

Monitoring Logs

This section describes these topic:

- “About Collector logs”, next
- “Setting the Logging Level”, on page 202
- “Tailing Logs Files”, on page 202
- “Things to watch out for”, on page 202

About Collector logs

You can configure the Collector to log to three log files:

- **transaction.log**—Only the start and end of loads and the retrieval of loads are recorded here. This is a pipe-delineated file used by PTLs.
- **activity.log**—All Collector activity logs to this file.
- **error.log**—Error messages generated from `activity.log` appear in this file. The mapping of error codes to messages is stored in the following file:

  ```
  <Sensage_Home>/latest/share/locale/<locale>/collector/errors.xml
  ```

  The above file is loaded at startup and used whenever an error code is specified. You can also specify how Collector alerts users of errors in this file (using email or pagers, for example).

To specify whether the Collector logs to files, and to specify the logging levels and file location, edit the logging section of the `config.xml` file for editing. This file is located in:

```
<Sensage_Home>/latest/etc/collector/
```

The logging section is:

```
<!-- Log level 3 is most verbose -->
<Logging level="1" type="syslog"/>

<!-- Uncomment below for file logging -->
<!--
<Logging level="1" type="file" file="/opt/sensage/latest/var/log/collector/
error.log"/>
<Logging level="2" type="file" file="/opt/sensage/latest/var/log/collector/
activity.log"/>
<Transaction>/opt/sensage/latest/var/log/collector/transaction.log</Transaction>
-->
```

Logging to syslog

To log to syslog, the only change you should make is to the log level. Then restart the Collector.

To set File-Based Logging

To use file-based logging, comment out the syslog line and uncomment the remaining lines. If you are not using the default HawkEye AP home directory, you must also modify the path.

Setting Both syslog and File-Base Logging

Make both changes described above. In other words, set the logging level for syslog and uncomment the lines for file-based logging. Modify the path if necessary and restart the Collector. For more information, see “Starting, Stopping, and Restarting the Collector”, on page 200.

Log Formats for Transaction Logs

The following fields appear in `transaction.log`:

```
timestamp
type
name
fileSource
startTimestamp
endTimestamp
bytesCopied
errorCode
errorMsg
minTimestamp
maxTimestamp
uploadid
recordsLoaded
matchFailures
atloadCommand
PTLPath
atloadVersion
atloadExitStatus
```

Log Formats for Activity and Error logs

The following fields appear in `activity.log`, `error.log`, and associated alerts:

```
Timestamp (in ISO time format: YYYY-MM-DDTHH:mm:ss)
Process ID
Log level of message (1 for major activity, 2 for notices, 3 for verbose)
Error code, if any (defaults to 0 for no error)
Class Name
Line number
Message
```

Setting the Logging Level

Set the logging level in `config.xml` with the `level` attribute in the `<Logging>` element. You can specify multiple `<Logging>` tags, which allows you to specify different logging parameters for each level. For example, the following configuration specifies that *all logs* (level 3) are logged to a file while *errors* (level 2) are logged to syslog:

```
<Logging level=”2”> type=”syslog”/>
<Logging level="3"> type=”file” file=”tmp/activity.log”/>
```

Specify one of these logging levels.

Level	Meaning
1	Least verbose. Only critical errors or severe warnings are logged.
2	Moderately verbose. Event history is logged, as well as critical errors and sever warnings. Set logging to level 2 if you intend to tail log files. For more information, see “Tailing Logs Files”, next.
3	Most verbose. Hexis Cyber Solutions recommends that you set logging to level 3 during installation, initial configuration, and troubleshooting

Tailing Logs Files

You can view the log files using any text editor, or you can view the most recent entries using the `tail` command.

To view data logged to a file

```
tail -f <Sensage_Home>/latest/var/log/collector/<logfile>
```

where *`<logfile>`* is `activity.log`, `error.log`, or `transaction.log`.

To view data logged to syslog

```
tail -f /var/log/messages
```

Things to watch out for

As you monitor logs, watch out for the following:

- Failed Retrievers
- Failed Loaders
- Failed Loads
- Parse Failures
- Corrupted or Missing Configuration File

Failed Retrievers

A Retriever may go down if any of its processes fail (for example, `BackupDir` for backing up raw log data). In most cases, you need to restart the Collector after fixing the problem. When a retriever is configured to use a preprocessor and the preprocessor fails to generate an output file, the Collector will rename the `.in` files to `.noload` in the spool directory.

To determine if this is the problem, run the preprocessor (with `.in` and `.out` files) outside of the Collector. If you do not see your postprocessed data in the `.out` file, fix the preprocessor script (rename the `.noload` files to `.raw`). Then fix the preprocessor in the Collector (rename the `.noload` files to `.raw` in the spool directory).

Failed Loaders

Restart the Collector if a loader goes down (time out if `atload` fails). Some other issues you may see include

- Zero-byte log files.
- Incorrect permissions on directories and files—generates critical error.
- Low or no free space on disk—causes Log Queues to become disabled (restart to fix).
- Unmounted file systems—do not use NFS; it will throw error on Retrievers.
- Files being unlinked during processing—might not throw error if no metafile gets created. Clean out the queues and start over. Delete plugin files. Remove `MD5` lines for those files (still could get duplicate loads).

Failed Loads

On failed loads, you need to get the unloaded files back into the system. Loads can fail in two locations:

- In the main queue directory; here, rename log files with the `.noload` extension to `.log` and restart the Collector.
- In the spool directory (means that preprocessing failed); rename the `.noload files` to `.raw`, then rerun the preprocessor.

To trace back to the failure, set the logging level to `3` and investigate `activity.log`.

For more information, see “Setting the Logging Level”, on page 202 and “Handling Unsuccessful Loads”, on page 204.

Parse Failures

Parse failures mean that there is a problem in the PTL file. These failures may also occur if the log format changes, or if log files become corrupt.

Also see the ParseFailure setting described in Defining Loaders in Chapter 3, “Collector Configuration” in the *Event Collection Guide.* You can have multiple Loaders mapped to the same parse failure location.

Corrupted or Missing Configuration File

The `clsetup configure sensage` command creates a backup version of the configuration file in `<*Sensage_Home*>/latest/etc/collector/config.xml.*N*` where *`N`* reflects the number of times you have reconfigured the Collector. Each time you reconfigure the Collector, the system creates a new `config.xml.1` file and increments the numbered extensions of the existing `config.xml.*N*` files. If `config.xml` is deleted or corrupted, rename *`config.xml.1`* to *`config.xml`*.

HANDLING UNSUCCESSFUL LOADS

This section describes these topics:

- “About Unsuccessful Loads”, next
- “Viewing Data from Unsuccessful Loads”, on page 204
- “Tracking Uploads in the EDW”, on page 204

About Unsuccessful Loads

Log data does not always load correctly into the EDW instance. Incorrectly loaded data can include loads that partially completed or loads that were aborted and reloaded. The EDW tracks every upload to the system, storing the unique identifier of each load in the `_uploadid` column of each table. Loads that fail upload are tagged as inconsistent. By default, the data in an inconsistent load does not display when you query the EDW.

NOTE: A successful load does not necessarily insert every row from the source log file. Typically log files contain data that cannot be parsed and loaded; however, bad data does not prevent these files from loading successfully. A failed load is caused by a protocol error, such as not receiving all the expected data or a load cancelation.

Viewing Data from Unsuccessful Loads

Each row in EDW tables tracks the ID of the upload operation inserted the row. Rows inserted by an unsuccessful load are excluded from queries by default.

If you want to view all data in a table, including data from unsuccessful loads or data currently being loaded, you must specify the `{INCLUDE_BAD_UPLOADS}` modifier on the table in the `FROM` clause. Include the curly braces as part of the modifier.

NOTE: If a query and a load occur simultaneously and the query continues after the load completes, you may see all, none, or part of the data from the new load.

Tracking Uploads in the EDW

In addition to loading log data into the specified tables, `atload` saves information in the EDW about the load. You can view this load information by querying these system tables:

- `system.upload_info`

This table enables administrators to check on upload status in the system. When the system is healthy, this table returns one row for each upload. When the upload data is not consistent across all the nodes in the cluster, this table returns a value of `false` in the CONSISTENT column. In this case, the table may return multiple rows with the same value in the UPLOADID column; all of these rows are marked as inconsistent.

- `system.raw_upload_info`

  This table enables a support person to see the raw data that is distributed across the system in order to troubleshoot an inconsistency problem revealed in the `system.upload_info` data.

Both of these system tables contain a unique identifier for the upload: the value of the `_uploadid` column. These tables store additional information for each load operation, which includes: the minimum and maximum timestamps, the number of lines in the source log data, the number of lines successfully parsed by the PTL file, the number of rows loaded into the specified SLS table, the PTL file and client signatures, and whether the load was successful.

Additionally, the `system.upload_info` table returns the consistency data and the `system.raw_upload_info` returns the logical name of the source node for the log data.

As it begins loading data into the log tables, the EDW inserts a new row in these system tables to track the upload. Initially, the value of the SUCCESSFUL column defaults to `false`. When the load completes successfully, this column's value changes to true.

Hexis Cyber Solutions recommends that you regularly query the `system.upload_info` table for loads with inconsistent data (the value of the CONSISTENT column is `false`). Inconsistent data was not correctly replicated across the cluster. Whenever you discover an inconsistent load, query the `system.raw_upload_info` table and capture the results in a file. Contact Hexis Cyber Solutions Technical Support.

Running Multiple Collectors

If a situation requires that you set up multiple Collectors (on different hosts), they should be configured to each manage a subset of log queues, and the subsets should not overlap. You can achieve this setup by using a daisy-chain configuration. For more information, see Creating Daisy Chains in Chapter 3, “Collector Configuration” in the *Event Collection Guide*.

Scheduling Retrievers and Loaders

When configuring Retrievers and Loaders, you define when retrieving and loading occurs using one of two methods:

- The PollInterval and Period Elements—Use the `<PollInterval>...</PollInterval>` element to specify *how often* loading and retrieving occurs. For example, you can specify that loads happen every hour. This is the most common method of specifying when loading and retrieving occur. You can combine the `<PollInterval>` element with the optional `<Period>...</Period>` element to specify when loading or retrieval begins.

  Use this method when configuring Merge loaders and when configuring Retrievers that log into other systems to see if there are files to be retrieved. These Retrievers include:

  - FTP Retrievers

  - SFTP Retrievers
  - HTTP Get Retrievers
  - LEA Retrievers
  - Script Retrievers.

- The Schedule Element—Use the `<Schedule> ... </Schedule>` element to specify a *specific time* for Retrievers and Loaders to run. For example, you can specify that loading occurs at 8:00 and 14:00 every day.

IMPORTANT: Do not combine the above methods. Use only the `<PollInterval>` with the optional `<Period>` element or the `<Schedule>` element.

For additional information, see Defining Retrievers and Defining Loaders in Chapter 3: Collector Configuration in the *Event Collection Guide.*

The Schedule Element

The <`Schedule>` element has the following syntax:

```
<Schedule>minute hour day_of_month month day_of_week</Schedule>
```

Use the `<Schedule>` element to define an event that should occur at a specific time. At the end of every event, the Collector tests the event conditions again to determine whether the event should run again. If a given event runs over the time when the next event of the same type would be scheduled, the second (and subsequent) missed events are not queued, they are ignored.

Fields in the Values of Schedule Elements

The value of a `<Schedule>` element has these positional fields.

Positional Field	Allowed Numeric Values
minute	0-59
hour	0-23
day_of_month	1-31
month	1-12
day_of_week	0-7 (0 and 7 represent Sunday)

Separate the fields with spaces. You must specify a value for each field. If you are unsure what specify for a positional field, specify an asterisk (`*`) but *do not specify an asterisk for all five fields*. If you are unsure of what to specify for all fields, do not specify the `<Schedule`> element and HawkEye AP will use the default behavior of once every minute.

Usage Notes

- A field can contain an asterisk (*), which functions identically as listing every possible value for that field. For example, an asterisk in the *day_of_month* field specifies events occur every day.

- Ranges of numbers are allowed. Ranges are two numbers separated by a hyphen. The specified range is inclusive. For example, `8-11` for in the *`hour`* field specifies execution at hours 8, 9, 10 and 11.
- Lists are allowed. A list is a set of numbers or number ranges separated by commas. For example, `1,2,5,9,0-4,8-12`.
- Names for the *`month`* and *`day_of_week`* fields are allowed. Use the first three letters of the particular day or month. Case does not matter. Number ranges and lists of names are not allowed.
- The day of a command's execution can be specified by two fields: *`day_of_month`* and *`day_of_week`*. If both fields have a schedule specification, the command runs when the current time matches the specification in either field matches. For example, the following schedule specification causes a command to be run at 4:30 AM on the 1st and 15th of each month, plus every Friday.

  ```
  <Schedule>30 4 1,15 * 5</Schedule>
  ```

- Do not specify an asterisk for all five fields. If you are unsure of what to specify for all fields, do not specify the `<Schedule>` element and HawkEye AP will use the default behavior of once every minute.

Examples Schedule Specifications

To run once an hour, on the first minute of the hour. (i.e. 12:00):

```
<Schedule>0 * * * *</Schedule>
```

To run every five minutes of every hour of every day:

```
<Schedule>0,5,10,15,20,25,30,35,40,45,50,55 * * * *</Schedule>
```

To run every second hour, starting at midnight:

```
<Schedule>0 0,2,4,6,8,10,12,14,16,18,20,22 * * *</Schedule>
```

To run once an hour, Monday through Friday.

```
<Schedule>0 * * * 1-5</Schedule>
```

The PollInterval and Period Elements

The `<PollInterval>` and `<Period>` elements have the following syntax:

```
<PollInterval>Interval</PollInterval>
<Period>Minute Hour DayOfMonth Month DayOfWeek</Period>
```

The `<PollInterval>` Interval defines how often, in seconds, the event in question will be attempted during the defined `<Period>`. Poll intervals of less than 60 seconds are not recommended. If the event runs longer than the poll interval or runs into the next period, the event is allowed to complete. (You can also specify `<PollInterval>` by adding `hour, hours`, `minute`, `minutes`, `second`, or `seconds`. For example, the following are all equivalent:`<PollInterval>1 hour</PollInterval>`, `<PollInterval>3600</PollInterval>, <PollInterval>3600 Seconds</PollInterval>.)`

The syntax for values in the `<Period>` element is the same as for the `<Schedule>` element. For a complete explanation of the syntax, see "The Schedule Element", on page 206.

EXAMPLES POLL INTERVAL AND PERIOD SPECIFICATION

To specify that an event runs every 5 minutes (300 seconds) from 1:00 AM through 5:59 AM:

```
<PollInterval>300</PollInterval>
<Period>* 1-5 * * *</Period>
```

The specification above is equivalent to the following `<Schedule>` specification:

```
<Schedule>0,5,10,15,20,25,30,35,40,45,50,55 1-5 * * *</Schedule>
```

To specify that an event runs every hour continuously, starting at the half-hour:

```
<PollInterval>3600</PollInterval>
<Period>30 * * * *</Period>
```

The specification above is equivalent to the following `<Schedule>` specification:

```
<Schedule>30 * * * * *</Schedule>
```

Scheduling and the Unchanged For Attribute

Regardless of which element you use to set the polling frequency, you can also set how long source log files should remain unchanged before the retriever transfers them for processing. In other words, you separately specify how frequently the Collector polls for events and how long the Retriever keeps an event log file before it releases it for processing.

To set the length of time the Retriever keeps an event log file before processing, you set the `<UnchangedFor>` attribute of the `<Plugin>` element. For more information about this element and attribute, see Plugin in Chapter 3, "Collector Configuration" of the *Event Collection Guide*.

The Collector determines when to take a file for processing based upon:

- Polling cycle
  - The frequency set either for the `<Schedule>` element or the `<PollInterval>` and `<Period>` elements

    or

- If no values have been set for `<Schedule>` or `<PollInterval>` and `<Period>`, the frequency specified for `<CycleDelay>` (See CycleDelay in Chapter 3, "Collector Configuration" of the *Event Collection Guide)*
  -

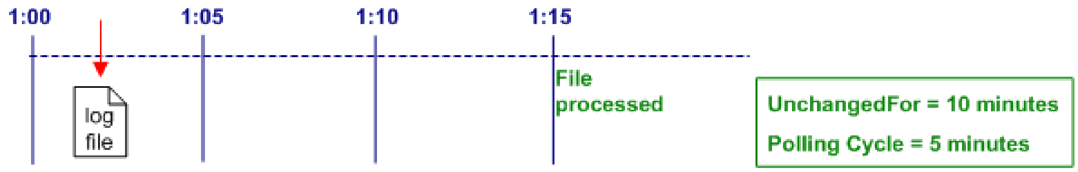
  and

- The value specified for the `<UnchangedFor>` attribute of the `<Plugin>` element

  Each time it polls, the Collector uses the `<UnchangedFor>` setting to determine whether a file has remained unchanged long enough to be ready for processing.

Figure 7-1 illustrates how the Collector processes a file when the `<UnchangedFor>` attribute has been set to 10 minutes and the polling cycle has been set to 5 minutes.

Figure 7-1 : Illustrating the Relationship of Schedule Frequency & UnchangedFor Attribute



As illustrated above, a log file arrives at 1:02, which is 2 minutes after the last Collector poll. At 1:05 and again at 1:10, the Collector checks whether the log file has changed. Because the file has not changed and 10 minutes has not elapsed since the file was written, the Collector ignores the file at these times. However, at the 1:15 poll, because 10 minutes *has* elapsed since the file was written, the Collector takes the file for processing.

NOTE: The value of `<CycleDelay>` in `config.xml` is irrelevant if you set either the `<Schedule>` element or the `<PollInterval>` and `<Period>` elements.

BACKING UP AND RESTORING SYSTEM AND LOG DATA

Files needed to back up for restoring the system are:

- `<`*`Sensage_Home`*`>/latest/etc/collector/config.xml`
- all MD5 data in the state directory, as specified by the `<StateRoot>` element in `config.xml`. For more information, see Root Directories Used by the Collector in Chapter 3, “Collector Configuration” in the *Event Collection Guide*.

BACKING UP RAW LOG DATA

You can configure retrievers to make backups of raw, downloaded logs before beginning preprocessing. Add a `<BackupDir>` element to the retriever configuration in `<`*`Sensage_Home`*`>/latest/etc/collector/config.xml`, as the next example shows:

```
<Retriever name="filesystem" type="filesystem" enabled="1" process="files1"
    method="hardlink" deleteOriginal="1" >
    ... other elements ...
    <BackupDir>file:///nfs1/backups</BackupDir>
    <!-- Copies file with scp to example.com using fred's account
       Note: this requires public/private keypairs.  -->
    <BackupDir>scp://fred@example.com/nfs2/backups</BackupDir>
    <!--  Copies file with scp using -C for stream compression. -->
    <BackupDir>scpc://fred@example.com/nfs2/backups</BackupDir>
</Retriever>
```

Log files are backed up after completing an atomic file write; that is, file system copy or ftp copy. You can also use scp, scpc, or an arbitrary user-specified command to do log backups.

IMPORTANT: :All commands are executed in parallel. If any command fails, the entire backup fails, and the retriever shuts down to prevent buildup of files that have not been backed up.

Description of Log File Names

Source log files are renamed to have a zero-padded, 3-digit priority number at the beginning of the filename. Non-prioritized files are marked with `000`. To prioritize a log, change the number to 1 higher than the highest number in the log queue. Loaders will look for the file with the highest number when considering which file to load.

Log files also have an ISO timestamp added to their name showing the time at which they were gathered, a sequence ID in the event that two logs are gathered during the same timestamp, and a string identifying the retriever that retrieved them.

The configuration file `config.xml` defines the log-queue directories where log files are stored. For more information, see Configuring Log Queues in Chapter 3, “Collector Configuration” in the *Event Collection Guide*.

Syntax for Log File Names

The names of event-log files that are processed by the Collector have this syntax:

```
PPP-YYYYMMDDtHHMMSS-SEQ-RETR.EXT
```

Filename Field	Meaning
PPP	Three-digit priority
YYYMMDDtHHMMSS	Date and time
SEQ	Numeric sequence number, normally `0` unless more than one file is generated per second
RETR	Retriever class name that generated this file
EXT	File extension

Extensions for Log File Names

The names of event-log files that are processed by the Collector these extensions, which indicate the processing states of log files.

Filename Extension	Meaning
`.out`	An output file in the process of transfer or preprocessing; All `.out` files with modification dates over one day old are deleted automatically.
`.in`	An intermediate file in an ongoing preprocess stage; All `.in` files with modification dates over one day old are deleted automatically.
`.pmd`	A file with no MD5 yet computed. At same time, a metadata (`.meta`) file is created.
`.meta`	An XML file containing meta data regarding a log file with the same name.
`.org`	A file needing backup before it can be renamed to `.raw`. Occurs if MD5 doesn't match.
`.noload`	A file whose preprocessor or loader has crashed. Rename to `.raw` to attempt preprocessing and loading again.

Filename Extension	Meaning
`.raw`	A file needing preprocessing. A file may only be preprocessed by the retriever name which created it
`.log`	A file ready for a loader to process.
`.wrk`	For daisy-chain configurations, a file created in the process of converting to a `.tar` file.

EXAMPLE LOG FILE NAME

```
000-20040101t010101-2-ftp1.log
```

REPRIORITIZING LOG FILES

You can prioritize log files within `config.xml`. Assign the `id` attribute to an integer, where “1” processes first, “2” next, and so on. The `id` attribute appears in the `<PTL>`, `<Preprocessor>`, and `<Plugin>` elements.

Prioritizing Log Files Manually

You can prioritize the order of log files in log queues manually. Edit the numerical prefix in a filename to place the log file at the appropriate spot in the log queue. Loaders search for the log file with the highest priority when selecting the next file to load.

NOTE: Hexis Cyber Solutions recommends that you prioritize log files with the `id` attribute in the `config.xml` file. Only use the manual procedure described below for troubleshooting or similar purposes.

To prioritize a log file by changing the priority field in the filename:

1 Create a hard link to the metadata file (found in `<LogQueue>/spool`).

```
# ls 000-20040315T101023-000-cpixrt.ext 000-20040315T101023-000-cpixrt.meta
# ln 000-20040315T101023-000-cpixrt.meta 001-20040315T101023-000-cpixrt.meta
```

2 Rename the log file, incrementing the priority prefix as you rename it. For example:

```
# mv 000-20040315T101023-000-cpixrt.log 001-20040315T101023-000-cpixrt.log
```

3 Remove the original metadata file.

```
# rm 000-20040315T101023-000-cpixrt.meta
```

About metadata files

Each gathered log file is paired with an XML-based metadata file of the same name. The metadata file contains information about the source of the log data and its current status. The metadata file is created when the log file is fully transferred into the Collector (that is, when the `.pmd` file is created). The extension of a log file changes as it moves through the system, but its corresponding metadata file keep its `.meta` extension throughout.

For more information about log file names and extensions, see “Description of Log File Names”, on page 210.

CLEANING UP PROCESSED LOG FILES

Once a log file has been processed, it is put into the `<LogQueue>/done` subdirectory. The system administrator must archive or remove these files from the system. HawkEye AP software includes a cleaner script with the Collector as an example of the type of script you should write to schedule the automatic deletion of processed log files. Run the script or program from the command line, and specify the path to your LogRoot as an option.

On Linux, run this script:

```
<Sensage_Home>/latest/bin/collector-extras/cleaner.pl
```

To learn more about LogRoot, see Root Directories Used by the Collector in Chapter 3, “Collector Configuration” in the *Event Collection Guide*.

CHAPTER 8

Administering Users and Authentication

This chapter contains the following sections:

- "Authentication", next
- "Users, Roles, and Permissions", on page 218
- "Creating and Managing Distribution Filters", on page 242

AUTHENTICATION

This section describes the following subsections:

- "Overview", next
- "HawkEye AP Authentication Model", on page 214
- "Deploying the Authenticator", on page 215
- "Integrating with Active Directory", on page 216
- "Using SSL Encryption with Authentication", on page 216

Overview

Authentication and authorization are security processes that identify users and determine the functions they can perform. HawkEye AP authentication and authorization includes two main verification checks, listed below.

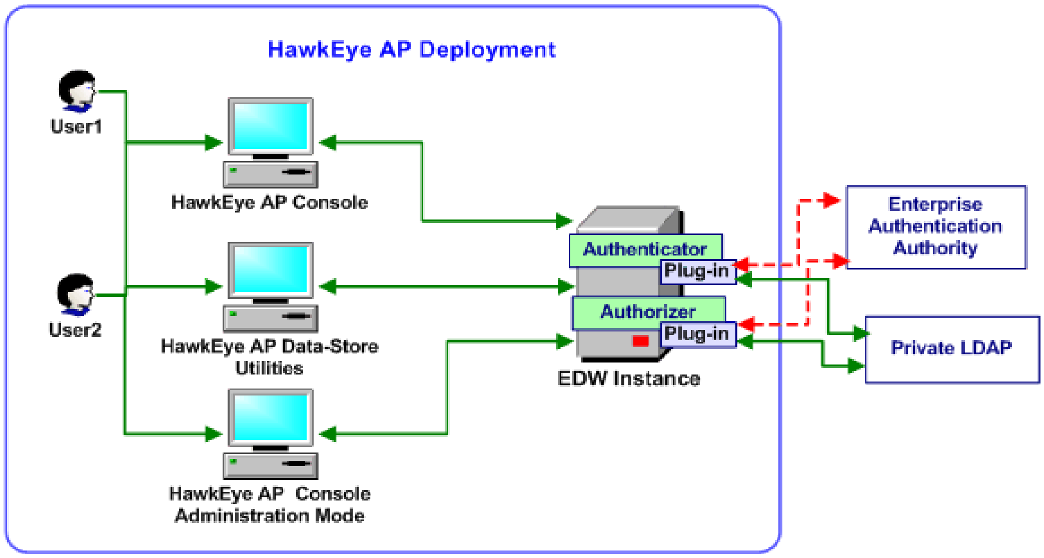
- **Authentication**—Identifies and validates users through their user name and password.
- **Authorization**—Locates and verifies the user's roles and permissions from the authentication authority.

For information on managing users, roles and permissions, see the "Users, Roles, and Permissions", on page 218. For information on permissions and objects, see ""Managing Access to HawkEye AP Console Reports, Dashboards, and Folders", on page 237.

HawkEye AP Authentication Model

A typical HawkEye AP authentication model consists of a HawkEye AP deployment integrated into the private LDAP (Lightweight Directory Access Protocol) included with HawkEye AP and an external enterprise authentication authority, as illustrated below.

Figure 8-1: HawkEye AP Authentication Mode



The authentication model above illustrates:

1 Two users log into the same EDW instance through HawkEye AP Console, HawkEye AP data-store utilities (such as `atmanage` or `atload`), or the HawkEye AP Administrator Console.

2 Each of the three HawkEye AP user-interface components sends the users' names and passwords to the EDW.

3 The EDW uses the following to validate and manage HawkEye AP users:

- **Authenticator**—verifies passwords

  The Authenticator gets password information by binding either to an external authentication authority such as Active Directory (broken red line in Figure 8-1) or to the HawkEye AP-private LDAP (solid green line). The Authenticator plug-in binds the EDW instance to the specified authentication authority and is specific to the authority to which it is bound. The authentication authority returns the authentication information to the EDW.

- **Authorizer**—gets permission and role information

  The Authorizer gets permission information from the HawkEye AP-private LDAP. It gets role information by binding either to an external authentication authority (such as Active Directory) or to the HawkEye AP-private LDAP. The broken red line in Figure 8-1 illustrates binding to the external authentication authority, and the solid green line illustrates binding to the HawkEye AP-private LDAP.

4 After the Authenticator validates the user and the Authorizer returns the user's roles and permissions, HawkEye AP allows the user to perform only those tasks and access only the data allowed by the user's roles and permissions.

- The result of the authentication determines which of the HawkEye AP Console components display to the user. For example, if two users with different permissions log into HawkEye AP Console, one user may have access to Administration mode while another can only access Dashboards mode. Additionally, the authentication determines which of a displayed component's objects are available to the user; for example, two users may have access to Dashboards mode but one may have access to many more reports than the other and one user may see alert widgets while the other does not. For more information, see the “Managing Access to EDW and HawkEye AP Console Objects”, on page 220.

- Authentication for report access and usage requires an additional security check. The Application Manager compares the user's roles and permissions to an additional layer of access control for each report object. For more information, see “Users, Roles, and Permissions”, on page 218.

Deploying the Authenticator

The HawkEye AP Authenticator functions in one of two ways, depending on whether the enterprise uses the HawkEye AP-provided LDAP or its own authorization authority:

- **Standalone**—The Authenticator uses the HawkEye AP-provided private LDAP directory to authorize and authenticate users.

- **Integrated**—The Authenticator uses an external authentication and authorization system, such as Active Directory, to integrate the HawkEye AP installation into an enterprise's authorization authority.

NOTE: Because the Authenticator is a distributed application, it is not associated with a specific piece of hardware.

Standalone Deployment

Standalone deployment refers to an EDW instance connected to the HawkEye AP-private LDAP authentication authority. For information on configuring an EDW instance to use LDAP authentication, see EDW Configuration in Chapter 2, “Configuring HawkEye AP” in the *Installation, Configuration, and Upgrade Guide*.

Integrated Deployment

When an enterprise has its own authentication system, such as Active Directory, you can configure an EDW instance to integrate with the authentication system. Integrating an EDW instance to an enterprise's authentication authority centralizes user name and password management.

NOTE: An EDW instance is always integrated to one, and only one, authentication authority.

Integrating with Active Directory

Active Directory is a Windows-based authentication authority that manages users and user groups. The Active Directory authority authenticates each user's name and password at login. After you configure Active Directory integration, the EDW logs into the Active Directory server for enumeration of groups and users and to perform Kerberos authentication on the user. The EDW uses an “unprivileged user” connection to read group memberships from the Active Directory to authenticate the user.

For information on configuring Active Directory integration, see Setting up Active Directory Integration in Chapter 6, “Configuring Active Directory Integration” in the *Installation, Configuration, and Upgrade Guide*.

NOTE:

- You must install HawkEye AP to use the HawkEye AP-private LDAP, and then reconfigure it to use Active Directory.
- Installing Microsoft Windows Server 2003 Enterprise Edition operating system on a server automatically installs Active Directory on the server. The HawkEye AP Active Directory integration allows your EDW instance to use the Microsoft Active Directory authentication authority that is used by an enterprise server.

HawkEye AP uses Kerberos, a strong authentication protocol based on key cryptography, to process authentication data between the EDW and Active Directory. Kerberos authenticates users on a host within a domain that is part of the Kerberos realm.

By default, each HawkEye AP SLS Authenticator uses a plug-in that integrates into the HawkEye AP-provided LDAP authentication authority. To integrate an EDW instance into an Active Directory authentication authority, you must specify the required values when you add or configure your EDW instance.

IMPORTANT: To integrate the EDW with Active Directory, you should be knowledgeable about the following:

- Microsoft® Active Directory®
- Microsoft® Windows Server™ 2003 Enterprise Edition operating system
- Microsoft® Windows Kerberos Version 5 Authentication Protocol.
- Lightweight Directory Access Protocol (LDAP)
- HawkEye AP authorization and authentication

Using SSL Encryption with Authentication

By default, Secure Sockets Layer (SSL)/Transport Layer Security (TLS) is enabled for communication between the LDAP server and all EDW hosts. You must use command-line utilities to modify default settings.

Configuration Options

You can control the use of SSL/TLS security protocols through the following `"clsetup configure sls"` options:

`--ldap-organization=[domain_name]`

If your LDAP domain name isn't recognized by the certificate engine, rename it with this option. If it is not specified, `--ldap-organization` is extracted from the domain name.

`--ldap-use-tls={no|yes>}`

Set to `"no"` to disable Transport Layer Security (TLS). The default is `"yes"`.

`--ldap-regenerate-cert={no|yes>}`

By default, the EDW does not regenerate an SSL certificate if one already exists in:

```
<SenSage_Home>/latest/etc/atslapd/atslapd.crt
```

To regenerate the SSL certificate, specify `"yes"`.

Configuration File Reference

These configuration files contain lines relevant to encryption:

- `slapd.conf`—configures the private LDAP server provided by HawkEye AP. It is located in:

  ```
  <Sensage_Home>/latest/etc/atslapd
  ```

  Relevant lines in the `slapd.conf` file are:

  ```
  TLSCipherSuite HIGH
  TLSCertificateFile <SenSage_Home>/latest/etc/atslapd/atslapd.crt
  TLSCertificateKeyFile <SenSage_Home>/latest/etc/atslapd/atslapd.key
  ```

- `ldap.conf`—configures the EDW use of LDAP servers. It is located in:

  ```
  <Sensage_Home>/latest/etc/sls
  ```

  Relevant lines in the `ldap.conf` file for all the EDW hosts:

  ```
  # tell EDW whether to verify the certificate presented by the LDAP server.
  # If this option is set to "yes" and the LDAP server cert does not match
  # its hostname or is not signed by a CA trusted by the SLS,
  # SLS authenticationwill fail.
  ssl-verifycert=yes

  # The file containing the PEM-encoded certificate of the
  # certificate authority (CA) which issued the LDAP server cert.
  ssl-cacert=

  # The directory containing one or more PEM-encoded certificates
  # from CAs which are trusted by SLS.
  ssl-cacertdir=
  ```

USERS, ROLES, AND PERMISSIONS

This subsection describes the following topics:

- “Overview”, next
- “Special Users and Roles”, on page 219
- “Managing Access to EDW and HawkEye AP Console Objects”, on page 220
- “Role-Based Access to Functionality in the HawkEye AP Console”, on page 224
- “HawkEye AP Console: Creating and Managing Users, Roles, and Permissions”, on page 227
- “Predefined Users and Roles”, on page 227
- “Managing Access to EDW and HawkEye AP Console Objects”, on page 220

Overview

The HawkEye EDW and other HawkEye AP components authenticate and authorize data access by checking users, roles, and permissions. Users, roles, and permissions are related in the following ways:

- Roles and their associated EDW Actions and NameSpaces permissions specify what areas of the EDW, HawkEye AP Console, and event-log data users can access and what actions they can perform.
- View, Edit, and Run permissions can be granted for each role associated with a report, dashboard, or folder.
- EDW Actions and NameSpaces permissions are granted to roles.
- Users are assigned to roles and inherit each role's EDW Actions and NameSpace permissions.
- An administrator can assign users to multiple roles and set permissions for the roles.
- An administrator cannot grant a permission directly to a user.

HawkEye AP provides two tools that enable administrators to manage users and roles:

- The `atmanage` command-line utility, which enables the administrator to:
  - add and delete users and roles
  - manage the association between users and roles
  - control access to reports
  - test a user's permissions
  - create and remove permissions
  - add and remove permissions to roles
  - enable and disable users and roles
  - transfer ownership of a view to another user
- The Security module of HawkEye AP Console, which enables the administrator to:

- add and delete users and roles
- manage the association between users and roles
- grant or revoke EDW Action and NameSpace permissions associated with a role
- set a user’s email address—required to set a schedule that emails scheduled output to the user.
- set or reset a user’s password

For information on using these tools, see:

- “HawkEye AP Console: Creating and Managing Users, Roles, and Permissions”, on page 227
- “Command Line: Creating and Managing Users, Roles, and Permissions”, on page 238

Special Users and Roles

Although most users are specific to a site's requirements, there are some conditions that require special users. Because special users and roles are considered critical, they cannot be deleted. They are always stored exclusively within an EDWinstance and never in the enterprise authority. HawkEye AP provides the following predefined special users and roles:

- The **system** user, assigned the **`system`** role, is used by internal, automated processes to perform work requiring system authorization. The system role is never assigned to an end user; it is strictly for the use of internal, automated processes. The system user has read access to all information and processes within an EDW instance (both from the command line and the HawkEye AP Console). It is always a member of a system authentication group.
- The **administrator** user is assigned to the **administrator** role, which grants full permission to access, modify, and delete all HawkEye AP objects (such as tables and views) in the EDW; is required for a user to administer an EDW instance.

  A user with administrator-role privilege can grant any of the existing permissions to any role. This user can also assign another user to the `administrator` role. When individual users are assigned to the administrator roles, they have the full privileges granted to the roles. Individual users, however, can be deleted. Because the special administrator user exists independently of individual users, this user still exists even if all users with administrator privileges have been deleted. The inability to delete the administrator user prevents system lockout. For more information, see “Predefined Users and Roles”, on page 227.
- The `guest` user**,** assigned the `guest` role, does not have any permissions by default. Never assign the guest role to a user. This role was created only to enable backward compatibility for early HawkEye AP releases.
- The `analyzer.alerts` role grants users access to Security, Exception, and System Alerts. (Users must also have **View** permission for the dashboard containing the Security, Exception, or System Alert Widgets.)

Limitations on Changing Special Roles and Users

The limitations on changing permissions, roles, and users are:

- `system`
  - The `system` user cannot be deleted or disabled

    - The system role cannot be deleted (but it can be disabled)
    - The system role cannot be removed from the system user
- administrator (prevents system lockout)
    - The administrator user cannot be deleted or disabled
    - The administrator role cannot be deleted or disabled
    - The administrator role cannot be removed from the administrator user
- guest
    - The guest user cannot be deleted or disabled
    - The guest role cannot be deleted (but it can be disabled)
    - The guest role cannot be removed from the guest user

For more information, see “Enabling and Disabling Users and Roles”, on page 240.

Managing Access to EDW and HawkEye AP Console Objects

To access the EDW and the event-log data it stores, a user must have:

- A valid HawkEye AP user name and password
- Membership in a role with the appropriate namespace and EDW Action permissions

To access HawkEye AP Console objects such as reports, dashboards, folders, and alert widgets, a user must also have permission specific to the object.

This section describes the following topics:

- “Namespace Permissions”, next
- “EDW Actions Permissions”, on page 221
- “HawkEye AP Console Object Permissions”, on page 222
- “Creating and Granting Permissions for HawkEye AP Console Objects”, on page 226

Namespace Permissions

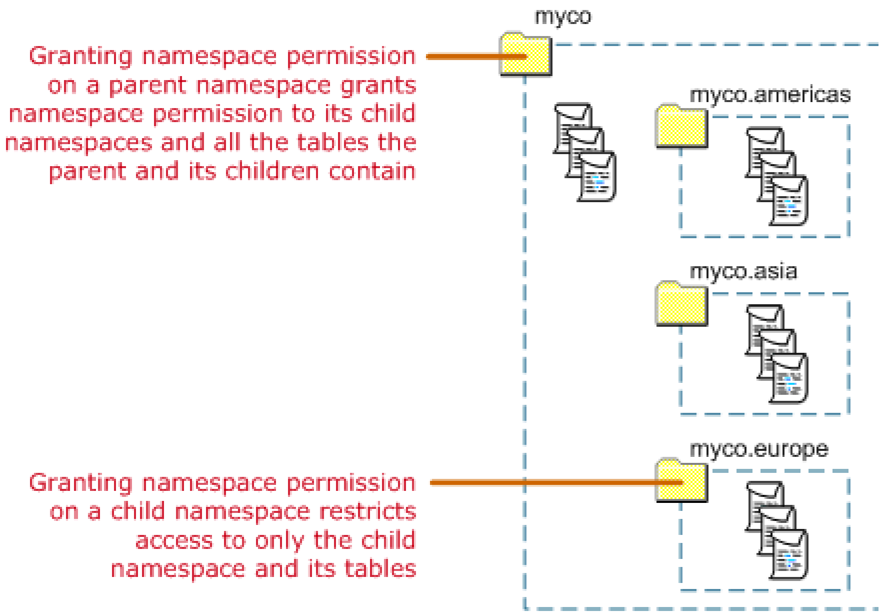
The EDW organizes tables and views within a hierarchy of namespaces. Administrators specify the namespace for tables and views when they create them. If no namespace is specified, the table is created in the default namespace. A namespace can contain multiple tables, but a table resides within a single namespace. Namespaces allow the EDW to contain many tables with the same names, provided tables with the same name are in different namespaces. More importantly, namespaces allow an administrator to apply different access control to different groups of tables and to different user roles.

Users can access tables and views only in those namespaces to which their assigned roles have been granted the namespace permission. When administrators grant the namespace permission, it applies to the specified namespace and any namespaces within it. For example, assume your organization has the root namespace mcyo, which contains three subsidiary namespaces: myco.americas, myco.europe, and myco.asia. If you grant a role permission to the myco namespace, a member of that role can access tables in all four namespaces. If you grant a role namespace permission only on `myco.americas`, a member of that role can access tables only in that namespace; the user cannot access tables in `myco`, `myco.europe`, or `myco.asia`.

Typically administrators grant namespace permission on target namespaces that are deep within the namespace hierarchy. For example, to limit London employees to access tables only in `myco.europe.london`, grant them namespace permission only to this namespace, meaning that users in London can access only the tables in the `myco.europe.london` namespace.

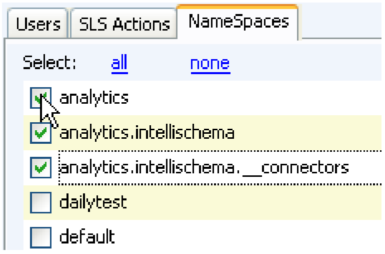
Figure 8-2 illustrates how namespace hierarchies work.

Figure 8-2: Namespaces Illustrated


myco
Granting namespace permission on a parent namespace grants namespace permission to its child namespaces and all the tables the parent and its children contain
myco.americas
myco.asia
myco.europe
Granting namespace permission on a child namespace restricts access to only the child namespace and its tables

In HawkEye AP Console, you enable namespaces for each role. HawkEye AP Console automatically enables child namespaces when its parent is selected. You can, however, select only the child namespace. As shown in Figure 8-3, when you select the `analytics` namespace, the child namespaces of `analytics.intellischema` and `analytics.intellischema._connectors` is also selected automatically.

Figure 8-3: Namespace Selection



☑ analytics
☑ analytics.intellischema
☑ analytics.intellischema.__connectors
☐ dailytest
☐ default

EDW Actions Permissions

A role must have explicit EDW Actions defined for the if EDW Actions enable users to work with the tables and views in the namespace(s) where the role has been granted NameSpace permissions.

In order to perform operations on reports and schedules (such as running a report or modifying a schedule), the roles to which users are assigned must have the proper namespace permissions and the `sls.select` EDW Action permission. To cancel a report query after it has been submitted, the user's role must have `sls.canceltask` EDW Action permission.

These are the only task-based permissions required by HawkEye AP Console operations. The other predefined permissions, while important from the EDW perspective, have no effect on a user's ability to use HawkEye AP Console.

NOTE: Note: By default, the `analyzer.reports` role has the `sls.select` EDW Action permission.

You can grant the following EDW Action Permissions to a role:

- `sls.admin`—users can view and manipulate authorization
- `sls.create`—users can create EDW objects, such as tables, views, and column filters
- `sls.rename`—users can rename EDW tables and views
- `sls.drop`—users can drop EDWobjects, such as tables and column filters
- `sls.canceltask`—users can cancel EDW tasks in progress
- `sls.compact`—users can compact EDW tables
- `sls.load`—users can load data into EDW tables
- `sls.retire`—users can retire/delete rows from EDW tables
- `sls.select`—users can select data from EDW tables and views

All these permissions are granted to the Administrator role.

HawkEye AP Console Object Permissions

To access reports, dashboards, and folders, a user must be a member of a role that has been granted the appropriate EDW Action permissions, Namespace permissions, and HawkEye AP Console Object permissions for each dashboard, report, or folder. When an administrator grants HawkEye AP Console Object permissions to a role for a specific dashboard, report, or folder, you specify whether users associated with the role can View, Edit, or Run the HawkEye AP Console object.

The EDW maintains the namespace and task-level permissions. However, the Application Manager maintains permissions over the reports, folders and dashboards it manages. You manage permissions on reports, dashboards, and folders using the Options Pane in Reports or Dashboard mode of HawkEye AP Console by assigning Run, View, or Edit permissions to each role associated with the report, dashboard, or folder. You cannot manage them with command-line utilities. For more information, see “Managing Access to HawkEye AP Console Reports, Dashboards, and Folders”, on page 237.

Any user who has been granted **Edit** permission on the parent folder can grant the following HawkEye AP Console Object permissions to roles associated with reports, dashboards, and folders:

- View

- Edit
- Run
- View, Run
- View, Edit
- Run, Edit
- All (View, Edit, Run)

DESCRIPTION OF VIEW, EDIT, AND RUN PERMISSIONS

The View, Edit, and Run permissions allow users to do the following, depending on the context:

Permission	Context			
	Reports	**Report Folders**	**Dashboards**	**Dashboard Folders**
View	• View report cache entries • View the SQL and Search Criteria	• View shortcut cache entries for reports to which the users has View permission • View the Folder in the navigator	• View the dashboard and its widgets • The user can only view report and alert widgets for which the user has View permission.	• View the Folder in the navigator • View the dashboards contained in a folder. • The user can only view dashboards for which the user has View permission. • The user can only view report and alert widgets for which the user has View permission.
Edit	• Edit the report definition • Change roles and permissions on the report	• Delete shortcuts to reports for which the user has Edit permission. • Add shortcuts to reports for which the user has View, Edit, or Run permission. • Change roles and permissions on the folder	• Change a dashboard. • The user can only add reports for which the user has View, Edit, or Run permission. • The user must have **analyzer.alerts** permission to add alert widgets to the dashboard. • Change roles and permissions on the dashboard	• Add or delete dashboards contained in the folder. • The user must have Edit permission on individual dashboards. • Change roles and permissions on the folder
Run	• Run the report	• Run reports contained in the report folder • The user can only run reports for which the user has Run permission.	• Run reports contained in the dashboard • The user can only run reports for which the user has Run permission.	

Managing User Shadow Roles

When you create a new user, HawkEye AP creates both a user and a *shadow* role with the same name. The user is automatically added to the shadow role. Use the shadow role to grant permissions to that user only. By default, shadow roles have no permissions in the Event Data Warehouse (EDW). Use the Security module of HawkEye AP Console or the `atmanage` command-line utility to grant EDW permissions to shadow roles and thereby to the individual user.

For more information, see:

- Security module—“HawkEye AP Console: Creating and Managing Users, Roles, and Permissions”, on page 227

`atmanage` utility—“Command Line: Creating and Managing Users, Roles, and Permissions”, on page 238

As a general rule, use shadow roles only for granting permissions to individuals. Although you can add other users to someone's shadow role, you obscure the purpose of shadow roles when you do. Delete shadow roles as soon as you create users, unless you intend to use the shadow roles right away for individual permissions. Instead, create roles related to specific functions within your enterprise and assign new users to them.

IMPORTANT: If you delete a user’s shadow role without also assigning the user to a role so that the user belongs to no role, the user will not display in the Security Tree user list in HawkEye AP Console. If an administrator later attempts to recreate the user, the attempt fails because HawkEye AP Console sees the new user as a duplicate of the existing one. To fix this problem, an administrator must use the `atmanage` utility to assign the user to a role.

Considerations:

- With large systems, the number of roles can become quite large.
- Administrative overhead increases as the number of roles increases.
- Use function-specific roles to grant users the permissions they require instead of using the user-specific (shadow) roles created automatically when a new user is created.

After you identify the roles your enterprise needs, follow this general procedure for adding new users from the Security module:

1. Add a user
2. Add the user to appropriate enterprise roles that you defined.
3. Remove the default, user-specific shadow role created when you added the user.

Role-Based Access to Functionality in the HawkEye AP Console

You use roles to control functionality and data that users are allowed to access. Each role is granted a set of EDW Action permissions that control the user’s access to the EDW and Namespace permissions that control users access to data by namespace. In addition, you apply "HawkEye AP Console Object" permissions to roles associated with each report, dashboard, or folder to control access to those objects. For more information on setting permissions for these objects, see Viewing and Assigning Report, Dashboard, and Folder Permissions in Chapter 3, “Running, Viewing, and Managing Reports” in the *Reporting Guide.*

HawkEye AP provides the following two special roles:

- `administrator`–allows access to all HawkEye AP functionality
- `analyzer.alerts`–allows access to Security, Exception, and System Alerts (users must also have **View** permission for the dashboard containing the Security, Exception, or System Alert Widgets.
- `analyzer.reports`—allows access to Reports mode and grants Read permission to view folders within the All Reports Definition folder, however, a user having this role can only view the folders and cannot view the reports or report links.
- `analyzer.reports.creator`—allows the user to create a report. When a user creates a new report, Write permission is assigned to this role.

Default Roles and Permissions

You use folders to establish default permissions for reports, dashboards, and folders. For each folder you can grant the following permissions for each available role:

- View—allows the user to view a report or dashboard, or to view a list of items contained in the folder
- Edit—allows the user to modify a report definition, dashboard, or folder
- Run—allows the user to run a report or dashboard

The initial permissions for a report, dashboard, or folder are determined by the permissions set on the folder where the new report, dashboard, or folder is created. In a new installation of HawkEye AP software, the roles and permissions are defined as shown in Table 8-1.

Table 8-1: Initial roles and permissions

Folder	Role	HawkEye AP Console Object Permission
Dashboard folders (created at the top-level)	administrator	View, Edit, Run
	analyzer.dashboards	View
Report folders (created at the top-level)	administrator	View, Edit, Run
	analyzer.reports.creator	View, Edit, Run
	analyzer.reports	View
All Report Definitions folder	administrator	View, Edit, Run
	analyzer.reports.creator	View, Edit, Run
	analyzer.reports	View

Any new reports, dashboards, or folders you create in the above folders will take on the roles and permissions indicated in Table 8-1. If you want a different set of initial roles and permissions, a HawkEye AP administrator can create new folders under these folders that have a different set of permissions where users can create new reports, dashboards or folders that have the desired set of initial permissions.

IMPORTANT: Note the following regarding default permissions granted to new objects:

- A user assigned to the administrator role can change the permissions on the All Reports Definitions folder. These permissions are copied to any new reports created by HawkEye AP users.
- When you drag a report from the All Reports Definitions list into a folder, you create a shortcut to the report. The permissions set on the report are copied to the report shortcut. To run, view, or edit a report, a user must have appropriate permissions for both the shortcut and the actual report itself.
- Default permissions for reports and dashboards created at the top level are as shown in Table 8-1 and cannot be changed.

Creating and Granting Permissions for HawkEye AP Console Objects

To grant a user View, Edit, or Run permissions for a report, dashboard, or folder, you use HawkEye AP Console Reports or Dashboards mode to select the object, and then you use the Options pane to grant View, Edit, or Run permission to the roles. Users who are members of these roles will be granted or denied access based on these permissions.

For more information, see Viewing and Assigning Report, Dashboard, and Folder Permissions in Chapter 3, “Running, Viewing, and Managing Reports” in the *Reporting Guide*

Predefined Users and Roles

HawkEye AP provides several predefined users and roles:

- The `guest` user and `guest` role
- The `system` user and `system` role
- The `administrator` user

These special roles do NOT have corresponding special users:

- `analyzer.alerts`
- `analyzer.reports`
- `analyzer.dashboards`
- `analyzer.reports.creator`

For more information about these users and roles, see “Special Users and Roles”, on page 219 and “Limitations on Changing Special Roles and Users”, on page 219.

HawkEye AP Console: Creating and Managing Users, Roles, and Permissions

An administrator can use the Security Module of HawkEye AP Console Administration mode to manage users, roles, and permissions granted to roles.

Accessing the Security Module

The Security Module is available from in Administration Mode in HawkEye AP Console.

To access the Security module

1. Open HawkEye AP Console
2. Click **Administration** in the Toolbar
3. In the Navigator (on left side), click **Security**.

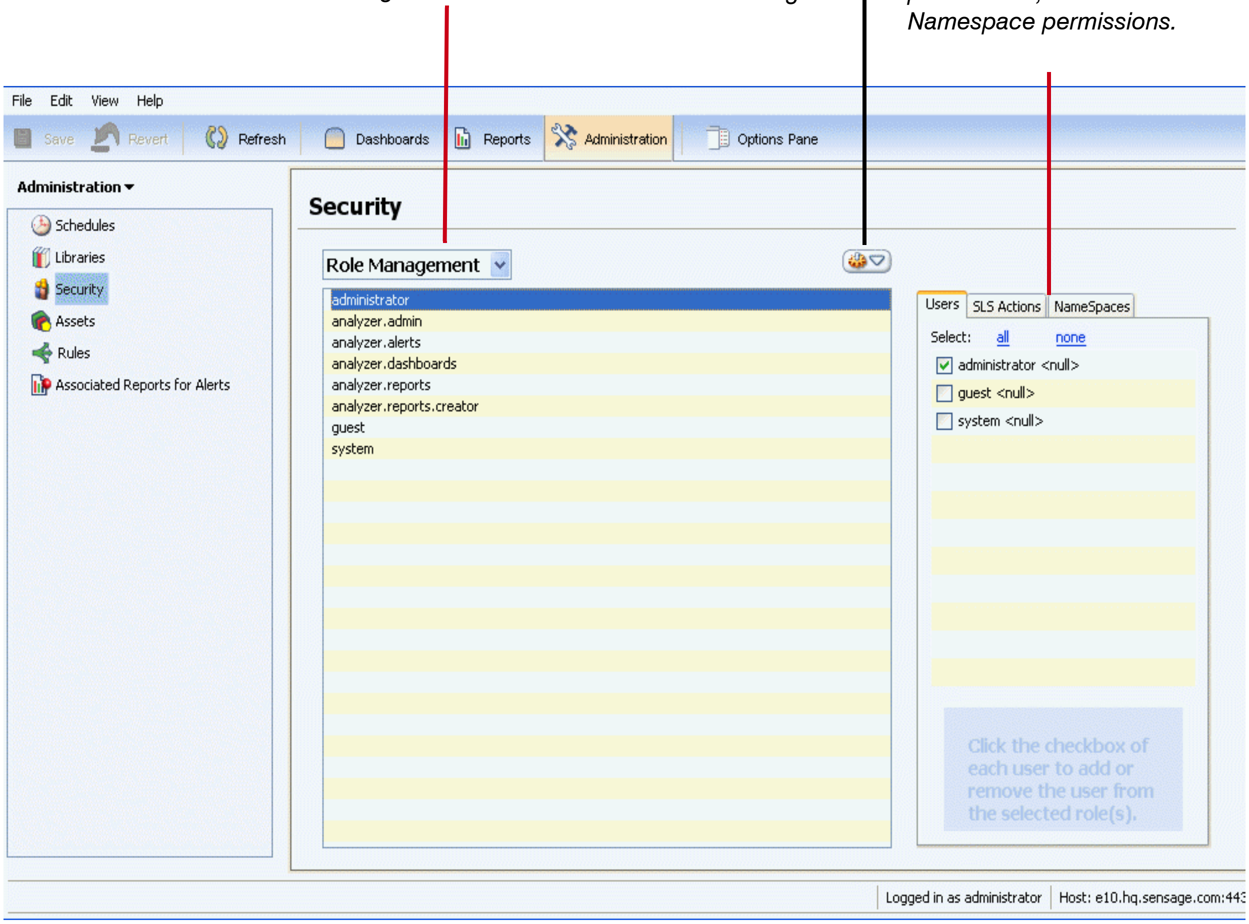
   The Security Module displays, as shown in Figure 8-4.

Figure 8-4: Security Module Window

*The **Security Mode Selector** selects user or role management.*

*The **Action Menu** selects tasks related to user and role management.*

*The **Properties panel** allows you to associate roles with users, SLS action permissions, and Namespace permissions.*



administrator
analyzer.admin
analyzer.alerts
analyzer.dashboards
analyzer.reports
analyzer.reports.creator
guest
system

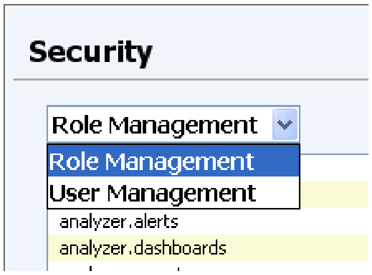
☑ administrator <null>
☐ guest <null>
☐ system <null>

Click the checkbox of each user to add or remove the user from the selected role(s).

Logged in as administrator | Host: e10.hq.sensage.com:443

The Security module has two modes: *User Management* and *Role Management.* You select the mode by using the drop-down list in the Security Mode Selector, as shown in Figure 8-5.

Figure 8-5: Security Mode Selector



analyzer.alerts
analyzer.dashboards

Role Management Mode

Figure 8-4 shows the Security Module in Role Management mode. In this mode, there are Properties panels available on the right. These panels allow you to:

- Add users to a role
- Remove users from a role
- Enable Namespace permission for a role
- Enable EDW Actions for a role

User Management Mode

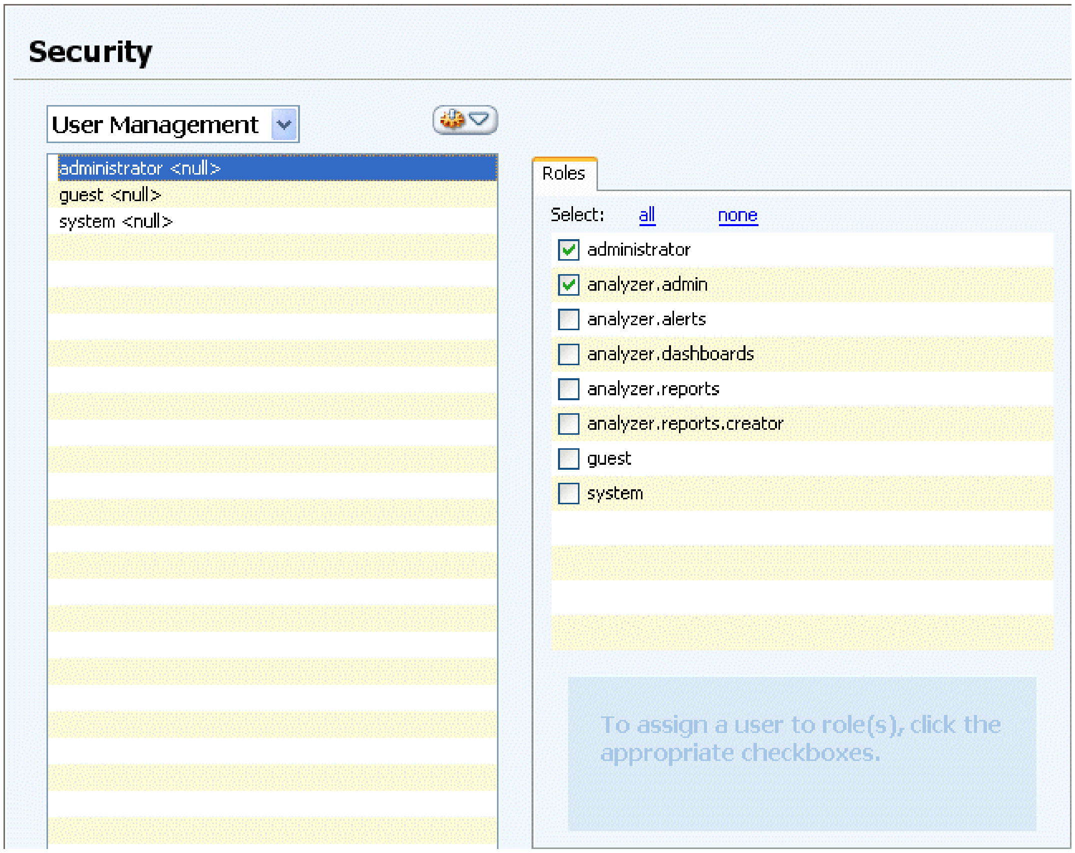
When you select **User Management** mode, a list of users defined in your deployment displays on the left and a single Properties panels displays on the right where you can assign a user to one or more roles.

In this mode you can:

- Add new users
- Edit existing users
- Delete users
- Set passwords for users

For each user, you can edit the user's full name, email address, and password.

Figure 8-6: User Management Mode



Security

User Management

administrator <null>
guest <null>
system <null>

Select: all none

☑ administrator
☑ analyzer.admin
□ analyzer.alerts
□ analyzer.dashboards
□ analyzer.reports
□ analyzer.reports.creator
□ guest
□ system

To assign a user to role(s), click the appropriate checkboxes.

Creating and Managing Users and Roles

An administrator or a user assigned to the administrator role can create roles and users and manage the relationships between them. This topic describes how to use the Security module to create and manage users and roles. To learn how to perform these tasks through the command line instead, see "Command Line: Creating and Managing Users, Roles, and Permissions", on page 238.

NOTE: Users must be assigned to the administrator role to access Administration mode in HawkEye AP Console.

When you use HawkEye AP Console to add a new user, you must supply the following information:

Field	Description
Username	The login name of the user. This field cannot be edited except when you create a new user.
Full Name	The full name of the user.

Field	Description
Email Address	The user's email address. An email address is required to email scheduled items such as reports, dashboard, and folders and to email notifications of query completion to the user.
Password	The password for this user. **IMPORTANT:** If you do not set a password when you create a new user, the individual cannot log in. Always assign a default password, such as `changeme`, for each user you create.
Role Membership	A checklist of available roles to which you can assign the selected user. A selected role indicates that the user has been assigned to the role. Changes made here are reflected the Role Membership Checklist for the role.

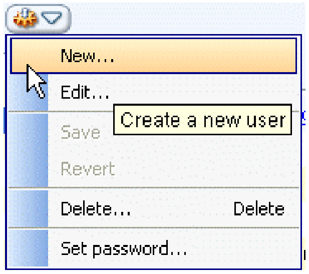
Adding a User

To add a user:

1 Select User Management from the Security Mode Selector

2 Click the **Action Menu** and select **New ...**

Figure 8-7: Adding a User



The **New User** dialog box displays.

3 In the **New User** dialog box:

a Enter the user's full name.

b Enter the User ID for this user.

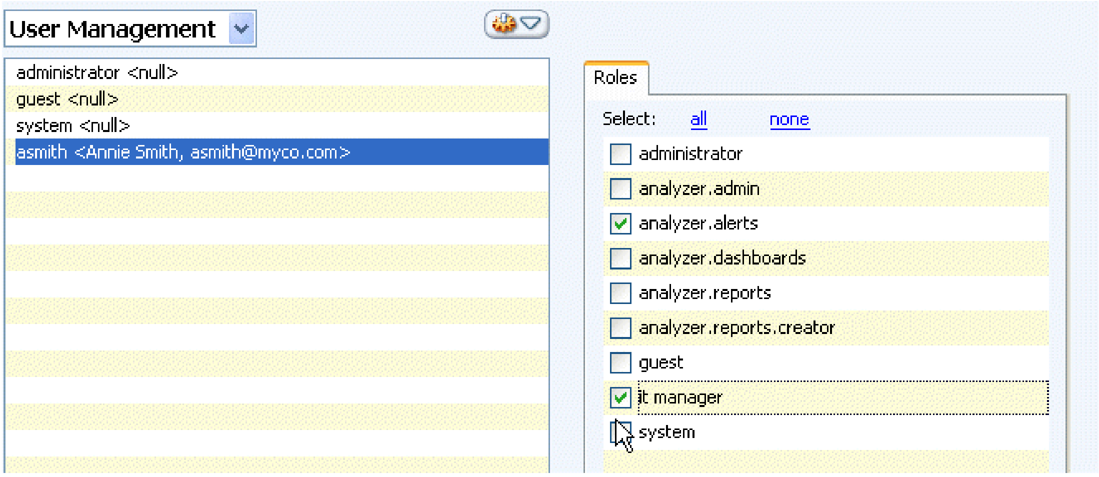
c (Optional) Enter the users’s email address.

d Enter the password for this user, and retype the password in the Re-type password box. A user must have a password to access HawkEye AP Console.

e Click OK.

f In the **Roles** Properties panel, assign the user to desired roles by selecting the check boxes next to the role. Use the all or none links to select and deselect all roles.

Figure 8-8: Assigning Roles to Users



User Management

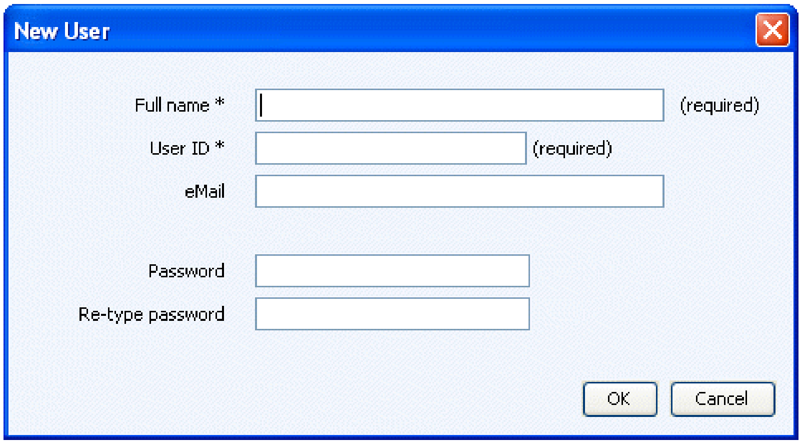
administrator <null>
guest <null>
system <null>
asmith <Annie Smith, asmith@myco.com>

Roles

Select: all none

- ☐ administrator
- ☐ analyzer.admin
- ☑ analyzer.alerts
- ☐ analyzer.dashboards
- ☐ analyzer.reports
- ☐ analyzer.reports.creator
- ☐ guest
- ☑ it manager
- ☐ system

Figure 8-9: New User Dialog Box



New User

Full name * ________ (required)

User ID * ________ (required)

eMail ________

Password ________

Re-type password ________

4 Click the **Action Menu** and select **Save**.

Modifying a User

You can change the name and email address of a user.

To modify a user

1 Select User Management from the Security Mode Selector.

2 Click the user's name in the list of users.

3 Click the Action Menu and select **Edit** (or Right-click on the user and select **Edit**)

The Edit User dialog box displays.

4 In the dialog box, change the desired values.

5 Click **OK**.

6 Change any Role assignments as needed. Use the all or none links to select and deselect all roles.

7 Click the **Action Menu** and select **Save**

Changing a User’s Password

To change a user’s password

1 Select User Management from the Security Mode Selector.

2 Click the user's name in the list of users.

3 Click the Action Menu and select **Set Password...** (or Right-click on the user and select **Set Password...**)

The **Set Password** dialog box displays.

4 Enter the users’s current password in the **Old Password** field. Note that for an administrator, the system ignores any value entered for the current password.

5 Enter the user’s new password in the **New password** field, and enter it again in the **Re-type new password** field. If the values of the New password fields do not match, you will be prompted to re-enter them.

6 Click **OK**.

7 Click the **Action Menu** and select **Save.**

Deleting a User

To delete a user:

1 Select User Management from the Security Mode Selector.

2 Click the user's name in the list of users.

3 Click the Action Menu and select **Delete** (or Right-click on the user and select **Delete**)

A confirmation dialog box displays

4 Click **OK** to confirm you want to delete this user.

Creating and Managing Roles

When you create a new role, you supply the following information:

- The name of the role
- Users who should be assigned to the new role
- EDW Action permissions granted to the new role
- Namespace permissions granted to the new role

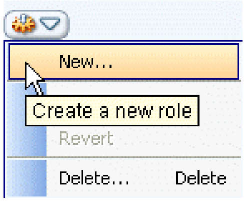
Adding a Role

To add a new role:

1 Click the Security Mode Selector and select Role Management.

2 Click the Action Menu and select **New ...**

Figure 8-10: Adding a role



The New Role dialog box displays.

3 In the New Role dialog box, enter a unique role name.

4 Click **OK**.

5 Select the Users tab in the Properties panel

6 Assign existing users to the role by selecting the checkbox next to the user’s ID.

7 Select the EDW Actions tab.

8 Enable actions for the new role by selecting the checkbox next to the EDW Actions.

9 Select the NameSpaces tab in the Properties panel.

10 Enable namespace permissions for the new role by selecting the checkbox next to the namespace.

11 Click the Action Menu and select **Save**.

Modifying a Role

To modify a role:

1 Click the Security Mode Selector and select Role Management.

2 Select role's name from the list of Roles displayed on the left.

3 Select the Users, EDW Actions, or NameSpace tabs and change the selected values as needed. Use the all or none links to select and deselect all roles, Namespaces, or EDW Actions.

4 Click the Action Menu and select **Save**.

Deleting a Role

To delete a role:

1 Click the Security Mode Selector and select Role Management.

2 Select role's name from the list of Roles displayed on the left.

3 Click the Action Menu and select Delete (or right-click on the role and select Delete).

Granting Namespace and Task Permissions to Roles

To use HawkEye AP Console to grant namespace permission to a role

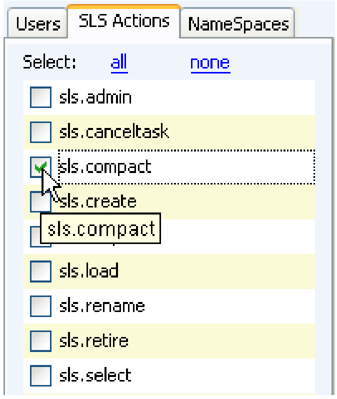
1 Click the Security Mode Selector and select Role Management.

2 Select role's name from the list of Roles displayed on the left.

3 Select the EDW's SLS Actions Tab in Properties panel.

4 Select which SLS Actions you want to assign to this role by selecting the checkbox next to their name. Use the all or none links to select and deselect all EDW Actions.

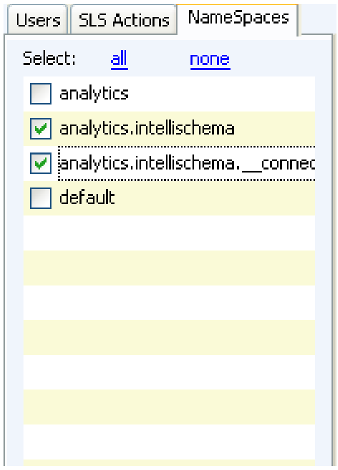
Figure 8-11: Enabling EDW Actions



□ sls.admin
□ sls.canceltask
☑ sls.compact
□ sls.create
□
□ sls.load
□ sls.rename
□ sls.retire
□ sls.select

5 Select the NameSpaces tab in the Properties panel. (See Figure 8-11.)

6 Enable namespace permissions you want to assign to this role by selecting the checkbox next to the namespace. Selecting a parent namespace also selects any children namespaces. Use the all or none links to select and deselect all Namespace permissions.



□ analytics
☑ analytics.intellischema
☑ analytics.intellischema.__connec
□ default

.

7 Click the Action Menu and select **Save**.

Assigning Users to Roles

From the Security module of Administration mode, you can assign a user to a role in either of two ways:

1 Select User Management from the Security Mode Selector.

2 Click the user's name in the list of users.

3 Select the check boxes of roles to which you want to assign the user and deselect check boxes of roles you want to revoke. Use the all or none links to select and deselect all roles.

4 Click the Action Menu and select **Save.**

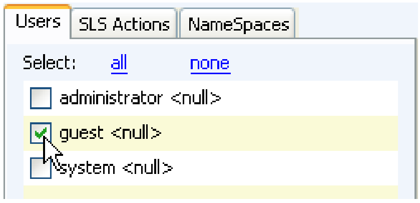
OR (alternate procedure)

1 Click the Security Mode Selector and select Role Management.

2 Select role's name from the list of Roles displayed on the left.

3 Select the Users tab in the Properties panel.

4 Select the check boxes of users you want to add the role and deselect check boxes of users you want to exclude from the role. Use the all or none links to select and deselect all users.



□ administrator <null>
☑ guest <null>
□ system <null>

5 Click the Action Menu and select **Save**.

Managing Access to HawkEye AP Console Reports, Dashboards, and Folders

After you configure users and assign them to roles, you can use these roles to manage access to dashboards, reports, and folders in HawkEye AP Console.

This section describes the following topics:

- "Ownership and Permissions for Reports", next
- "Ownership and Permissions for Schedules", on page 237
- "Ownership and Permissions for Dashboards", on page 237
- "Setting Permissions on Reports", on page 237
- "Setting Permissions on Report and Dashboard Folders", on page 237
- "Role-Based Access to Functionality in the HawkEye AP Console", on page 224

Ownership and Permissions for Reports

Reports do not have owners. To create a report, a user must belong to a role that is granted **Edit** permission for the folder where the report is created. Only an administrator or a user with **Edit** permission can change the report's permissions.

Ownership and Permissions for Schedules

Schedules do not have assigned owners or permissions.

Ownership and Permissions for Dashboards

Dashboards do not have assigned owners, but they do have permissions. When a dashboard or dashboard folder appears in a folder, a user who has Edit permission on the folder or who is assigned to the administrator role can set permissions on dashboards and dashboard folders.

Setting Permissions on Reports

You can set permissions on reports to restrict the type of access a user has to a report. For example, one user may have Edit permission that allows modification of a report while another user may have only View permission that allows the user to see only the output of the report. An administrator or a user granted Edit permission grants this access by assigning specific View, Edit, and Run permissions to specific roles using the Options Pane of a report definition.

For more information, see "Role-Based Access to Functionality in the HawkEye AP Console", on page 224.

Setting Permissions on Report and Dashboard Folders

When you assign View, Edit, and Run permissions to report folders, these permissions affect new reports and dashboards created in the folder or existing reports and dashboards that are moved into the folder. You must assign permissions to each report or sub-folder individually. You can grant or revoke permissions defined for the folder for each report, dashboard, or sub-folder contained in the folder.

For more information, see "Role-Based Access to Functionality in the HawkEye AP Console", on page 224.

Command Line: Creating and Managing Users, Roles, and Permissions

An administrator can use the `atmanage` utility to create and manage users, roles, and permissions. This utility is described in detail in "Managing an EDW Data Store", on page 111. An administrator can use the `atview` utility to display information about users, roles, and permissions. This utility is described in detail in "Examining the State of an EDW Data Store", on page 125.

The topics below present only examples of using command-line utilities to manage users, roles, and permission:

- "Creating and Assigning Users, Roles, and Permissions", next
- "Enabling and Disabling Users and Roles", on page 240
- "Listing Users, Roles and Permissions", on page 240
- "Changing Passwords", on page 241

Creating and Assigning Users, Roles, and Permissions

Users assigned to the administrator role use the `atmanage` utility to create users, assign users to roles, and add permissions to roles. The command has the following general syntax:

```
atmanage --user=<username> --pass=<password> <SLS-host>:<SLS-port> <action>
```

Where `<action>` can be one of the following security actions, discussed in this section:

- `adduser`
- `addrolepermission`
- `adduserrole`
- `setuserstate`
- `setrolestate`
- `changepassword`

NOTE: The name of a user or role can include any of the following alphanumeric characters (A-Za-z0-9_ .). Sensage recommends that you limit the name to 30 characters.

IMPORTANT: After you run the `atmanage` command to add or modify a user, the changes do not take effect for up to 10 minutes after the command executes. If you need to put your changes into effect immediately, you can restart the `appserver` component, but this will interrupt any in-progress work while the `appserver` component restarts. To restart the appserver, issue this command:

```
clsetup restart rt appserver
```

Creating a User Account in an SLS Instance

The example below creates two users, *chrissy* and *joe*:

```
atmanage --user=administrator --pass=s0mep@ss \
  localhost:8072 adduser chrissy chrissyp@ss

atmanage --user=administrator --pass=s0mep@ss \
  localhost:8072 adduser joe joesp@ss
```

In the example above, the administrator logged in to the host where the cluster's instance exists. Because the administrator is running the `atmanage` command from the same host as the instance, she used `localhost` and the port number of that instance.

Creating a Role in an SLS Instance

Create a role that defines the level of access you want to grant for a user, or group of users. The example below creates an `analyst` role:

```
atmanage --user=administrator --pass=s0mep@ss \
  localhost:8072 addrole analyst
```

Assigning Permissions to a Role

For the `analyst` role created above, assume users need only to query tables (`sls.select` permission) and cancel queries in progress (`sls.canceltask` permission). These permissions enable users to run and cancel running reports from HawkEye AP Console.

Assigning Task Permissions

The examples below assign two predefined permissions to the `analyst` role. For a list of these permissions, see “Default Roles and Permissions”, on page 225. For information on creating your own permissions, see “addpermission”, on page 117.

```
atmanage --user=administrator --pass=s0mep@ss localhost:8072 \
  addrolepermission analyst sls.select true

atmanage --user=administrator --pass=s0mep@ss localhost:8072 \
  addrolepermission analyst sls.canceltask true
```

Assigning Namespace Permissions

The examples below illustrate assigning different namespace permissions to different roles. The role to which users are assigned determines the scope of their access to company data.

- The following command grants London-only-level namespace permission to the `rptlondon` role (limits access to a single namespace):

```
atmanage --user=administrator --pass=changeme host07:8072 \
addrolepermission rptlondon sls.namespace myco.europe.london
```

- The following command grants access to all European namespaces to the `eur-mgr` role (the role assigned to European management):

```
atmanage --user=administrator --pass=changeme host07:8072 \
addrolepermission eur-mgr sls.namespace myco.europe
```

Assigning Users to a Role

The example below assigns *chrissy* and *joe* to the `analyst` role:

```
atmanage --user=administrator --pass=s0mep@ss localhost:8072 \
  adduserrole chrissy analyst

atmanage --user=administrator --pass=s0mep@ss localhost:8072 \
  adduserrole joe analyst
```

The examples above only show a small portion of the commands available to manage users, roles and permissions. For full details on those commands, see "Authentication Management", on page 116.

Enabling and Disabling Users and Roles

By default, all users and roles are enabled as soon as you create them. The `atmanage` utility provides two arguments that allow you to disable or enable specific users and roles: `setuserstate` and `setrolestate`.

Disabling a User

The example below illustrates how to disable a user named `georgem`.

```
atmanage --user=administrator --pass=pass:s0mep@ss localhost:8072 \
setuserstate georgem disable
```

Disabling a Role

The example below illustrates how to disable the `guest` role.

```
atmanage --user=administrator --pass=pass:s0mep@ss localhost:8072 \
setrolestate guest disable
```

Listing Users, Roles and Permissions

An administrator can use `atquery` to list the users, roles, and permissions currently available on the system. For example:

Listing All Users

```
atquery --user=administrator --pass=pass:s0mep@ss \
  localhost:8072 --namespace='system' -expression='SELECT * FROM users;'
```

In the example above, the administrator logged in to the host where the cluster instance is located; its port is 8072. To query the `users` table in the `system` namespace, the administrator specified `--namespace` as `'system'`. To execute the SQL `SELECT` statement, the administrator used the

`--expression` execute option. The command above selects all data from the `users` table. The examples below use this syntax to query other tables.

Listing Roles

Listing All Roles

```
atquery --user=administrator --pass=s0mep@ss localhost:8072 \
  --namespace='system' -e='SELECT * FROM roles;'
```

Listing All Roles Assigned to All Users

```
atquery --user=administrator --pass=s0mep@ss localhost:8072 \
  --namespace='system' -e='SELECT * FROM userroles;'
```

Listing Permissions

```
atquery --user=administrator --pass=s0mep@ss localhost:8072 \
  --namespace='system' -e='SELECT * FROM permissions;'
```

Listing All Permissions Assigned to All Roles

```
atquery --user=administrator --pass=s0mep@ss localhost:8072 \
--namespace='system' -e='SELECT * FROM rolepermissions;'
```

For more information on `atquery`, see "Querying Data", on page 89.

Changing Passwords

Changing a User's Password

An administrator can use the following `atmanage` command to add or change a user's password:

```
atmanage --user=<admin_user> --pass=<password> <cluster_list> \
changepassword <username> <password>
```

For example:

```
atmanage --user=administrator --pass=s0mep@ss "sls01:8072,sls02:8072,\
sls03:8072" changepassword joe s0mep@ss
```

IMPORTANT:

- You must be tunneled to the EDW host for encryption.
- Only a user that belongs to the administrator role can change another's password. Otherwise, users can change only their own passwords.

Changing Your Own Password

Users can change their own password. See Changing Your Password in Chapter 1, "Getting Started" in the *Reporting Guide*.

Changing the Administrator or Guest Password

An administrator can use the `atmanage` command to change the administrator and guest passwords. The default for these users is "`changeme`".

The examples below change the passwords for the administrator and guest:

```
atmanage --user=administrator --pass=s0mep@ss "sls01:8072,sls02:8072,\
sls03:8072" changepassword administrator adminp@ss

atmanage --user=administrator --pass=s0mep@ss "sls01:8072,sls02:8072,\
sls03:8072" changepassword guest guestp@ss
```

System Tables

The following tables exist in the system namespace to hold users, roles, and permissions:

- `system.users` and `system.users2`—user accounts
- `system.userroles` and `system.userroles2`—roles and the users to which they are attached
- `system.roles` and `system.roles2`—roles
- `system.permissions` and `system.permissions2`—permissions
- `system.rolepermissions` and `system.rolepermissions2`—permissions and the roles to which they are attached

For more information, see “System Tables”, on page 163.

Environment Variables

You can avoid repeatedly including the following options on the command line if you set them in the relevant environment variable:

```
--user ............. ADDAMARK_USER
--pass ............. ADDAMARK_PASSWORD
--shared-secret ..... ADDAMARK_SHARED_SECRET
```

For more information about these options, see “Authentication Options”, on page 112.

Creating and Managing Distribution Filters

Distribution Filters allow a report creator to more efficiently leverage the processing power of the SLS by running a report once and then distributing different parts of the same report to different groups of users. This is more efficient than running the same report multiple times with different parameters for different groups of users.

When configured, distribution filters enable users to see only the rows of the report that are relevant to them. For example, a user in the Accounting department may be limited to viewing rows of data in a report where the hostname is one of the organization's accounting servers, while users in the Manufacturing department can only view rows containing results from hosts belonging to the Manufacturing department.

Roles and Permissions

A user must have the `administrator` role to create Distribution filters using the Administration Mode of HawkEye AP Console. A Distribution Filter contains a column field and a filter expression that are used as criteria to limit the report’s output to rows where the specified column matches the filter expression. The filter also associates one or more roles with the filter. Users assigned to a role associated with a distribution filter can only view rows that match the criteria established in the Distribution filter. For more information, see “Users, Roles, and Permissions”, on page 218.

Any user granted Edit permission for a report can apply a Distribution filter to the report. Users who view the report must belong to a role associated with the filter to see rows that match the filter expression. Other users will not be able to view all rows in the report because the filter will not be applied.

For more information on applying Distribution filters to reports, see Using Distribution Filters to Limit Viewable Data by Role in Chapter 3, “Running, Viewing, and Managing Reports” in the *Reporting Guide.*

Distribution Filter Syntax (Expression and Rules)

Distribution filter expressions are written in standard Postgres SQL using the WHERE clause format, a syntax that makes it possible to distribute the same sections of the report to different users and groups. Although powerful, distribution filter expressions are complex to construct and require that you carefully follow SQL rules specific to the Postgres language when using them. For syntax details, refer to the Postgres SQL Documentation.

Note that when constructing Distribution Filter, syntax rules must be followed precisely. Typically, columns must be double-quoted and their values single-quoted. The column name must be in upper case. If you want to perform a case-insensitive match for last, first (space options) you must use precise syntax rules as in:

```
upper("PATIENT_NAME") ~ upper("last") AND UPPER("PATIENT_NAME") ~ upper("first")
```

With this in mind, following are general rules when creating filter expressions:

1 The valid values for a column depend on the column’s name and datatype.

2 A column that is a standard SQL reserved word, contains spaces, or is in mixed case, must be enclosed in double (" ") quotes.

3 A column that is a text datatype must have values enclosed in single (’ ’) quotes.

4 If a column value contains a single (’ ’) quote, you must add another single quote next to the existing quote; for example: "**Text**" = **’single**’**’quote’.**

5 The list of valid operators varies by datatype.

6 Searches are case-sensitive; to create case insensitive searches, use functions such as **upper** and **lower**.

7 The following are some useful operators:

Operator	Description	Example
<	Less than?	1 < 2
<=	Less than or equal to?	1<= 2
<>	Not equal?	1 <> 2
=	Equal?	1 = 1
>	Greater than?	2 > 1
~~	LIKE	’scrappy,marc,hermit’~~’%scrappy%’

Operator	Description	Example
!~~	NOT LIKE	’bruce’!~~’%al%’
~	Match (regex), case sensitive	’thomas’ ~ **’.thomas.’**
~*	Match (regex), case insensitive	’thomas’ ~ * **’.Thomas.**’
!‘	Does not match (regex), case insensitive	’thomas’ !~ ’**.Thomas.**’
!~*	Does not match (regex), case insensitive	’thomas’ !~ ’**.vadim.**’

NOTE: Functions can be used to concatenate fields together. Beside operators, functions may also be used to make searches case sensitive (lower) and insensitive (upper).

Distribution Filter Examples

1 Create a filter for a numeric column named **Digit** for a value of **5:**

```
"Digit" = 5
```

Note since Digit is a SQL reserved word you must enclose the column name in quotes, Since it’s a numeric column the value does not have single quotes.

2 Create a filter for a text column named **foo** with a value of **abc**:

```
foo = ’abc’
OR
"foo" = ’abc’
```

Note that you must enclose text values in single quotes.

3 Create a filter for a text column named **Link** and find all entries that contain **yahoo**:

```
select * from query_1100 where "Link" ~~ ’%yahoo%’
```

4 Create a filter to match on a text column named **Text** and match a value that equals **single’quote**:

```
select * from query_1100 where "Text" = ’single’’quote’’
```

5 Create a filter where the name column contains upper or lower **donovan:**

```
upper(name) ~~ ’DONOVAN’
```

Developing Distribution Filters

The easiest way to develop a filter is to use a SQL tool and verify that the Distribution Filter works. To do this:

1 Run the report.

2 Find the report’s ID:

```
select id from ss_report where name = ’MyTestReport’;
```

3 Find the report's query_results table. For example, if the **report id =123**, when the report runs a table will be created in the query_result schema named **query_123**. Use a SQL tool such as pgAdmin or DBVisualizer, or PSQL to test the filter against the **query_123** table.

4 Run standard SQL with the WHERE clause containing your filter such as:

```
select * from query_123 where "Digit" = 5
```

NOTE: Be aware of the following tool limitations:

(1) There is no SQL validation when creating/editing the distribution filter.

(2) The tool makes no effort to determine the columns in which distribution filters do not apply. All distribution filters are applied.

Creating Distribution Filters

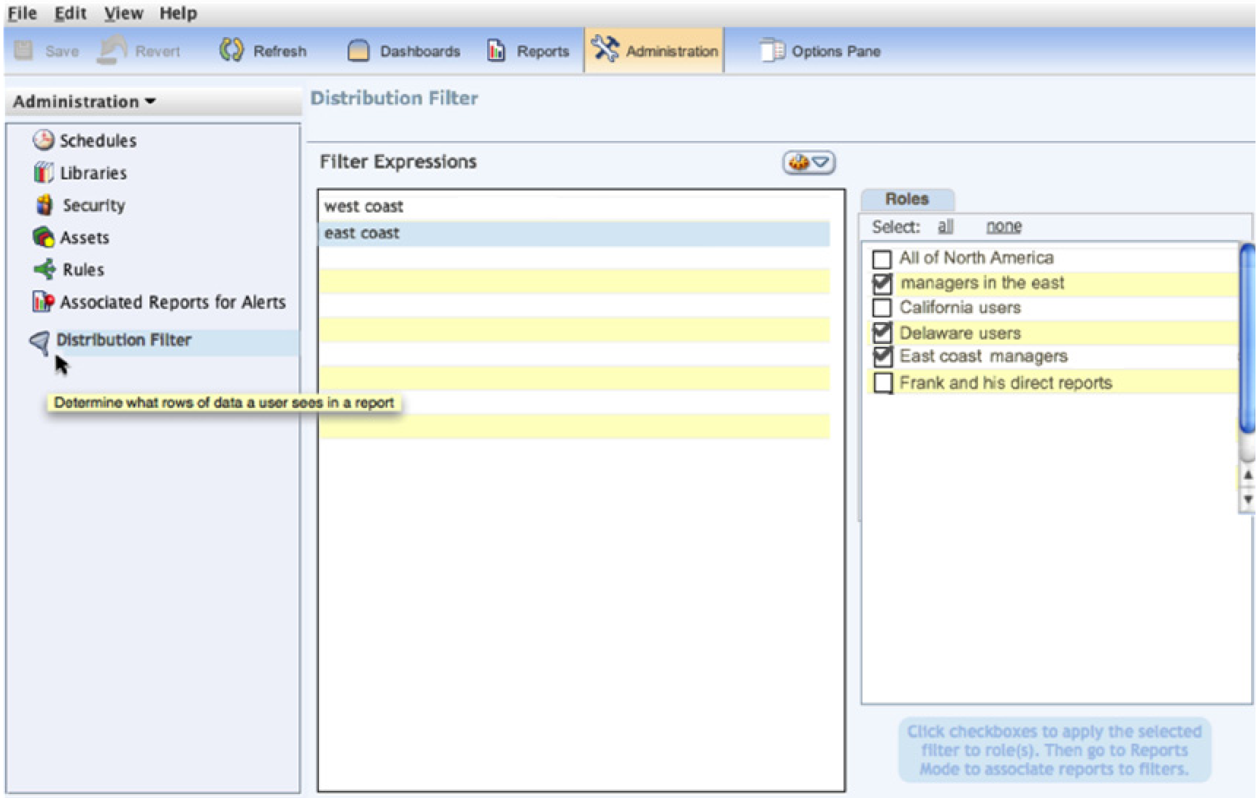
To create a distribution filter

1 Open HawkEye AP Console and navigate to Administration Mode.

2 Select Distribution Filters from the chooser in the left pane.

The **Distribution Filter** module displays.

Figure 8-12: Distribution Filter module



Filter Expressions:

- west coast
- east coast

Roles:

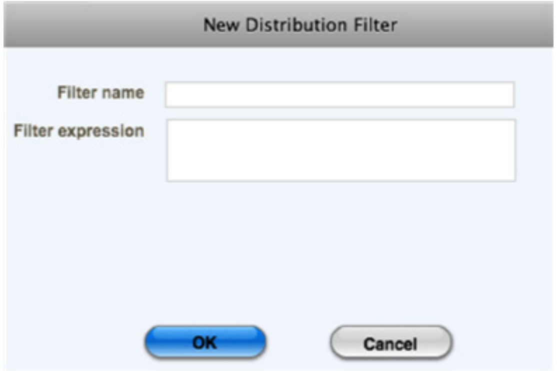
- □ All of North America
- ☑ managers in the east
- □ California users
- ☑ Delaware users
- ☑ East coast managers
- □ Frank and his direct reports

Click checkboxes to apply the selected filter to role(s). Then go to Reports Mode to associate reports to filters.

3 Click the Action menu and select **New ...**.

The **New Distribution Filter** dialog box displays.

Figure 8-13: New Distribution Filter dialog box



4 Enter a name for this Distribution filter in the Filter name field

5 Enter the Filter expression in the **Filter expression** field. This expression must use the syntax of a standard SQL `WHERE` clause. You can use multiple conditions connected by "`AND`" or "`OR`" comparison operators. Enclose literal values in single quotes.

For example:

- `hostname='myHost'`
  (Limits results to rows where the `hostname` column is equal to "`myHost`".)
- `"Day of Week" like 'T%'`
  (Limits results to rows where the "Day of Week" column is Tuesday or Thursday.)
- `"Day of Week" like '%s%'`
  (Limits results to rows where the "Day of Week" column is Tuesday, Wednesday, or Thursday.)
- `login_count>5 AND hostname='myHost'`
  (Limits results to rows where `login_count` is greater than 5 and `hostname` is equal to "`myHost`".)

NOTE: The Filter expression is not verified until a report runs that uses this distribution filter. Make sure you enter a valid `WHERE` clause.

6 Click **OK**.

7 Select the new filter from the list of filters.

8 In the Roles pane on the right, select the roles you want to associate with this Distribution Filter. Users assigned to these roles will only see report data that is filtered by the Filter expression.

9 Click the Action menu and select **Save**, or click the **Save** button on the toolbar.

Editing a Distribution Filter

A user must be associated with the administrator role to edit distribution filters.

To edit a Distribution Filter

1. Open HawkEye AP Console and navigate to Administration Mode.
2. Select Distribution Filters from the chooser in the left pane.

   The **Distribution Filter** module displays.
3. Click the Action menu and select **Edit ...**.
4. Select the Distribution Filter you want to edit.

   The **Edit Distribution Filter** dialog box displays.
5. Change values as needed in the Edit Distribution Filter dialog box.
6. Click **OK**
7. Select the Distribution filter you want to edit.
8. Change any role assignments as needed in the Roles pane.
9. Click the Action menu and select **Save**, or click the **Save** button on the toolbar.

Deleting a Distribution Filter

A user must be associated with the administrator role to delete distribution filters.

To delete a Distribution Filter

1. Open HawkEye AP Console and navigate to Administration Mode.
2. Select Distribution Filters from the chooser in the left pane.

   The **Distribution Filter** module displays.
3. Select the Distribution Filter you want to delete.
4. Click the Action menu and select **Delete ...**.

   A dialog box displays asking you to confirm the deletion.

CHAPTER 9

Archiving to Nearline Storage

This chapter contains the following sections:

- "Nearline Storage: Overview", next
- "Initially Configuring Nearline Storage", on page 252
- "Archiving of Local Data to Nearline Storage", next

NEARLINE STORAGE: OVERVIEW

This topic describes:

- "Models for Managing Local Storage Space", next
- "Types of Nearline Storage Space", on page 250
- "How Nearline Storage Works", on page 251
- "Managing the Archiving to Nearline Storage", on page 252

Models for Managing Local Storage Space

The default configuration of an EDW instance uses local disk space for its storage space. With local storage space, an EDW instance stores and retrieves data very quickly. However, as the amount of stored historical data grows, local disk space can run out.

The EDW component supports two methods for expanding EDW (Event Data Warehouse Server) storage capacity:

- **Archiving data**

  Archiving data requires configuring the EDW component to communicate with nearline storage devices. The EDW then can move old data from local storage onto nearline storage. Old data on local storage is replaced with references to its location on nearline storage. These references let you reclaim local storage space for new data, because references to archived data are much smaller than the data itself.

  Archiving data allows you to reserve local storage for recent data. Queries of recent data execute quickly because the data is on local storage. Historical data remains accessible on nearline storage, but queries execute more slowly. When a person queries the EDW for archived data, it automatically and transparently retrieves the data regardless of whether it is in local or nearline storage.

- **Retiring data**

  Retiring data deletes it from the storage space of EDW instances and reclaims the storage space that it consumed. After you retire data, it is completely gone and is not accessible to queries. Retire data instead of archiving it only if you do not need to access to the historical data that you retire.

IMPORTANT: Retiring data from an EDW table removes only data that is *not* stored under retention on a nearline storage device. The command skips archived data that is under retention and logs a message that provides the date in the future when the nearline storage device will allow you to retire the data. That date is the earliest you can retire the data. The command does remove archived data that is not under retention or its retention period has passed. For more information, see “Retiring Data”, on page 130.

Types of Nearline Storage Space

You can use the HawkEye AP Nearline Storage Server (NSS) to archive data to the following nearline storage devices:

- **EMC® Centera™ Nearline Storage Driver**

  This module allows your organization to take advantage of the unique compliance features that Centera provides. The EDW uses the proprietary Centera API to move data to and from the Centera device.

- **NetApp® SnapLock™ Nearline Storage Driver**

  This module allows your organization to take advantage of the unique compliance features that Network Appliance, Inc. provides. The EDW uses the SnapLock driver to move data to and from the NetApp SnapLock NFS (Network File System) system.

- **Hitachi HCAP® (Hitachi Content Archive Platform) Nearline Storage Driver**

  This module allows your organization to take advantage of the unique compliance features that Hitachi, Ltd. provides. The EDW uses the HCAP driver to move data to and from the Hitachi HCAP system.

- **Fujitsu Eternus® Content Addressable Storage (CAS) System Nearline Storage Driver**

  This module allows your organization to take advantage of the unique compliance features that Fujitsu provides. The EDW uses the proprietary Fujitsu API to move data to and from the Eternus device.

- **Remote NFS (Network) or CIFS (Common Internet) File System**

  Remote file systems allow your organization to archive historical data to regular storage devices on remote hosts. The remote file systems are mounted locally on the hosts where the EDW instance runs. The local mount points on all EDW hosts must be the same, but each mount point can reference a different remote file system.

In summary, nearline storage provides:

- Easier management of storage space. Scheduling the archiving of data to nearline storage helps you maintain sufficient local storage space for EDW instances.

- Easier expansion of storage space. If you require more space, add more storage to your HawkEye AP-supported nearline storage device.

- More evenly balanced data loads on EDW hosts. When you add hosts to an EDW instance, the newer hosts initially have less data on them than the older, existing hosts. Because older data is archived before newer data, older hosts in the EDW instance are generally the first to reclaim local storage space.

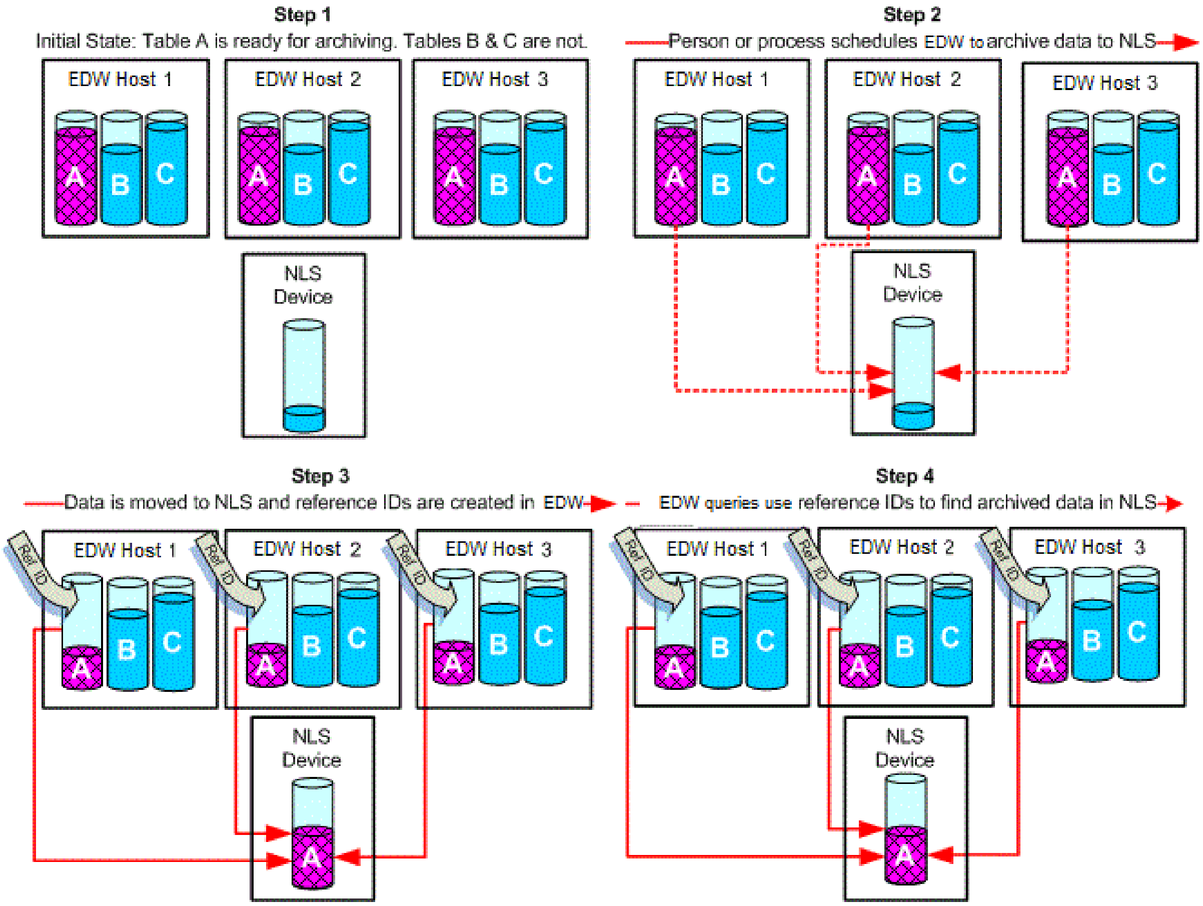
- Automatic retrieval of archived data. When people run queries, they do not need to know whether the are retrieved from local or nearline storage.

How Nearline Storage Works

Figure 9-1illustrates the states of a HawkEye AP nearline-storage archive process.

- **Step 1: Initial State**—The data store for the EDW instance resides on the local hosts, such as the three hosts shown in Figure 9-1.
- **Step 2: Archive Process**—An outside agent (a person or automated process) schedules archiving of EDW data onto the nearline storage device.
- **Step 3: Archive Complete**—After archiving completes, a BLOB (binary large object) that contains the user's data and other associated data is created on the NLS device. A reference ID to the blob is created in each data store of the EDW instance.
- **Step 4: Active** —When a query that requires access to the archived data on the nearline storage device is invoked on the EDW, the EDW uses the reference IDs to locate the data for retrieval.

Figure 9-1: Archiving to Nearline Storage


Step 1
Initial State: Table A is ready for archiving. Tables B & C are not.
EDW Host 1
EDW Host 2
EDW Host 3
A
B
C
NLS Device
Step 2
Person or process schedules EDW to archive data to NLS
Step 3
Data is moved to NLS and reference IDs are created in EDW
Ref ID
Step 4
EDW queries use reference IDs to find archived data in NLS

Managing the Archiving to Nearline Storage

You can manage the archiving of local data in an EDW instance manually or on a scheduled basis. HawkEye AP provides a command-line utility for manually archiving local data on demand.

Before people can manage archiving to nearline storage, a system administrator performs some initial configuration tasks. Thereafter, other people can manage archiving manually or on a schedule.

For more information on the command-line utility, see "Archiving of Local Data to Nearline Storage", on page 271.

Initially Configuring Nearline Storage

Before people can manage the archiving of local data to nearline storage, a system administrator performs these initial configuration tasks:

1. Define and manage nearline targets.

2. Configure the EDW for nearline storage.

The following sections describe these tasks:

- "Defining and Managing Nearline Targets: Overview", next
- "Specifying NSAs for Centera Systems", on page 254
- "Specifying NSAs for SnapLock Systems", on page 258
- "Specifying NSAs for HCAP Systems", on page 263
- "Specifying NSAs for Eternus Systems", on page 266
- "Specifying NSAs for Remote File Systems", on page 268
- "Configure the EDW for Nearline Storage", on page 269

Defining and Managing Nearline Targets: Overview

A *nearline target* is a specific Centera device, NetApp SnapLock NFS system, Hitachi HCAP system, Fujitsu Eternus system, or remote file system. You define a nearline target for the EDW component with two pieces of information:

- **NSI (Nearline Storage Identifier)**—a unique text identifier for a nearline storage target. The format of an NSI is the same regardless of the type of nearline storage.

- **NSA (Nearline Storage Address)**— a connection specifier for a nearline storage target. The format of an NSA differs according to the type of nearline storage device.

After you define nearline targets, people can use them when they manually archive local data from a table or schedule the archiving of local table data.

To define a nearline target, a user with administrator privilege (`sls.admin` permission) uses the `atmanage` utility to specify the nearline storage identifier and network storage address for the target. An administrator also uses the `atmanage` utility to delete nearline targets and to list them. The syntax has three forms:

- Adding a nearline target:

Syntax

```
atmanage <cluster_list> --user=<name> --pass=<pass> setnsi <nsi> <nsa>
```

Examples

```
atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
  --password=changeme   setnsi centera-1 centera://centera1.cascommunity.org

atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
  --password=changeme   setnsi Hexis Cyber Solutions-snaplock \
  "snaplock:///mnt/snaplock|retentionPeriod=31536000"

atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
  --password=changeme   setnsi Hexis Cyber Solutions-HCAP \
  "archivas:///mnt/HCAP|retentionPeriod=31536000"

atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
  --password=changeme   setnsi HawkEye AP-eternus "eternus://
username=EternusAdmin,password=u$erpa$$wd|policyName=oneyear_sox|TZ=Japan"
```

- Removing a nearline target:

Syntax

```
atmanage <cluster_list> --user=<name> --pass=<pass> deletensi <nsi>
```

IMPORTANT: Running `atquery` against data archived to a deleted NSI generates errors.

Examples

```
atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
  --password=changeme deletensi centera-1

atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
  --password=changeme deletensi Hexis Cyber Solutions-snaplock

atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
  --password=changeme deletensi Hexis Cyber Solutions-HCAP

atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
  --password=changeme deletensi HawkEye AP-eternus
```

- Listing nearline targets:

Syntax

```
atmanage <cluster_list> --user=<name> --pass=<pass> listnsi
```

Example

```
atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
  --password=changeme listnsi
```

Specifying NSAs for Centera Systems

When specifying the NSA for a Centera unit, precede the network storage address with the string:

```
centera://
```

The high-level NSA syntax for a Centera unit is:

```
centera://<connection_specification>[|<retention_specification>]
```

NOTE: In the syntax above, the pipe symbol (|) is a character in the syntax; it is not the meta-syntax character that separates mutually exclusive options.

The network storage address for a Centera unit has two parts:

- *<connection_specification>*—includes parameters legal in Centera for `FPPool_Open`.

  For more information, see "Specifying the Connection Specification", next

- *<retention_specification>*—specifies how long the data should be retained.

  For more information, see "Specifying Data Retention in Centera NSAs", on page 256.

Specifying the Connection Specification

The connection specification includes parameters legal in Centera for `FPPool_Open`. These parameters include:

- Host Identifiers—for more information, see "Specifying Host Identifiers in Centera NSAs", next
- PEA file location—for more information, see "Specifying the PEA File in Centera NSAs", on page 255
- The username/secret for the PAI module to be used by the application—for more information, see "Specifying the Username and Secret for the PAI Module in Centera NSAs", on page 255

The full syntax for the connection specification is one of the following:

```
<IP_Address>[:<port>]|<Host_Name>[:<port>]?<path_to_PEA_file_on_SLS_hosts>
```

or

```
<IP_Address>[:<port>]|<Host_Name>[:<port>]?name=<username>,secret=<password>
```

Specifying Host Identifiers in Centera NSAs

The simplest host identifier is an IP address or host name.

Syntax for specifying the host identifier of a Centera unit:

```
<IP_Address>[:<port>]|<Host_Name>[:<port>]
```

NOTE: In the syntax above, the pipe symbol (|) is the meta-syntax character that separates mutually exclusive options.

Examples

```
centera://centera1.cascommunity.org
```

```
centera://10.2.3.4
```

You can specify multiple Centera host identifiers in a single NSA. Specifying multiple host identifiers allows use, even when one or more Centera systems are unavailable.

Minimal syntax for a Centera NSA:

```
centera://<Host_Identifier>[,<Host_Identifier>[,<Host_Identifier>[...]]]
```

Example

```
centera://10.2.3.4,10.1.7.6
```

Specifying the PEA File in Centera NSAs

In addition to specifying the host identifier, you can specify the location of the Centera Pool Entry Authorization (PEA) file on your EDW hosts. If using authorized access (application profiles) on the Centera unit, you must:

- Copy the PEA file to all hosts of your EDW instance; use the same path structure on each host.
- Specify the path to the PEA file in your NSA definition; precede the path with a question mark (`?`).
- Enclose the entire NSA specification within quotation marks.

Syntax for specifying the location of the Centera PEA File:

```
"centera://<Host_Identifier>?<path_to_PEA_file_on_EDW_hosts>"
```

Examples

```
atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
  --password=changeme setnsi \
  "centera://10.2.3.4,10.006.07.8?/opt/sensage/latest/etc/nss/myProfile.pea"

atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
 --password=changeme setnsi\
 "centera://centera1.cascommunity.org?/opt/sensage/latest/etc/nss/myProfile.pea"
```

Specifying the Username and Secret for the PAI Module in Centera NSAs

To specify username/secret for the PAI module, you must:

- Include name/value pair specifications for the username and password; separate them with a comma.
- Precede the name/value pair specification with a question mark (`?`).
- Enclose the entire NSA specification within quotation marks.

Syntax for specifying the username and secret for a Centera unit:

```
"centera://<Host_Identifier>?name=<username>,secret=<password>"
```

NOTE: EMC may request a user name and password for the PEA file. HawkEye AP does not require a specific user name or password. Use relevant names and passwords for the Centera device.

Examples

```
atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
  --password=changeme \
  setnsi "centera://10.2.3.4?name=s0m3User,secret=s0m3Passwd"

atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
  --password=changeme \
  setnsi "centera://centera1.cascommunity.org?name=s0m3User,secret=s0m3Passwd"
```

NOTE: You also can assign multiple profiles on a connection string to access one or more clusters. For more information on PAI modules and the syntax of connection strings, refer to specifying parameters to `FPPool_Open` in the *Centera Programmer's Guide*.

Specifying Data Retention in Centera NSAs

You can configure Centera NSAs with a retention period. Setting a retention period requires HawkEye AP to retain references to the archived data in the EDW and the data itself on the Centera unit until the retention period passes and you retire the data from the EDW. Only after the retention period has passed can you retire data from the EDW and the NLS device. You must run `atquery` with the `retire` command on the EDW table to remove the archived data from the EDW and the Centera unit. For more information, see "Retiring Data", on page 130.

NOTE: When you retire archived data that is not under retention, HawkEye AP removes the references to the data from the EDW and removes the data itself from the Centera unit.

There are two ways to specify the retention period: `retentionPeriod` and `retentionClass`. Each of these is preceded by a pipe symbol (`|`).

- `retentionPeriod`—basic syntax

  ```
  |retentionPeriod=<num_seconds>
  ```

  Syntax for specifying the retention period for a Centera unit:

  ```
  "centera://<Host_Identifier>|retentionPeriod=<num_seconds>"
  ```

  ### Examples

  ```
  atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
    --password=changeme \
    setnsi "centera://centera1.cascommunity.org|retentionPeriod=31536000"

  atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
    --password=changeme \
    setnsi "centera://10.2.3.4,10.006.07.8|retentionPeriod=31536000"
  ```

  NOTE:

- When you specify a retention period of some specific number of seconds, you ensure that data archived with that NSI is under a fixed length of retention. The data will be available for deletion when that number of seconds has passed.
- After data has been archived with a retention period of a fixed number of seconds, the retention period cannot be changed on the Centera. You can only change the retention period by deleting and re-defining the target. Use `deletensi` to delete the existing one and `setnsi` to define a new one. For more information, see "Defining and Managing Nearline Targets: Overview", on page 252.
- The number of seconds you specify will be a very large number. For example, 86400 seconds represent a single day.

- `retentionClass`—basic syntax

```
|retentionClass=<class_name>
```

Syntax for specifying a retention class for a Centera unit:

```
"centera://<Host_Identifier>]|retentionClass=<class_name>"
```

Examples

```
atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
  --password=changeme \
  setnsi "centera://centera1.cascommunity.org|retentionClass=Hexis Cyber
Solutions"

atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
  --password=changeme \
  setnsi "centera://10.2.3.4,10.006.07.8|retentionClass=Hexis Cyber Solutions"
```

NOTE:

- The value you specify for *<class_name>* is provided by a Centera administrator. The class name specifies the retention period as defined by the administrator.
- When you specify a retention class on the Centera, the archived data is associated with a named retention class on the device. Depending on the model of the Centera (as illustrated in the table below), a retention class can be modified to represent a longer or shorter period of time. The changes are retroactive, so that a change to a retention class affects not only new data to be archived, but existing data archived with that retention class.

  NOTE: You must restart the nss for a change to retention class to become effective.
- The value you specify for *<class_name>* cannot be changed. The Centera administrator can change the underlying retention period that the class defines. If you need to change the class name instead, delete and re-define the target. Use `deletensi` to delete the existing one and `setnsi` to define a new one. For more information, see "Defining and Managing Nearline Targets: Overview", on page 252

IMPORTANT:

- If you specify data retention in a Centera NSA, enclose the entire NSA within quotation marks.
- The value you specify for a Centera NSA depends on the configuration of your system for failover, replication, and compliance mode, and your profile definitions. To determine how to specify the Centera NSA, consult the EMC Centera documentation that describes the parameters for the `FPPool_Open` call used by their API.

- If you require different retention periods for different data sources, create a separate NSA for each desired retention period.

- HawkEye AP recommends that you do not specify a default `retentionClass` on Centera. Doing so could cause undesired retention of test data.

- Certain models of the Centera usually or always define a default retention period. On these models, if you archive data without specifying a retention class or period, the data may or may not already be under some form of retention when you archive it. Therefore, HawkEye AP recommends that you always specify either a retention period or retention class (even if the class specifies no retention period).

  The table below illustrates the various Centera models and how they interpret retention classes and default retention:

Centera Package Installed	Default Retention Period Required?	Default Retention Period	Allowed Changes to Retention Class
Basic			Increase or Decrease
Governance			Increase or Decrease
Compliance Edition +			Only Increase

Configuring a Centera Unit for Single-Instance Storage

The Centera provides multiple ways to generate unique IDs (references) that identify the event-log data archived to it. For optimal storage capacity, Hexis Cyber Solutions recommends that you configure your Centera unit for single-instance storage. Because the EDW saves two copies of all event-log data (primary and secondary), using single-instance storage cuts data storage in half. For more information, see “EDW Architecture”, on page 24.

NOTE: Configuring your HawkEye AP installation such that multiple inserts to the same table occur at the same time reduces the benefit of single-instance storage for data loaded simultaneously to the same table.

Specifying NSAs for SnapLock Systems

To archive HawkEye AP data to a NetApp SnapLock system, perform the following steps:

1 “Change the Ownership of the NFS mountpoint”, next

2 “Create the NSI/NSA Pair”, on page 260

3 “Initialize the Compliance Clock”, on page 262

4 “Configure the EDW for Nearline Storage”, on page 269

Change the Ownership of the NFS mountpoint

Before you define the NSI (Nearline Storage Identifier) and NSA (Nearline Storage Address) for the SnapLock NSS driver, you must:

- Configure read/write access on your HawkEye AP host to the exported NetApp volume.

- Change ownership of the NFS mountpoint to the HawkEye AP user.

NOTE: Before you mount your NFS mountpoint as a hard or soft mount, consider the following:

- **hard mount**—If the connection to the Snaplock system breaks, the NFS server hangs indefinitely while it waits for the IO request to complete. When the NFS server hangs, the archive process also hangs while it waits for the Snaplock system to return online. This situation does not cause other operations to hang but it prevents the archive operation from completing and logging an error.
- **soft mount**—If the connection to the Snaplock system fails, the NFS server fails all IO requests, which causes the EDW to fail the archive operation.

Configuring Read/Write Access to the Exported NetApp Volume

To configure read/write access to the exported NetApp volume

1 If you know the name of the host on which NetApp SnapLock runs, you can use the SnapLock interface to determine the IP address of the SnapLock server. Use the hostname to log into SnapLock; then run the following command from within SnapLock:

```
/ifconfig -a
```

NOTE: The `-a` flag instructs `ifconfig` to display information about all network interfaces in the SnapLock system.

2 Use the returned IP address to open the NetApp FilerView window:

```
http://<filer_IP_address>/na_admin/
```

NOTE: Alternatively, you can use the SnapLock host name to open the NetApp FilerView window:

```
http://<filer_snaplock_hostname>/na_admin/
```

3 Select **NFS > Manage Exports** from the **FilerView** options to display the **Manage NFS Exports** window.

4 Click the name of the desired volume to open the NFS Export Wizard.

5 On the NFS Export Wizard Welcome screen, ensure the following options are selected:

- Read/Write Access
- Root Access
- Security

6 Click **Next** until the **NFS Export Wizard - Read-Write Access** window displays.

7 Enter the IP address of every host in the EDW instance that is mounted to the SnapLock device.

8 Click **Next** until the **NFS Export Wizard - Root Access** window displays.

9 Enter again the IP addresses of all of the hosts in the EDW instance.

10 Click **Next** until the **Commit** button displays; save the configuration and exit from the NFS Export Wizard.

Creating a Single NetApp SnapLock Appliance

You must create a mount directory for the appliance. The mount directory must be a subdirectory of the local mount point specified for the NSS driver and must be named `data0`.

The example below illustrates creation of this required directory and mounting a volume on it:

```
mkdir /mnt/snaplock/data0
mount -t nfs 10.0.1.232:<netapp_snaplock_NFS_export_volume> /mnt/snaplock/data0
```

NOTE: The IP address in the example above represents the SnapLock device.

Creating Multiple NetApp SnapLock Appliances

The EDW can access multiple NetApp SnapLock appliances from a single NSS driver. You must create a mount directory for each appliance. Each mount directory must be:

- A subdirectory of the local mount point specified for the NSS driver
- Named `data`*`N`*, `data`*`NN`*, or `data`*`NNN`*, where `data` is always lowercase and you can specify up to 254 directories; for example, `data0`, `data12`, and `data152`.

IMPORTANT: There must always be one NetApp SnapLock appliance mounted as `data0` on the local mount point specified for the NSS driver.

The example below illustrates creation of two directories (one required) and mounting a volume on them:

```
mkdir /mnt/snaplock/data0
mount -t nfs 10.0.1.232:/<netapp_snaplock_NFS_export_volume> /mnt/snaplock/data0
mkdir /mnt/snaplock/data1
mount -t nfs 10.0.1.212:/<netapp_snaplock_NFS_export_volume> /mnt/snaplock/data1
```

When the EDW archives data to NetApp Snaplock, it automatically writes to the appliance with the most available disk space.

Changing Ownership of the NFS Mountpoint

To change ownership of the NFS mountpoint

1. On the server that runs the HawkEye AP software, switch to root user.
2. Run the Unix `chown` command to change the ownership of the development mountpoint to the HawkEye AP user (typically, `lms`) in the HawkEye AP group (typically, `lms`); for example:

   ```
   chown lms:lms /mnt/snaplock
   ```

Create the NSI/NSA Pair

After you change the ownership of the NFS mountpoint to the HawkEye AP user, a HawkEye AP administrator uses a single `atmanage` command to enable the driver, specify the NSI (Nearline Storage Identifier), and identify the location (NSA) of the mount.

When specifying the NSA for a NetApp SnapLock system, precede the network storage address with the following string:

```
snaplock://
```

The complete NSA syntax for a SnapLock system is:

```
snaplock://<connection_specification>[|retentionPeriod=<num_seconds>]
```

NOTE: In the syntax above, the pipe symbol (|) is a character in the syntax; it is not the meta-syntax character that separates mutually exclusive options.

The network storage address for a SnapLock system has two parts:

- `<connection_specification>`—Specify the local mount point that you will use on all hosts in the EDW instance; this specification is required.
- `retentionPeriod`—Specify how long the data should be retained.

  NOTE:

  - If you do not specify a retention period, the data is deleted immediately when a retire or drop table command is run.
  - The number of seconds you specify will be a very large number. For example, 86400 seconds represent a single day and 31536000 seconds represent a single year.

Example

```
atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
  --password=changeme \
  setnsi Hexis Cyber Solutions-snaplock "snaplock:///mnt/
snaplock|retentionPeriod=31536000"
```

In the example above:

- Hexis Cyber Solutions-`snaplock` is the NSI.
- The NSA includes three forward slashes (/) — two are required by the syntax and one is required to denote the beginning of the path of the local mount point.

Modifying NetApp SnapLock Appliances or Their Mountpoints

After you load the HawkEye AP NSS driver for NetApp Snaplock, the EDW creates a map of connected appliances. The EDW uses the map to archive and query the data. If you add or remove an appliance or its associated mountpoint, you must recreate the map. To do so, you must delete the existing NSI NSA pair and then recreate them. For example:

```
atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
  --password=changeme deletensi Hexis Cyber Solutions-snaplock
atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
--password=changeme \
  setnsi Hexis Cyber Solutions-snaplock "snaplock:///mnt/
snaplock|retentionPeriod=31536000"
```

NOTE: Before you archive data to the SnapLock system, you must set the NetApp compliance clock, as described in “Initialize the Compliance Clock”, on page 262.

NFS Nearline Storage and Local Mount Points f

Generally, you set up one remote host as the nearline target for all EDW hosts when you use a NetApp SnapLock system for nearline storage. From the remote host, use NFS (Network File System) to export a directory to be shared with the EDW hosts. Then, mount the share on each

EDW host using the same mount point, which you specified as the network storage address when you defined the nearline target. You can mount different NFS shares on different EDW hosts, provided you mount them on the same mount point.

NOTE: The directory you export and share through NFS must allow the HawkEye AP system user read/write access. For shared directories and files, do not use the NFS feature that allows owners on the share to be different from those that are visible on the local hosts where the NFS share is mounted. The HawkEye AP system user must be the owner of files created on the NFS share.

IMPORTANT: The HawkEye AP system user is known on remote file systems by the numeric user ID, not by the alphanumeric user name. To ensure the EDW user can archive to the NetApp SnapLock nearline storage, verify that the HawkEye AP system user has the same numeric user ID on every host in the EDW instance.

Initialize the Compliance Clock

Before you archive data to a NetApp appliance, you must initialize the compliance clock on the appliance. Setting the clock enables files to be deleted after the retention period expires.

ComplianceClock™ is NetApp technology that maintains a secure time base to prevent tampering with compliant data. It prevents manipulation of the system clock to prematurely change or remove compliant data.

Because the compliance clock cannot be manually changed after initialization, verify the accuracy of the time on the NetApp appliance before you initialize the compliance clock.

NOTE: HawkEye AP recommends that the administrator synchronize the SnapLock compliance clock with the system clocks on all EDW host(s).

To initialize the compliance clock

Enter the following command on the NetApp appliance:

```
date -c initialize
```

To view the compliance clock

Enter the following command on the NetApp appliance:

```
date -c
```

For more information on managing the NetApp appliance, see the NetApp documentation.

NOTE: The compliance clock resides in the volume header of all volumes on which SnapLock has been licensed and ComplianceClock has been initialized. It is periodically updated by Data ONTAP, the NetApp operating system. Occasionally the compliance clock gets out of synchronization with the system clock. The `date -c` command reports the ComplianceClock value of the volume that is furthest in time from the system clock. Taking a volume offline for an extended period is the most common cause of the time discrepancy. After the volume is brought back online, Data ONTAP starts to make up the time difference, moving ComplianceClock toward the system clock at a rate of one week each year.

Specifying NSAs for HCAP Systems

To archive HawkEye AP data to a Hitachi HCAP system, you must install the HawkEye AP Hitachi HCAP NSS (Nearline Storage Server) driver on a Linux machine. There are several steps to this process:

1. "Change the Ownership of the NFS mountpoint", next
2. "Create the NSI/NSA Pair", on page 264
3. "NFS Nearline Storage and Local Mount Points", on page 265
4. "Enabling Retention", on page 265
5. "Configure the EDW for Nearline Storage", on page 269

CHANGE THE OWNERSHIP OF THE NFS MOUNTPOINT

Before you define the NSI (Nearline Storage Identifier) and NSA (Nearline Storage Address) for the Hitachi HCAP NSS driver, you must:

- Configure read/write access on your HawkEye AP host to the exported Hitachi volume.
- Change ownership of the NFS mountpoint to the HawkEye AP user.

NOTE: Before you mount your NFS mountpoint as a hard or soft mount, consider the following:

- **hard mount**—If the connection to the Hitachi system breaks, the NFS server hangs indefinitely while it waits for the IO request to complete. When the NFS server hangs, the archive process also hangs while it waits for the Hitachi system to return online. This situation does not cause other operations to hang but it prevents the archive operation from completing and logging an error.
- **soft mount**—If the connection to the Hitachi system fails, the NFS server fails all IO requests, which causes the EDW to fail the archive operation.

Configuring Read/Write Access to the Exported Hitachi Volume

For instructions on how to configure read/write access to the exported Hitachi volume, see *Administering HCAP* in the Hitachi Content Archive Platform (HCAP) documentation set.

Creating a Single Hitachi HCAP Appliance

You must create a mount directory for the appliance. The mount directory must be a subdirectory of the local mount point specified for the NSS driver and must be named `data0`.

The example below illustrates creation of this required directory and mounting a volume on it:

```
mkdir /mnt/hcap/data0
mount -t nfs 10.0.1.232:<hitachi_hcap_NFS_export_volume> /mnt/hcap/data0
```

Creating Multiple Hitachi HCAP Appliances

The EDW can access multiple Hitachi HCAP appliances from a single NSS driver. You must create a mount directory for each appliance. Each mount directory must be:

- A subdirectory of the local mount point specified for the NSS driver
- Named `dataN`, where `data` is always lowercase; for example, `data0`, `data1`, and `data2`.

IMPORTANT: There must always be one Hitachi HCAP appliance mounted as `data0` on the local mount point specified for the NSS driver.

The example below illustrates creation of two directories (one required) and mounting a volume on them:

```
mkdir /mnt/hcap/data0
mount -t nfs 10.0.1.232:/<hitachi_hcap_NFS_export_volume> /mnt/hcap/data0
mkdir /mnt/hcap/data1
mount -t nfs 10.0.1.212:/<hitachi_hcap_NFS_export_volume> /mnt/hcap/data1
```

When the EDW archives data to Hitachi HCAP, it automatically writes to the appliance with the most available disk space.

Changing Ownership of the NFS Mountpoint

To change ownership of the NFS mountpoint

1. On the server that runs the HawkEye AP software, switch to root user.
2. Run the Unix `chown` command to change the ownership of the development mountpoint to the HawkEye AP user (typically, `lms`) in the HawkEye AP group (typically, `lms`); for example:

   ```
   chown lms:lms /mnt/hcap
   ```

Create the NSI/NSA Pair

After you change the ownership of the NFS mountpoint to the HawkEye AP user and configure the EDW a HawkEye AP administrator uses a single `atmanage` command to enable the driver, specify the NSI (Nearline Storage Identifier), and identify the location (NSA) of the mount.

When specifying the NSA for a Hitachi HCAP system, precede the network storage address with the following string:

```
archivas://
```

The complete NSA syntax for a Hitachi HCAP system is:

```
archivas://<connection_specification>[|retentionPeriod=<num_seconds>]
```

NOTE: In the syntax above, the pipe symbol (|) is a character in the syntax; it is not the meta-syntax character that separates mutually exclusive options.

The network storage address for a Hitachi HCAP system has two parts:

- `<connection_specification>`—Specify the local mount point that you will use on all hosts in the EDW instance; this specification is required.
- `retentionPeriod`—Specify how long the data should be retained.

  NOTE:

  - If you do not specify a retention period, the data is deleted immediately when a retire or drop table command is run.
  - The number of seconds you specify will be a very large number. For example, 86400 seconds represent a single day and 31536000 seconds represent a single year.

Example

```
atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
  --password=changeme \
  setnsi Hexis Cyber Solutions-hcap "archivas:///mnt/
hcap|retentionPeriod=31536000"
```

In the example above, `Hexis Cyber Solutions-hcap` is the NSI.

Modifying Hitachi HCAP Appliances or Their Mountpoints

After you load the HawkEye AP NSS driver for Hitachi HCAP, the EDW creates a map of connected appliances. The EDW uses the map to archive and query the data. If you add or remove an appliance or its associated mountpoint, you must recreate the map. To do so, you must delete the existing NSI NSA pair and then recreate them. For example:

```
atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
  --password=changeme deletensi Hexis Cyber Solutions-hcap
atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator
  --password=changeme \
  setnsi Hexis Cyber Solutions-hcap "archivas:///mnt/
hcap|retentionPeriod=31536000"
```

NOTE: If you will be setting a retention period, you must enable retention before you archive data to the HCAP system, as described in “Enabling Retention”, on page 265.

NFS Nearline Storage and Local Mount Points

Generally, you set up one remote host as the nearline target for all EDW hosts when you use a Hitachi HCAP system for nearline storage. From the remote host, use NFS (Network File System) to export a directory to be shared with the EDW hosts. Then, mount the share on each EDW host using the same mount point, which you specified as the network storage address when you defined the nearline target. You can mount different NFS shares on different EDW hosts, provided you mount them on the same mount point.

NOTE: The directory you export and share through NFS must allow the HawkEye AP system user read/write access. For shared directories and files, do not use the NFS feature that allows owners on the share to be different from those that are visible on the local hosts where the NFS share is mounted. The HawkEye AP system user must be the owner of files created on the NFS share.

IMPORTANT: The HawkEye AP system user is known on remote file systems by the numeric user ID, not by the alphanumeric user name. To ensure the EDW user can archive to the Hitachi HCAP nearline storage, verify that the HawkEye AP system user has the same numeric user ID on every host in the EDW instance.

Enabling Retention

For HawkEye AP to set retention on an HCAP device, you must enable **atime** synchronization.

To enable atime retention

1 Use a web browser to log into your HCAP system.

2 Select **Policy** > **Retention**.

3 In the Policy Settings window, ensure the **Synchronize POSIX atime values and object retention settings** checkbox has been selected.

4 Click **Submit**.

For more information on retaining data on an HCAP system, see *Administering HCAP* in the Hitachi Content Archive Platform (HCAP) documentation set.

Specifying NSAs for Eternus Systems

The Fujitsu Eternus NLS device supports:

- Write Once Read Many (WORM) data access, which enables clients read access to data while it prevents write access
- Ability to transparently move data on the device from disk to tape, where it can remain accessible to EDW queries

  NOTE: The retrieval time of data stored on tape is significantly slower than when the data is fetched from disk.
- Policy files defined on the Fujitsu device that enable data retention

When you specify the NSA for an Eternus unit, precede the network storage address with the string:

```
eternus://
```

The complete NSA syntax for an Eternus unit is:

```
"eternus://<connection_specification>|<retention_policy>|<time_zone>"
```

The network storage address for an Eternus unit has three parts:

- `<connection_specification>`—specifies the login credentials
- `<retention_policy>`—specifies the name of the policy that determines the retention policy for the NSA/NSI pair
- `<time_zone>`—specifies the Eternus time zone

The next three topics build on each other to document configuration of an Eternus NSA:

- "Specifying the Connection in Eternus NSAs", next
- "Specifying the Connection and the Retention Policy in Eternus NSAs", on page 267
- "Specifying the Connection, the Retention Policy and the Time Zone in Eternus NSAs", on page 267

Specifying the Connection in Eternus NSAs

The syntax for specifying the connection of an Eternus unit includes a username and password for the Eternus. The syntax is:

```
username=<user_name>,password=<password>
```

Connection Example

```
eternus://username=EternusAdmin,password=u$erpa$$wdf
```

NOTE:

- The text you enter when you define the NSA is distributed in plain text to every host in your EDW instance.
- The IP address of the Fujitsu Eternus device is configured outside of HawkEye AP. For more information, refer to the Fujitsu manual: *Content Archive Manager Installation and Operation Manual*.

Specifying the Connection and the Retention Policy in Eternus NSAs

You must specify a retention policy when you configure an Eternus NSA. Setting a retention policy forces HawkEye AP to retain references to the archived data in the EDW and retain the data itself on the Eternus unit until the retention period passes and you retire the data from the EDW.

The syntax for specifying a retention policy is:

```
policyName=<policy_name>
```

NOTE: You must specify a retention policy whether or not you want to place the archived data under retention.

- If you do not want the data placed under retention, specify a policy that sets `0` (zero) days and `0` (zero) years.
- If you do want the data placed under retention, specify a policy that includes retention settings. The granularity of the retention period is until midnight of a specific day and year.

When you successfully retire the data from the EDW, the data is also removed from the Eternus unit. You must run `atretire` on the EDW table to remove the archived data from the Eternus unit. For more information, see “Retiring Data”, on page 130.

NOTE: When you retire archived data that is not under retention, HawkEye AP removes the references to the data from the EDW and removes the data itself from the Eternus unit.

IMPORTANT:

- When you define an Eternus NSA, enclose the entire NSA within quotation marks.
- If you require different retention periods for different data sources, create a separate NSA for each desired retention period.

  NOTE: Changing a policy date does not affect data already archived.

For information on specifying a retention policy, refer to the Fujitsu manual: *Content Archive Manager Installation and Operation Manual*.

Example of Specifying Retention in the NSA

```
atmanage "edw01:8072,edq02:8072,edw03:8072" --user=administrator
  --password=changeme setnsi HawkEye AP-eternus "eternus://
username=EternusAdmin,password=u$erpa$$wd|policyName=oneyear_sox"
```

Specifying the Connection, the Retention Policy and the Time Zone in Eternus NSAs

When you configure an Eternus NSA, you must specify the same time zone that has been internally set for the Eternus.

The syntax for the time_zone is:

```
TZ=<time_zone>
```

The allowed values for `<time_zone>` are listed in Appendix C: Time Zones.

Examples of Setting the Time Zone in the NSA

NOTE: The following examples illustrate the full syntax for specifying an Eternus NSA: the connection, the retention policy, and the time zone.

- **Japan**

```
atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
  --password=changeme setnsi SenSage-eternus "eternus://
username=EternusAdmin,password=u$erpa$$wd|policyName=oneyear_sox|TZ=Japan"
```

- **Pacific Standard Time**

```
atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
  --password=changeme setnsi SenSage-eternus "eternus://
username=EternusAdmin,password=u$erpa$$wd|policyName=oneyear_sox|TZ=PST8PDT"
```

Specifying NSAs for Remote File Systems

When you specify the NSA for a remote file system, precede the network storage address with:

```
directory://
```

The complete NSA syntax for a remote file system is:

```
directory://<local-NFS-mount-point>
```

Substitute *<local-NFS-mount-point>* with the local mount point that you will use on all hosts in the EDW instance.

Example

```
directory:///nss
```

Generally, you set up one remote host as the nearline target for all EDW hosts when you use remote file systems for nearline storage. For information on how to use the NFS (Network File System) to export a directory to be shared from the remote host to the EDW hosts, see “NFS Nearline Storage and Local Mount Points f”, on page 261.

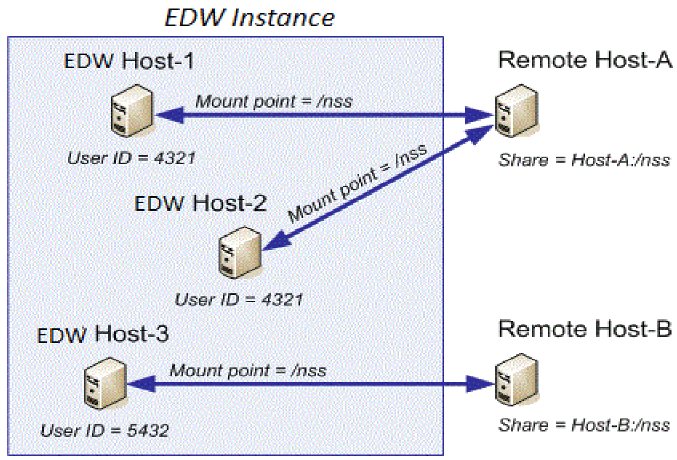
NOTE: Before you mount your NFS mountpoint as a hard or soft mount, consider the following:

- **hard mount**—If the connection to the remote file system breaks, the NFS server hangs indefinitely while it waits for the IO request to complete. When the NFS server hangs, the archive process also hangs while it waits for the remote file system to return online. This situation does not cause other operations to hang but it prevents the archive operation from completing and logging an error.
- **soft mount**—If the connection to the remote file system fails, the NFS server fails all IO requests, which causes the EDW to fail the archive operation.

CONFIGURING NFS NEARLINE STORAGE FOR HAWKEYE AP SYSTEM USERS WITH DIFFERENT IDS

The HawkEye AP system user is known on remote file systems by the numeric user ID, not by the alphanumeric user name. This can cause a problem for nearline storage if the numeric user IDs for the HawkEye AP system user are different on different EDW hosts. In this situation, you must create a different NFS share for each of the user IDs assigned to the HawkEye AP system user on the EDW hosts.

For example, assume you have a 3-host EDW instance. The HawkEye AP system user has "4321" as the user ID on two of the hosts; the user ID is "5432" on the other host.



All three hosts use the same local mount point, "/nss". The two hosts where the user ID is "4321" mount the share "Host-A:/nss"; the other host where the user ID is "5432" mounts the share "Host-B:/nss". Alternate shares do not need to be on separate hosts; you can export and share different directories from the same remote host.

As an alternative, you can change the user IDs for the HawkEye AP system user to be the same on all EDW hosts. For more information, see The HawkEye AP System User in Chapter 2, “Configuring HawkEye AP” in the *Installation, Configuration, and Upgrade Guide*.

Configure the EDW for Nearline Storage

Typically, there is no need to modify the default configuration settings for nearline storage. If your site does need to modify these, you must do so after you configure your EDW instance. To modify the configuration of nearline storage, run `clsetup configure sls` with the required parameter(s) and values. It is recommends that you talk to Hexis Cyber Solutions Technical Support before you configure these parameters.

Use the parameters documented below to configure the EDW to archive to the nearline storage devices you defined with the `atmanage` command. For more information on `clsetup`, see “Configuring or Reconfiguring an EDW Instance (clsetup)”, on page 37.

Parameters that Apply to All Devices

The most important nearline-storage parameters specify cache location, cache size, and thread pool size. These parameters, which apply to all nearline storage devices, are:

- **cache location**—The value of this parameter defaults to: `<Sensage_Home>/latest/data/nss/cache`. The parameter is:

  ```
  --nss-cachedir=<nss_directory>
  ```

- **cache size**—The value of this parameter defaults to `10` GB. Keep the default size unless you get the following error: `Cannot free cache space, cache is too small for SLS load`. If this error occurs, double the cache size. If you continue getting an error, keep doubling the size until the error no longer displays. The parameter is:

  ```
  --nss-cachesize=<size>
  ```

- **thread pool size**—This value depends on the size of the nearline storage appliance. The number of threads represent the number of concurrent actions performed on the nearline storage appliance at any one time; this value defaults to `12`. The parameter is:

  ```
  --nss-threadpoolsize=<size>
  ```

- **Disk read buffer size**—This value specifies the size of the disk read buffer. The value of this parameter defaults to `1000000`. The parameter is:

  ```
  --nss-diskreadbuffer=<size>
  ```

- **Cache depth**—This value specifies the number of directories below the parent cache directory. Increasing this number can improve performance by limiting the number of files stored in each subdirectory. The value of this parameter defaults to `2`, meaning that any cache entry in the NSS resides two subdirectories down from the parent cache directory. The maximum value is 4.

  NOTE: If you set `nss-cachedepth` to 4, the NSS device can take over an hour to start.

  ```
  --nss-cachedepth=<depth>
  ```

IMPORTANT: Contact your Hexis Cyber Solutions representative for assistance in determining the appropriate settings for these parameters. Typically, there is no need to change the default values.

Parameter that Applies to Specific Devices

In addition to the parameters that apply to all nearline storage devices, there is a parameter that is specific to each nearline storage device. This parameter enables you to change the blobsize of the specified nearline storage device.

The default value of this parameter depends on the specific nearline storage device. Typically there is no need to change the default value of this parameter.

Available options for this parameter are:

```
--nss-centera-blobsize
--nss-snaplock-blobsize
--nss-hcap-blobsize
--nss-eternus-blobsize
--nss-directory-blobsize
```

Syntax Examples

The full syntax for modifying nearline storage from a command line are shown below.

- EMC® Centera™ Nearline Storage Driver

```
clsetup configure sls --nss-cachedir=<nss_directory> --nss-cachesize=<size> \
  --nss-threadpoolsize=<size>   --nss-diskreadbuffer=<size> \
  --nss-centera-blobsize=<size>
```

- NetApp® SnapLock™ Nearline Storage Driver

```
clsetup configure sls --nss-cachedir=<nss_directory> --nss-cachesize=<size> \
  --nss-threadpoolsize=<size> --nss-diskreadbuffer=<size> \
  --nss-snaplock-blobsize=<size>
```

- Hitachi HCAP® Nearline Storage Driver

```
clsetup configure sls --nss-cachedir=<nss_directory> --nss-cachesize=<size> \
  --nss-threadpoolsize=<size> --nss-diskreadbuffer=<size> \
  --nss-hcap-blobsize=<size>
```

- Fujitsu Eternus™ Nearline Storage Driver

```
clsetup configure sls --nss-cachedir=<nss_directory> --nss-cachesize=<size> \
  --nss-threadpoolsize=<size>   --nss-diskreadbuffer=<size> \
  --nss-eternus-blobsize=<size>
```

- Remote File Systems

```
clsetup configure sls --nss-cachedir=<nss_directory> --nss-cachesize=<size> \
  --nss-threadpoolsize=<size> --nss-diskreadbuffer=<size> \
  --nss-directory-blobsize=<size>
```

Archiving of Local Data to Nearline Storage

You can archive the local data in a table to nearline storage with the `archivetbl` subcommand of the `atmanage` utility. You can archive the entire table, or you can limit the archiving to a specific time frame. You will need to run this command every time you want to archive data. You may want to write a script that calculates the maximum and minimum timestamp parameters required by the command, and then schedule that script to run on a regular schedule using a scheduling utility such as the Linux `cron` utility.

Syntax

```
atmanage <cluster_list> --user=<name> --password=<pass> \
--namespace=<namespace> archivetbl <table_name> <NSI> \
[<max_timestamp> [<min_timestamp>]]
```

where:

- No time stamps—specifies archiving the entire table.
- One time stamp—specifies archiving data that is older than the time stamp.

- Two time stamps—specifies a time frame; the first time stamp represents the latest or maximum timestamp, followed by the earliest or minimum time stamp.
- Timestamps should be in ISO8601 date format (`YYYY-MM-DDTHH:MM:SS`)

NOTE: Some data older than the cutoff timestamp may not be archived immediately, due to the how the EDW stores data. A record of the archived data displays after the archive command has completed.

TIP: To determine the names of tables to archive, run the following command to list tables in the EDW:

```
atview <host:port> tables --namespace=<namespace> --user=<user> --
pass=<password>
```

For more information, see Listing Tables in Chapter 4, “Administering an EDW Instance” in the Administration Guide.

Examples

The following examples pertain to an EMC Centera device, but pertain to all HawkEye AP nearline storage devices.

The following command archives the `"Firewall"` table to the NSI identified by `"centera-1"` up to January 1, 2007, midnight.

```
atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
  --password=changeme --namespace=documentation \
  archivetbl Firewall centera-1 2007-01-01T00:00:00
```

The following command archives the `"Firewall"` table to the NSI identified by `"centera1"` between January 1, 2006, midnight and January 1, 2007, midnight.

```
atmanage "edw01:8072,edw02:8072,edw03:8072" --user=administrator \
  --password=changeme --namespace=documentation \
  archivetbl Firewall centera-1 2007-01-01T00:00:00 2006-01-01T00:00:00
```

It is possible that not all data will be archived immediately. HawkEye AP moves data to nearline storage only when all data in the same directory meets the age requirements.

CHAPTER 10

Administering Assets and Monitoring Alerts

This chapter contains the following sections:

Overview

HawkEye AP *alerts* provide timely visibility into events that may require immediate attention or further investigation. The HawkEye AP system raises alerts in response to pre-defined conditions, which are either user-defined or are HawkEye AP-defined. You can use HawkEye AP Console to view alerts in alert widgets, view the events that contributed to raising the alert, and define email notifications to be sent when an alert is raised.

HawkEye AP categorizes alerts as follows:

- **Security Alerts** are raised in response to activity in monitored systems. For example, a security alert can be raised when a particular user logs in, when certain servers are accessed, or when specified patterns of activity are detected. Security alerts are a feature of the HawkEye AP Real-Time system and require *parsing rules* to parse the incoming streams of event data and *alerting rules* to evaluate incoming event data and trigger alerts.

  Some alerting rules trigger alerts based on criteria in the rule. Other alerting rules provide a window in which you can specify values for a set of criteria. Rules that enable you to set your own criteria values are called *configurable alerting rules*. You use HawkEye AP Console to define configurable alerting rules as well as to view security alerts in the *Security Alerts widget*. You can also use HawkEye AP Console to investigate alerts by running reports against values that appear in the alert.

  For example, if a security alert indicates an unauthorized access by a particular user, you can search other event data stored in the Event Data Warehouse (EDW) for events that involve that user. You can also examine the sequence of events that caused the alert condition.

  For information on using the Security Alert Widget, see Viewing Security Alerts in Chapter 2, "Using Dashboards" of the *Reporting Guide*. For information on configuring alerting rules to raise alerts, see Chapter 8: Creating Alerting Rules from Templates, also in the *Reporting Guide*.

- **Exception Report Alerts** are raised when a scheduled HawkEye AP report raises an exception report alert. The alerts are raised when a designated report returns one or more rows. For example, a report that lists after-hours logins could raise an alert when a user logs in after hours. You view Exception Report Alerts using the *Exception Alerts widget* in HawkEye AP Console . For more information, see "Monitoring Exception Alerts", on page 293.

- **System Alerts** are raised in response to failures within the HawkEye AP system itself. For example, an alert is raised when data fails to load as expected into the EDW. You use HawkEye AP Console to view system alerts in the *System Alerts widget*. For more information, see "Monitoring System Alerts Using HawkEye AP Console", on page 289.

NOTE: HawkEye AP Console displays alerts for eight days or until 351 alerts have been raised, whichever occurs first.

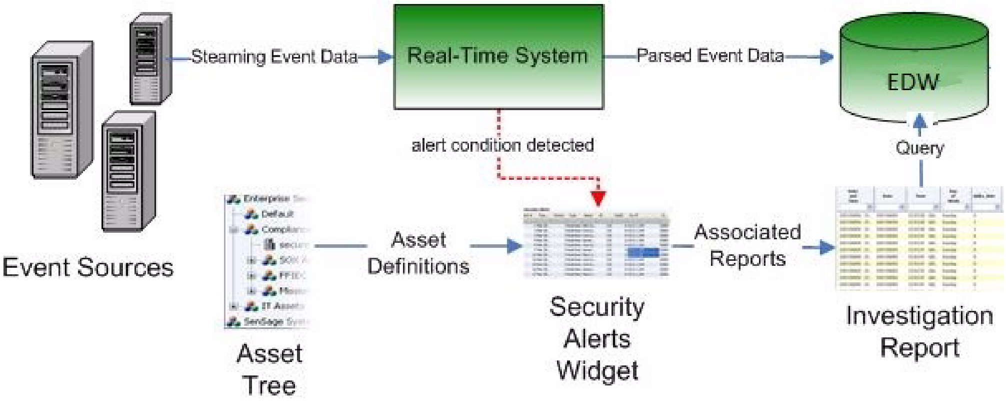
About the HawkEye AP Alert System

Each category of alert flows through your HawkEye AP deployment in different ways, discussed next.

Security Alert Flow

As shown in Figure 10-1, the HawkEye AP Real-Time system processes streaming event data. It parses the raw event data into normalized event data and uses alerting rules to detect alert conditions. It also stores parsed data in the EDW. When it detects an alert condition, it displays an alert in the Security Alerts widget, categorized by the available assets and asset groups. You can investigate the alert in the Security Alerts widget by running a report that uses criteria from the event data that raised the alert. You can either run a report specifically associated with the alert or run any report on your system to which you have access. If email notifications are configured in the asset, the Real-Time system sends an email about the alert to the users on the notification list.

Figure 10-1: Security Alert Flow

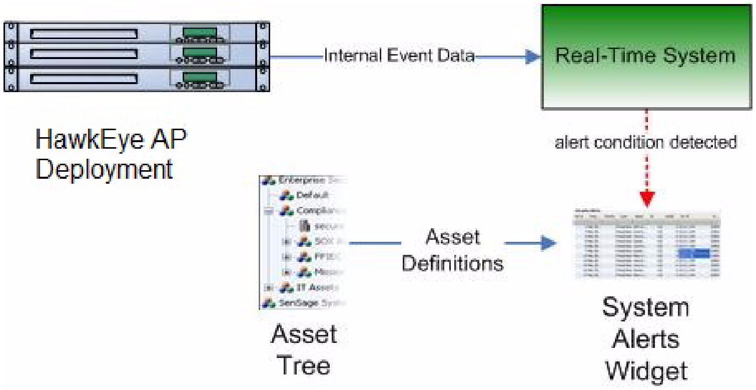


For more information, see Chapter 8: Creating Alerting Rules from Templates in the *Reporting Guide.*

System Alert Flow

System alerts are generated internally by a running HawkEye AP deployment. No configuration is required to generate system alerts. As shown in Figure 10-2, the HawkEye AP Real-Time system processes internal event data; when it detects alert conditions, it displays an alert in the System Alerts widget.

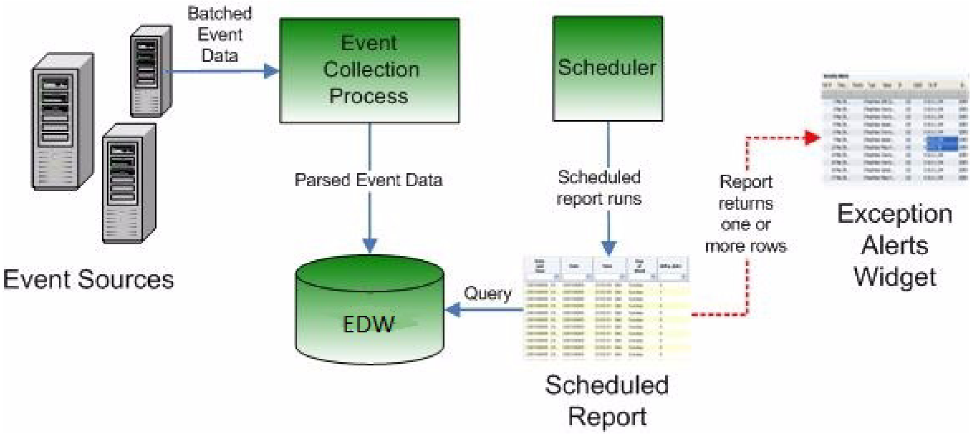
Figure 10-2: System Alert Flow



Exception Report Alert Flow

Exception Report Alerts are created when a scheduled report run returns one or more rows. The schedule for the report must be configured to send “Exception Report Alerts” when one or more rows appear in the report. Each row in the Exception Alerts widget displays the name of the report and the number of rows returned. You can right-click on the row to quickly view the report.

Figure 10-3: Exception Report Alert Flow



For more information, see Output Tab: Specifying Destination in Chapter 7, “Creating and Editing Schedules” in the *Reporting Guide*.

Creating Asset Groups and Assets to Organize Security Alerts

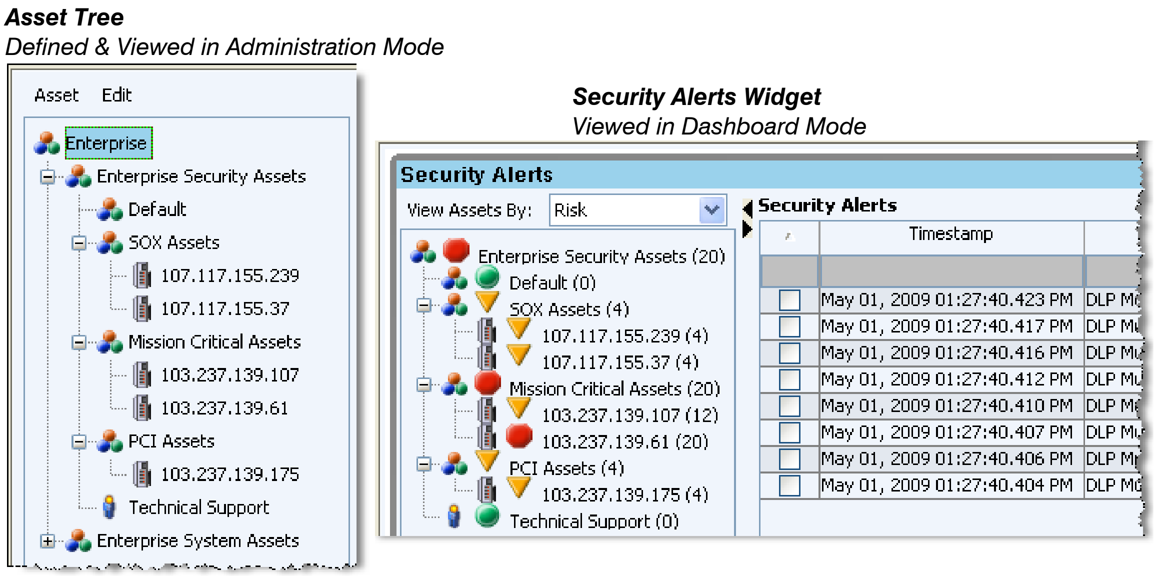
By default, all security alerts display in a single table in the Security Alerts widget. To make alert viewing more meaningful and precise, Hexis Cyber Solutions recommends that you define an IP Range asset or User List asset for each server and user that you particularly want to track. To make alert viewing even more meaningful, you can organize your IP Range assets and User List assets into Asset Groups. Whenever an alert is raised against one of your defined assets, the alert displays below the specific asset group to which the asset belongs.

The table below illustrates the three types of Security assets that display in the asset tree and in the Security Alerts widget, as well their associated icons.

Asset Icon	Meaning
	Asset Group—expand the group to display the assets and additional groups it contains.
	IP Range—identified by a single IP address or a range of IP addresses
	User List—identified by a user name

Figure 10-4 illustrates the relationship between user-defined assets and asset groups, which are defined in HawkEye AP Console Administration mode, and their display in the Security Alerts widget in Dashboards mode.

Figure 10-4: An Asset Tree in the Assets Module and its Corresponding Alerts Widget



Asset Tree
Defined & Viewed in Administration Mode

Security Alerts Widget
Viewed in Dashboard Mode

	Timestamp	
☐	May 01, 2009 01:27:40.423 PM	DLP Mo
☐	May 01, 2009 01:27:40.417 PM	DLP Mu
☐	May 01, 2009 01:27:40.416 PM	DLP Mu
☐	May 01, 2009 01:27:40.412 PM	DLP Mu
☐	May 01, 2009 01:27:40.410 PM	DLP Mi
☐	May 01, 2009 01:27:40.407 PM	DLP Mu
☐	May 01, 2009 01:27:40.406 PM	DLP Mi
☐	May 01, 2009 01:27:40.404 PM	DLP Mo

When you organize your users and servers into asset groups, and an alert is raised against one of a group's children, the asset group displays a status icon that represents the highest level alert in the subordinate assets of the group. For example, Figure 10-4 illustrates the **Mission Critical Assets** group with two child assets. One of the children displays the high-priority red octagon and the other displays the lower-priority yellow triangle. Even when the parent, Mission Critical Assets, is collapsed, it displays a red octagon, which allows you to see at a glance that there are severe alerts in the group. For more information about the meaning of Security-alert icons, see Understanding and Using Threat and Risk Views in Chapter 2, "Using Dashboards" in the *Reporting Guide*.

When you click on a node in the asset tree, only alerts for the selected node display in the alerts widget table. At the highest node of the asset tree, all alerts display.

If you have defined assets and asset groups for your important assets but an alert is raised against a server or user that has not been associated with a user-defined asset, that alert displays in the **Default** asset group along with all other undefined assets. You do not need to define the

Default asset group because HawkEye AP provides it. This group displays above all user-defined asset groups. For more information, see “Fixed and User-Defined Asset Groups”, on page 278.

When you organize your assets, you can easily identify the machines or persons associated with a security alert. For instance, you might define an asset that corresponds to machines located in a remote office by defining the range of IP addresses assigned to machines in that location. If an alert is raised for an event that uses an IP address in that range, HawkEye AP emails an alert notification to the email addresses defined for the asset and adds a row containing details of the event to the Security Alerts widget.

Similarly, when you define assets based on user names and an alert is raised for an event involving one of these users, HawkEye AP sends notifications to the email address defined for that user and adds a row containing details of the event to the Security Alerts widget. Note that the notification email is not sent to the user represented by the user asset but to email recipients defined in that asset.

You define these assets using the HawkEye AP Console *Assets Module*, discussed in “Using HawkEye AP Console to Manage Assets and Asset Groups”, next.

Using HawkEye AP Console to Manage Assets and Asset Groups

You use HawkEye AP Console in Administration mode to manage and view Assets. The system and security assets that you maintain in the Assets module also display in the corresponding System and Security Alerts widgets, along with the details of events that raise an alert.

For convenience, you can organize security assets into one or more *asset groups* that match your organizational requirements. For example you can create asset groups for divisions of your company, geographical regions, or other criteria. You can nest asset groups within other asset groups as needed. For each asset, you specify properties that define its *Name*, its *Description*, a *Value* that determines the relative importance of the asset, CIDR Blocks (IP address ranges) or a list of *user IDs*, and *Notifications*, a list of email addresses to which alerts are sent.

You use the Assets module of HawkEye AP Console to:

- create new security assets and asset groups and set their properties
- view HawkEye AP system asset groups and modify their notification recipients

To access the Assets Module

1 Open HawkEye AP Console.

See Chapter 1, “Getting Started”, on page 25 in the *Reporting Guide*.

2 Click **Administration** from the Toolbar.

HawkEye AP Console changes to Administration Mode.

3 Click **Assets** in the Navigator.

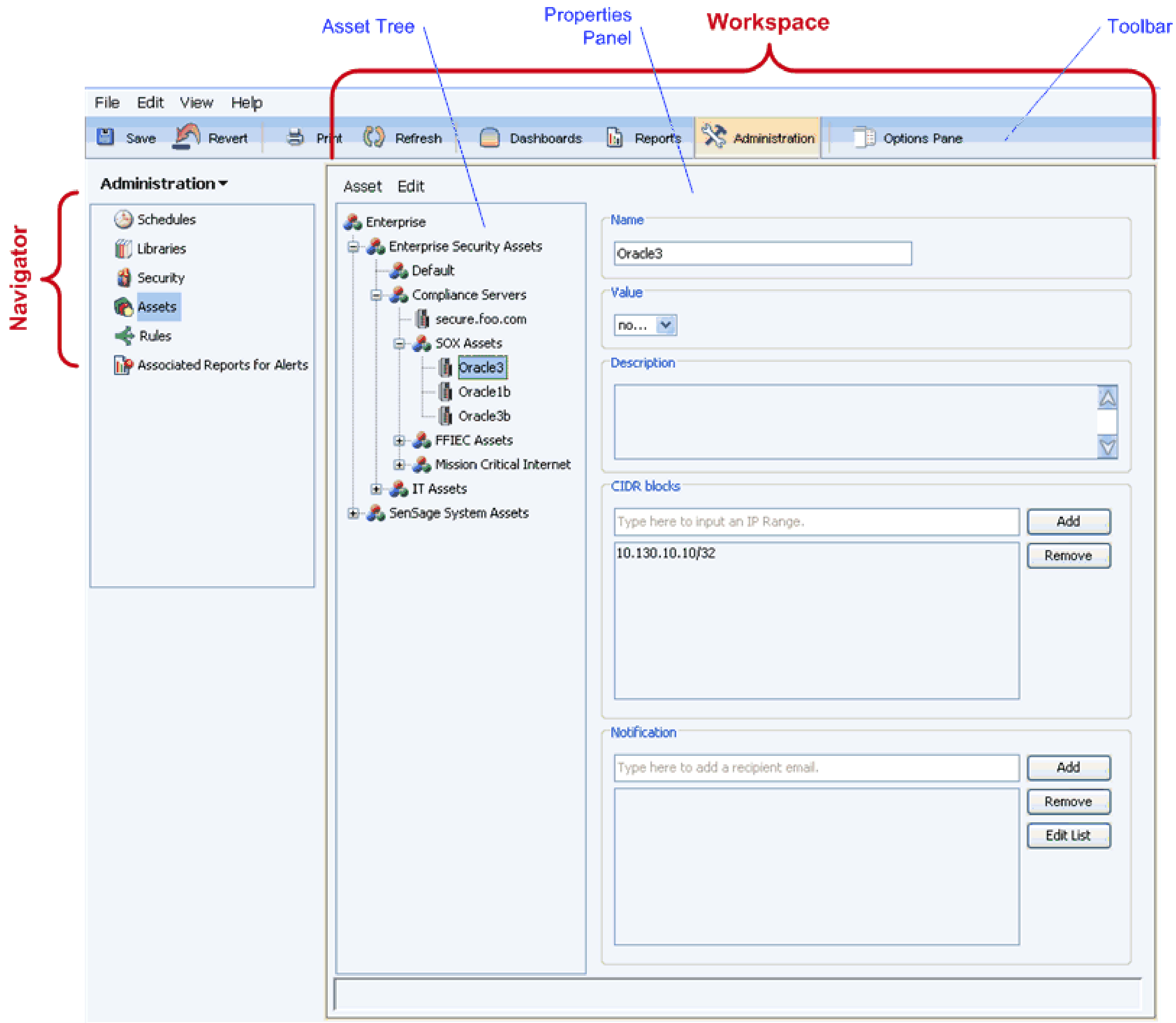
The Assets module opens in the Workspace area.

The Assets module has two window panes:

- On the left, the **Asset Tree** displays enterprise security assets and HawkEye AP system assets.
- On the right, the **Properties Panel** displays information about the assets currently select in the Asset Tree and enables you to edit the information displayed.

Figure 10-5 illustrates the Assets Module.

Figure 10-5: Assets Module



Name: Oracle3

Value: no...

CIDR blocks: 10.130.10.10/32

Fixed and User-Defined Asset Groups

Some of the asset groups in the tree are *fixed* and cannot be changed. A fixed asset group cannot be deleted from the tree or moved from one group to another. In the example above, the asset group labeled Enterprise Security Assets is fixed; you cannot delete nor move it. However, you can change its properties, such as its name and value.

In addition to being fixed, some asset groups are *read-only*. A read-only asset group does not permit its properties to be changed, other than its list of notification recipients. In the example above, the asset group labeled **Default** is fixed and read-only; you cannot change its name and you cannot move it.

Asset groups that are not fixed are *user-defined* asset groups. You can add user-defined asset groups anywhere within the Enterprise Security Assets group. You can modify user-defined groups, move them, and delete them. In the example above, the asset group labeled SOX Assets is a user-defined group.

Individual assets are always user-defined assets. In the example above, the asset labeled Adam's Server is a user-defined asset.

Figure 10-6 below illustrates how various types of assets and asset groups display in HawkEye AP Console .

Shared Assets

User-defined groups and individual assets can belong to multiple groups. The definition of an asset that belongs to multiple groups exists only once in the system. The definition is shared among the groups to which the asset belongs. After you modify the properties of a shared asset that one group displays, you can see the changes through the other groups that share the asset.

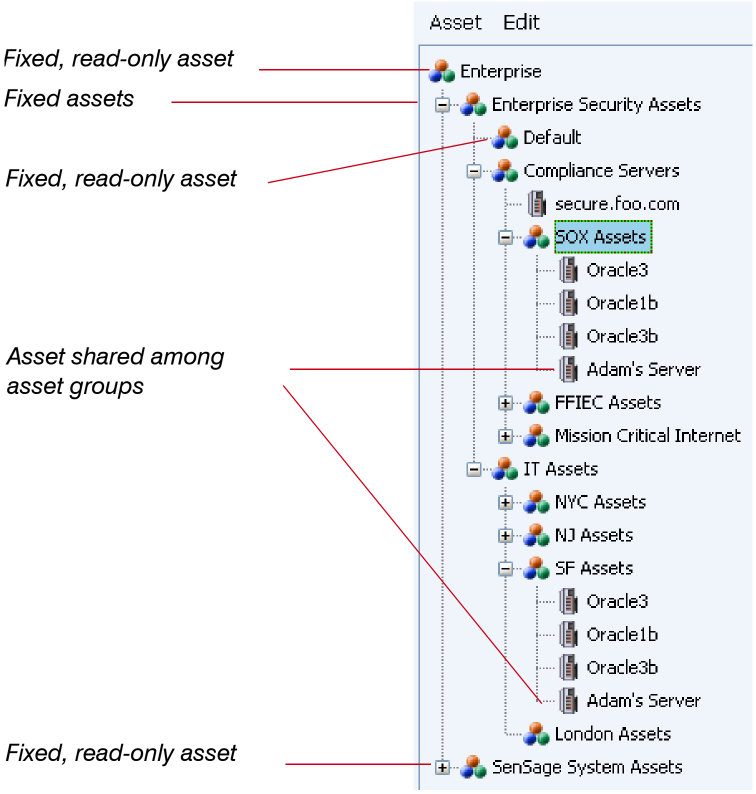
In Figure 10-6, Adam's Server displays below two different groups: SOX Assets and SF Assets. However, there is only one definition of Adam's Server. Selecting the Adam's Server that belongs to SF Assets displays the same property values as those that are displayed for the Adam’s Server that belongs to SOX Assets. After you change the properties for Adam's Server in one group, the changes are reflected by the Adam's Server in the other group.

Share individual assets among groups to categorize your security assets in different ways. For example, you may want to categorize assets by the way your enterprise uses them. In the example above, the asset group Compliance Servers represents a usage categorization. Adam's Server is in the SOX Assets subcategory because your enterprise uses his server for Sarbanes-Oxley compliance. At the same time, you may want to categorize assets by their geographic locations. Adam's Server belongs to the SF Assets subcategory of IT Assets, which categorizes assets geographically.

For more complete information on how the HawkEye AP Console manages shared assets, see “Sharing Security Assets”, on page 285.

Figure 10-6 illustrates how various types of assets and asset groups appear in HawkEye AP Console .

Figure 10-6: Asset Tree



Asset Properties

The properties that display in the Properties Panel correspond to the type of asset selected in the Asset Tree. The Enterprise Security Assets tree contains three kinds of assets: IP Range assets, User List assets, or Group assets.

An IP Range asset has the following properties:

- **Name**—should be unique to the asset instance. Although the name "Adam's Server" is unique to the asset instance in the example above, the Asset Tree displays it twice because it is shared among two groups and not because the name has been used twice in separate, individual assets.

  IMPORTANT: The Asset Manager allows you to enter duplicate asset names and it does not check to see if IP ranges or user names are duplicated among the assets. Plan the organization of your Asset Tree carefully to avoid duplication and confusion.

- **Value**—one of five options (**none**, **low**, **normal**, **high**, and **max**) that specify the relative importance of the asset to your enterprise. The Alert Monitor uses this value to compute the degree of risk when an alert is against the asset: it multiplies the asset value by the priority of alerts raised against the asset. Value defaults to `none`.

For more information, see Understanding and Using Threat and Risk Views in Chapter 2, “Using Dashboards” in the *Reporting Guide*.

- **Description**—an optional text field that provides meaningful information for colleagues who also manage assets.
- **CIDR blocks**—an IP address or range of IP addresses that identify the asset. For more information, see the following link: http://searchnetworking.techtarget.com/sDefinition/0,,sid7_gci213850,00.html.
- **Notification**—email addresses of those who should be notified when alerts are raised against the asset.

Properties of a User List asset differ from those of an IP Range asset, but only in that a User List contains a list of valid users rather than a list of CIDR blocks. Properties of an Asset Group differ from those of an IP Range or User List only in that an Asset Group has neither a list of valid users nor CIDR blocks. The Real-Time component uses IP Ranges and users to associate alerts with assets.

Creating Security Assets

Before you create a security asset, determine:

- which group or groups you want the asset to belong to
- the type of asset you want to create: IP Range, User List, or Group
- what property values to specify, such as the value your enterprise assigns to the asset and notification recipients who receive email when alerts are raised against the asset

This section describes the following topics:

- “Creating a New IP Range Asset”, next
- “Creating a User List Asset”, on page 284

Creating a New IP Range Asset

The example in the following procedure creates an IP Range asset, Julie's Server, that belongs to two groups: SOX Assets (a usage categorization) and London Assets (a geographic categorization).

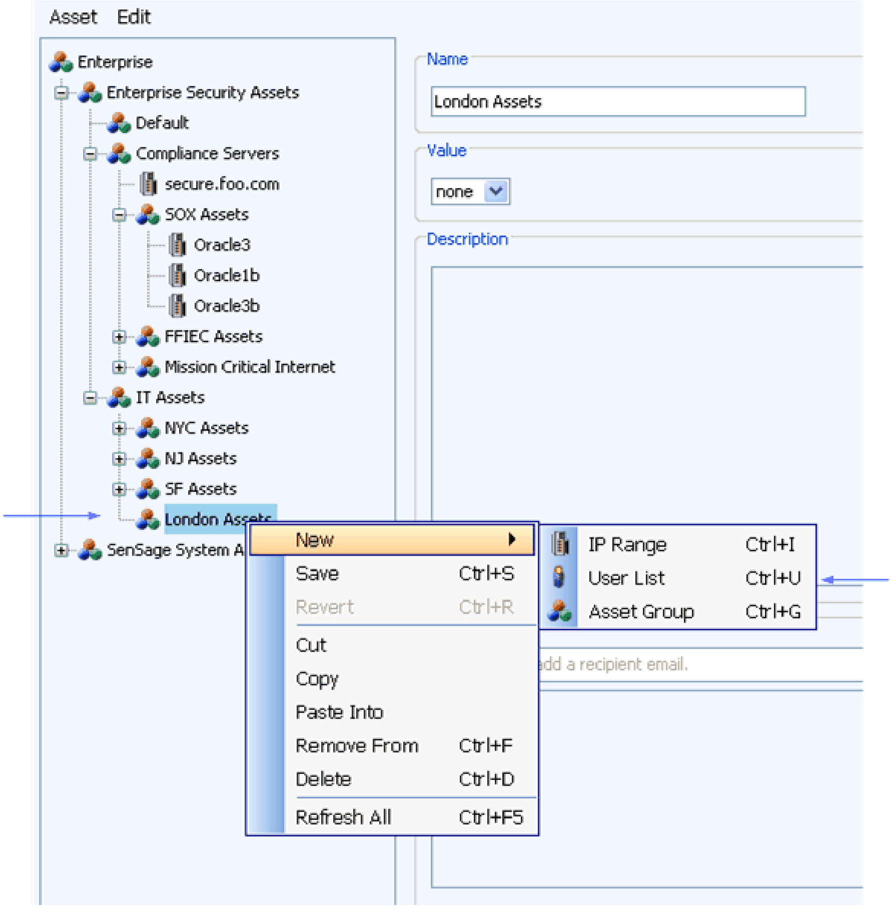
To create a new IP Range asset

1 Locate the group where you want the asset displayed and right-click below that parent in the Asset tree.

NOTE: You cannot create assets in the fixed, read-only Default group.

2 From the popup menu, select **New** and then **IP Range**, as shown in Figure 10-7.

Figure 10-7: Specifying Asset Type



Name: London Assets

Value: none

Description

Right-click the group where you want your new asset displayed

Select the type of asset you want to create from the **New** popup menu

The tree displays the new asset and selects it; the HawkEye AP Console Properties Panel on the right displays its initial values.

3 In the **Name** field, enter the name of your new asset.

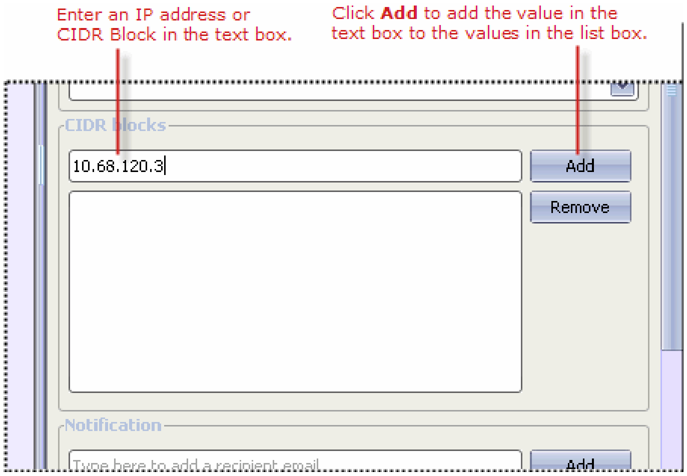
4 In the **Value** dropdown, select the value that best represents its importance to your enterprise.

IMPORTANT: To avoid problems with the Risk view in the Alerts module, do not leave the value of assets set to **none**.

5 Optionally, in the **Description** field, enter meaningful information for colleagues who also manage assets.

6 In the **CIDR blocks** group box, type a CIDR block or individual IP address in the text box and click **Add**, as shown in Figure 10-8.

Figure 10-8: Adding an IP Address or CIDR Block

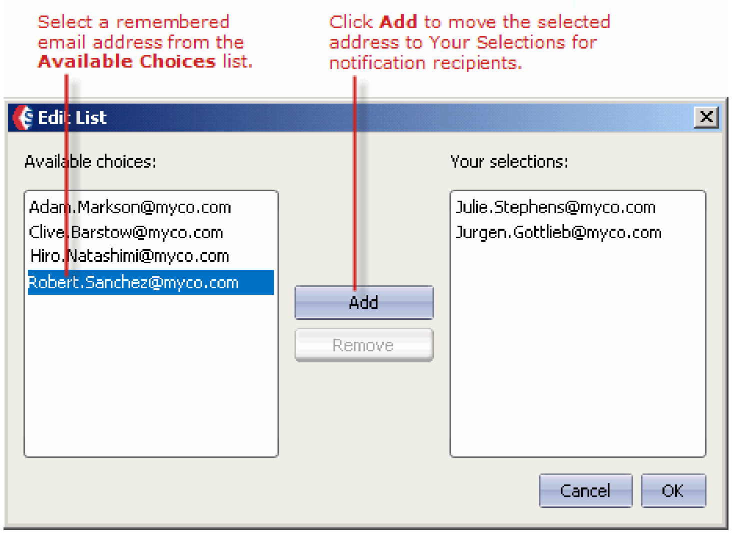

Enter an IP address or CIDR Block in the text box.
Click **Add** to add the value in the text box to the values in the list box.
CIDR blocks
10.68.120.3
Add
Remove
Notification
Type here to add a recipient email
Add

NOTE: To create additional IP addresses in your IP Range asset, repeat this step.

7 In the **Notification** group box, specify email addresses of the recipients to notify when alerts are raised against the asset. You can add new email addresses or select previously added email addresses from a list.

- To add a new address, type the email address in the **Notification** text box and click **Add**. Repeat the procedure for each new email address you want to add.
- To select previously added email addresses from a list, click **Edit List** to display the dialog box shown in Figure 10-9.

Figure 10-9: Selecting Existing Email Addresses



a In the list of **Available choices** on the left, select one or more email addresses.

b Click **Add**.

c When the list of **Your selections** has the email addresses you want, click **OK**.

8 From the Asset menu, select **Save**.

TIP: You can save changes to asset properties by pressing CTRL+S or right-clicking the asset and selecting **Save** from the popup menu.

Creating a User List Asset

To create a User List asset

1 Locate the group where you want the asset displayed and right-click below that parent in the Asset tree.

NOTE: You cannot create assets in the fixed, read-only Default group.

2 From the popup menu, select **New** and then **User List**.

The tree displays the new asset and selects it; the Properties Panel on the right displays its initial values.

3 In the **Name** field, enter the name of your new asset.

4 In the **Value** dropdown, select the value that best represents its importance to your enterprise.

IMPORTANT: To avoid problems with the Risk view in the Alerts module, do not leave the value of assets set to **none**.

5 Optionally, in the **Description** field, enter meaningful information for colleagues who also manage assets.

6 In the **Notification** group box, specify email addresses of the recipients to notify when alerts are raised against the asset. You can add new email addresses or select previously added email addresses from a list.

- To add a new address, type the email address in the **Notification** text box and click **Add**. Repeat the procedure for each new email address you want to add.
- To select previously added email addresses from a list, click **Edit List** to display the dialog box shown in Figure 10-9.
  - *In the list of **Available choices** on the left, select one or more email addresses.*
  - *Click **Add**.*
  - *When the list of **Your selections** has the email addresses you want, click **OK**.*

7 In the **Users** group box, do one of the following:

- Type a username in the text box and click **Add**. Repeat this procedure to add more usernames.

  or

- Click **Edit List** and select usernames from a list of previously entered usernames.

  NOTE: The list contains only user names that have been entered in other User List assets. It does not contain user names for HawkEye AP users unless someone has previously included the user name in a User List asset.

8 From the Asset menu, select **Save**.

TIP: You can save changes to asset properties by pressing CTRL+S or right-clicking the asset and selecting **Save** from the popup menu.

Sharing Security Assets

A single asset can belong to multiple groups. To cause the Asset Tree to display a single asset under more than one group, *share* the asset with other groups. You can share an asset with as many groups as you find useful. You can share IP Range assets, User assets and Group assets. Later, you can remove a shared asset from one group without affecting its membership in other groups.

Sharing an Asset with Another Group

When you share an asset with another group, you copy and paste it. However, the copy command works differently from a traditional copy operation. Instead of creating a new instance as the copy, the copy command creates a reference to the underlying asset definition. It is the reference to the underlying asset that you paste into the other group.

To share an asset with another asset group

1 Locate the asset in the Asset Tree and right-click to display the popup menu.

2 From the popup menu, select **Copy**.

3 Locate the other asset group in the tree where you want the asset displayed, right-click and select **Paste Into** from the popup menu.

Removing a Shared Asset from a Group

Sometimes your categorization scheme changes, and you no longer want a shared asset to belong to a particular group. If you remove a shared asset, the system removes it from the group without affecting its membership in other groups. Removing an asset never deletes the asset from the system.

To remove a shared asset from all groups, see “Deleting an Asset”, on page 289.

To remove a shared asset from a group

1 Locate the asset in the Asset Tree and right-click to display the popup menu.

2 From the popup menu, select **Remove From**.

Orphaned Assets

If you remove an asset that is not shared, it no longer belongs to a group but remains in the system. The asset becomes a so-called “orphan.” If you delete an asset group, assets within it that are not shared with other groups become orphaned, also.

So that you can select orphaned assets, the system migrates them to the fixed Enterprise Security Assets group. From there, you can move the assets to more meaningful groups, or you can delete them permanently from the system.

Modifying, Moving, and Deleting Security Assets

To modify an asset, select it in the Asset Tree in Administration mode. Its properties display in the Properties Panel to the right of the tree. You can perform the following actions on assets that you select:

- Change the Value of an asset—In the right panel, select a new value from the dropdown. The value you set for an asset affects the calculation of risk for alerts raised against the asset. For more information, see Understanding and Using Threat and Risk Views in Chapter 2, "Using Dashboards" in the *Reporting Guide*.
- Change the **Description** of an asset—In the right-hand panel, edit the text.
- Add and remove IP addresses, Users, and Notification recipients—see "Modifying Asset Properties", next.
- Move an asset—see "Moving an Asset", on page 289.
- Delete an asset—see "Deleting an Asset", on page 289.

Modifying Asset Properties

When you modify the properties of an asset, the Asset Manager prompts you to save your changes or revert to the original values before you select another asset in the tree. To save you changes explicitly, press CTRL+S before selecting another asset.

The properties you can modify depend on the kind of security asset.

This section describes the following topics:

- "Asset Group", next
- "Computer Asset", on page 287
- "Human Asset", on page 287
- "Adding IP Addresses, Notification Recipients, or Users", on page 288
- "Removing IP Addresses, Notification Recipients, or Users", on page 288

Asset Group

Property	Meaning
Name	The name displayed for the asset group in the trees of the Asset Manager and the Security Alerts widget. **NOTE:** You cannot modify the name of the permanent groups: • **System** • **Default** (displays below **Enterprise Security Assets**) • **Enterprise System Assets** • All groups that **Enterprise System Assets** contains
Description	A textual description of the asset group.

Property	Meaning
Value	The value your enterprise places on the asset group as a whole. Select from these values: **none**, **low**, **normal**, **high**, and **max**. The value you select affects the Risk view in the Security Alerts widget.
Notification	A list of email addresses that receive email messages when alerts arrive from assets in the group.

Computer Asset

Property	Meaning
Name	The name displayed for the computer asset in the trees of the Asset Manager and the Security Alerts widget. **NOTE:** You cannot modify the name of the permanent groups: • **System** • **Default** (displays below **Enterprise Security Assets**) • **Enterprise System Assets** • All groups that **Enterprise System Assets** contains
Description	A textual description of the computer asset.
Value	The value your enterprise places on the computer asset. Select from these values: **none**, **low**, **normal**, **high**, and **max**. The value you select affects the Risk view in the Security Alerts widget.
CIDR Blocks	A list of IP addresses of Classless Inter-Domain Routing (CIDR) blocks that identify a single computer asset or a collection of computer assets. Use of the following formats for specifying IP addresses in the CIDR blocks group box of an IP List asset: • Single IP Address—a complete, unique IP address with the following format: *<nnn>.<nnn>.<nnn>.<nnn>* • Block of Addresses—a CIDR block that expresses a range of single IP addresses with the following format: *<nnn>.<nnn>.<nnn>.<nnn>/<CIDR-prefix>*
Notification	A list of email addresses that receive email messages when alerts arrive from assets in the group.

Human Asset

Property	Meaning
Name	The name displayed for the human asset in the trees of the Asset Manager and the Security Alerts widget.
Description	A textual description of the human asset.
Value	The value your enterprise places on the human asset as a whole. Select from these values: **none**, **low**, **normal**, **high**, and **max**. The value you select affects the Risk view in the Security Alerts widget.
Notification	A list of email addresses that receive email messages when alerts are raised about the human asset.
Users	A list of user IDs that represent the people (or systems) that your enterprise monitors.

Adding IP Addresses, Notification Recipients, or Users

For the complete procedure to add an IP address or user, see:

- “Creating a New IP Range Asset”, on page 281
- “Creating a User List Asset”, on page 284

To add IP addresses, Notification recipients, or Users

- To add IP addresses, notification recipients, or users that do not yet exist in your system, enter the data in the appropriate field and click **Add**.

- To add a notification recipient or user that already exists in the system:

  a Click **Edit List** to display the Edit List dialog.

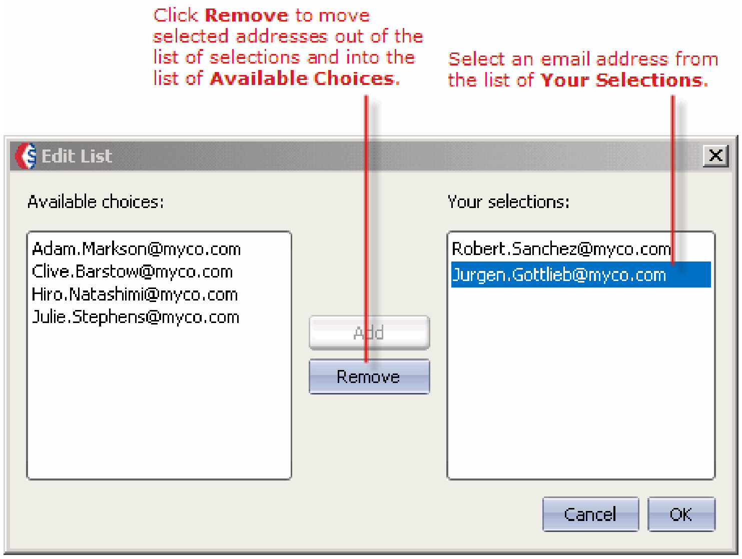
  b Select as many email addresses or users IDs under **Available choices** as desired and click **Add**.

  c When all desired email addresses or user IDs display in the **Your selections** field, click **OK** to confirm the edited list or click **Cancel** return to the asset window without changing the list.

Removing IP Addresses, Notification Recipients, or Users

Figure 10-10 illustrates the dialog in which you remove a notification recipient. The procedure for doing so is below the illustration.

Figure 10-10: Removing a Notification Recipient



Click **Remove** to move selected addresses out of the list of selections and into the list of **Available Choices.**

Select an email address from the list of **Your Selections.**

Available choices:

Adam.Markson@myco.com
Clive.Barstow@myco.com
Hiro.Natashimi@myco.com
Julie.Stephens@myco.com

Your selections:

Robert.Sanchez@myco.com
Jurgen.Gottlieb@myco.com

To remove IP addresses or Notification recipients

- To remove IP addresses or notification recipients from the system, select the value in the appropriate field and click **Remove**.

- To remove a notification recipient from an asset without removing the recipient from the system:

  a Click **Edit List** to display the Edit List dialog.

  b In the list of **Your selections**, choose email addresses you no longer want and click **Remove**.

  c After removing all unwanted email addresses, click **OK**.

Moving an Asset

If you copy an asset and later you move one of the copies from one group to another, you do not affect the other copies by moving one of them. In other words, group membership of the other copies remains the same.

To move an asset

1. Right-click the asset in the tree and select **Cut** from the popup menu.
2. Right-click the Asset Group to which you want to move the asset and select **Paste Into** from the popup menu.

Deleting an Asset

When you delete an asset from a group, its definition is deleted completely from the system. If the asset you delete is shared among other groups, it is deleted from those groups, too. After you delete an asset, it is not displayed anywhere in the tree.

If you delete an asset group, assets that belong to it and are not shared with other assets groups become orphaned. For more information see “Orphaned Assets”, on page 285.

To delete an asset

1. Locate the asset in the Asset Tree and right-click to display the popup menu.
2. From the popup menu, click **Delete**.
3. Click **Yes** to confirm the deletion.

MONITORING SYSTEM ALERTS USING HAWKEYE AP CONSOLE

This section includes the following topics:

- “System Assets and the System Alerts Widget”, next
- “Specifying Severity Level for System Alerts”, on page 291
- “Viewing the Information in a System Alert ”, on page 292
- “Modifying System Assets”, on page 293

System alerts inform you about failures in your HawkEye AP system. The System Alerts widget displays these alerts for specific HawkEye AP system assets.

NOTE: In order to access the alert widgets, users must have the `analyzer.alerts` role set on their HawkEye AP user account. (See "Administering Users and Authentication", on page 213.)

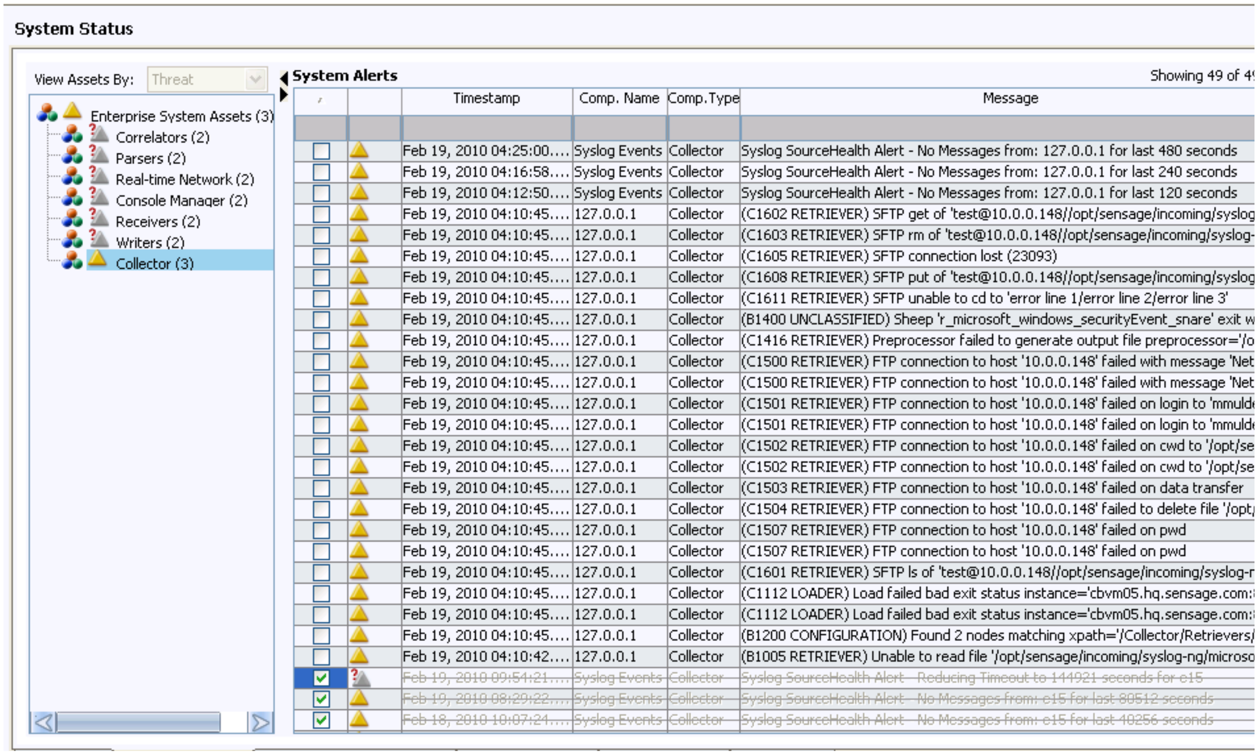
System Assets and the System Alerts Widget

HawkEye AP system assets display in the Asset module of HawkEye AP Console Administration mode. The asset tree displays a permanent asset group labeled Enterprise System Assets. Expand the group to see asset groups for the following HawkEye AP components and modules:

- Application Manager
- Parsers
- Receivers
- Real-time Network
- Collector

Figure 10-11 illustrates the System Alerts widget, accessed from HawkEye AP Console Dashboards mode.

Figure 10-11: System Alerts Widget



System Status

View Assets By: Threat

System Alerts

Showing 49 of 4

		Timestamp	Comp. Name	Comp.Type	Message
☐	⚠	Feb 19, 2010 04:25:00....	Syslog Events	Collector	Syslog SourceHealth Alert - No Messages from: 127.0.0.1 for last 480 seconds
☐	⚠	Feb 19, 2010 04:16:58....	Syslog Events	Collector	Syslog SourceHealth Alert - No Messages from: 127.0.0.1 for last 240 seconds
☐	⚠	Feb 19, 2010 04:12:50....	Syslog Events	Collector	Syslog SourceHealth Alert - No Messages from: 127.0.0.1 for last 120 seconds
☐	⚠	Feb 19, 2010 04:10:45....	127.0.0.1	Collector	(C1602 RETRIEVER) SFTP get of 'test@10.0.0.148//opt/sensage/incoming/syslog
☐	⚠	Feb 19, 2010 04:10:45....	127.0.0.1	Collector	(C1603 RETRIEVER) SFTP rm of 'test@10.0.0.148//opt/sensage/incoming/syslog-
☐	⚠	Feb 19, 2010 04:10:45....	127.0.0.1	Collector	(C1605 RETRIEVER) SFTP connection lost (23093)
☐	⚠	Feb 19, 2010 04:10:45....	127.0.0.1	Collector	(C1608 RETRIEVER) SFTP put of 'test@10.0.0.148//opt/sensage/incoming/syslog
☐	⚠	Feb 19, 2010 04:10:45....	127.0.0.1	Collector	(C1611 RETRIEVER) SFTP unable to cd to 'error line 1/error line 2/error line 3'
☐	⚠	Feb 19, 2010 04:10:45....	127.0.0.1	Collector	(B1400 UNCLASSIFIED) Sheep 'r_microsoft_windows_securityEvent_snare' exit w
☐	⚠	Feb 19, 2010 04:10:45....	127.0.0.1	Collector	(C1416 RETRIEVER) Preprocessor failed to generate output file preprocessor='/o
☐	⚠	Feb 19, 2010 04:10:45....	127.0.0.1	Collector	(C1500 RETRIEVER) FTP connection to host '10.0.0.148' failed with message 'Net
☐	⚠	Feb 19, 2010 04:10:45....	127.0.0.1	Collector	(C1500 RETRIEVER) FTP connection to host '10.0.0.148' failed with message 'Net
☐	⚠	Feb 19, 2010 04:10:45....	127.0.0.1	Collector	(C1501 RETRIEVER) FTP connection to host '10.0.0.148' failed on login to 'mmulde
☐	⚠	Feb 19, 2010 04:10:45....	127.0.0.1	Collector	(C1501 RETRIEVER) FTP connection to host '10.0.0.148' failed on login to 'mmulde
☐	⚠	Feb 19, 2010 04:10:45....	127.0.0.1	Collector	(C1502 RETRIEVER) FTP connection to host '10.0.0.148' failed on cwd to '/opt/se
☐	⚠	Feb 19, 2010 04:10:45....	127.0.0.1	Collector	(C1502 RETRIEVER) FTP connection to host '10.0.0.148' failed on cwd to '/opt/se
☐	⚠	Feb 19, 2010 04:10:45....	127.0.0.1	Collector	(C1503 RETRIEVER) FTP connection to host '10.0.0.148' failed on data transfer
☐	⚠	Feb 19, 2010 04:10:45....	127.0.0.1	Collector	(C1504 RETRIEVER) FTP connection to host '10.0.0.148' failed to delete file '/opt/
☐	⚠	Feb 19, 2010 04:10:45....	127.0.0.1	Collector	(C1507 RETRIEVER) FTP connection to host '10.0.0.148' failed on pwd
☐	⚠	Feb 19, 2010 04:10:45....	127.0.0.1	Collector	(C1507 RETRIEVER) FTP connection to host '10.0.0.148' failed on pwd
☐	⚠	Feb 19, 2010 04:10:45....	127.0.0.1	Collector	(C1601 RETRIEVER) SFTP ls of 'test@10.0.0.148//opt/sensage/incoming/syslog-r
☐	⚠	Feb 19, 2010 04:10:45....	127.0.0.1	Collector	(C1112 LOADER) Load failed bad exit status instance='cbvm05.hq.sensage.com:
☐	⚠	Feb 19, 2010 04:10:45....	127.0.0.1	Collector	(C1112 LOADER) Load failed bad exit status instance='cbvm05.hq.sensage.com:
☐	⚠	Feb 19, 2010 04:10:45....	127.0.0.1	Collector	(B1200 CONFIGURATION) Found 2 nodes matching xpath='/Collector/Retrievers/
☐	⚠	Feb 19, 2010 04:10:42....	127.0.0.1	Collector	(B1005 RETRIEVER) Unable to read file '/opt/sensage/incoming/syslog-ng/microso
☑	?⚠	~~Feb 19, 2010 09:54:21....~~	~~Syslog Events~~	~~Collector~~	~~Syslog SourceHealth Alert - Reducing Timeout to 144921 seconds for c15~~
☑	⚠	~~Feb 19, 2010 08:29:22....~~	~~Syslog Events~~	~~Collector~~	~~Syslog SourceHealth Alert - No Messages from: c15 for last 80512 seconds~~
☑	⚠	~~Feb 18, 2010 10:07:24....~~	~~Syslog Events~~	~~Collector~~	~~Syslog SourceHealth Alert - No Messages from: c15 for last 40256 seconds~~

When you select the Enterprise System Assets group in the tree, the alert table displays all alerts from your HawkEye AP system, regardless of component or module. If you select an asset group for a specific kind of HawkEye AP component or module, the alerts table displays only alerts about components or modules of that type.

The Enterprise System Assets tree on the left displays a number within parentheses at the end of each component name. These numbers represent the priority of the alert as defined in its alerting rule. The higher the number, the greater the priority.

Information in an alert identifies specific components and modules by name in the **Comp. Type** column. The name of component as it is defined for the deployment displays in the **Comp. Name** column.

As indicated by the blue highlight on the COLLECTOR system asset in Figure 10-11, the example widget currently displays only Collector alerts. All values in **Comp. Type** column echo this information. The values in the **Comp. Name** column indicate problems with two different Collectors, named **Syslog Events**, and **127.0.0.1**.

The **Message** column displays the full log message. For a Collector message, the format is:

```
(<error_code> <error_type>): <full_error_message>
```

For most Collector messages, the error type is `UNCLASSIFIED`. However, several of the most common messages have been classified as one of the following:

- `RETRIEVER`

  Messages classified as `RETRIEVER` are further identified in the beginning of the message text as either `FTP` or `SFTP`.

- `CONFIGURATION`
- `LOADER`

Specifying Severity Level for System Alerts

As illustrated in Figure 10-11, the System Alerts widget displays icons before each asset. These icons indicate the severity level of alerts against each asset. The Severity levels are:

Status Icon	Severity	Significance	HawkEye EPL Constant
🛑	Fatal	denotes an unrecoverable failure; requires you to restart the component or module	SEVERITY_FATAL
▼	Error	denotes a recoverable failure; may require a configuration change	SEVERITY_VISIBLE_ERROR
			SEVERITY_ERROR
⚠	Warning	denotes an issue to address that might lead to failure; for example, certain limits are exceeded	SEVERITY_WARNING
▬	Detail	denotes no system alerts	SEVERITY_DETAIL
	Verbose		SEVERITY_VERBOSE

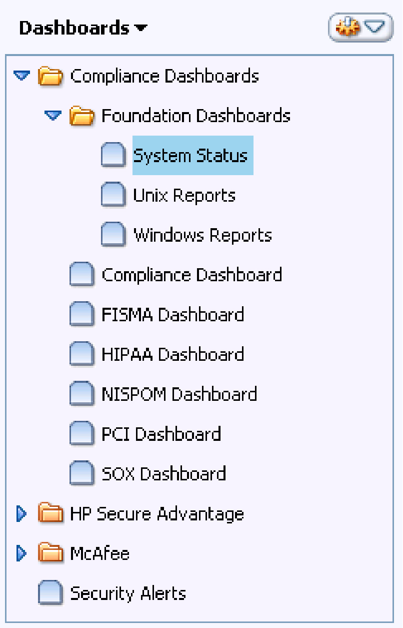
For more information on the six HawkEye Event-Processing Language constants that represent the level of severity for a System alert, see sendSystemAlert() in Appendix A: HawkEye Event Processing Language (HEPL) Reference of the *HawkEye Event Processing Language Developers Guide*.

Viewing the Information in a System Alert

Although you can create your own dashboard and drag the System Alerts widget to it, you can access this widget in the **System Status** dashboard, which is included among the Foundations Dashboards (below Compliance Dashboards). These dashboards are installed automatically if you install the Foundation Analytics Package when you install and configure your HawkEye AP system. The **System Status** dashboard includes several pages that include helpful system status information. Figure 10-11 illustrates this dashboard; the pages at the bottom of the dashboard indicate the other types of information that the dashboard provides.

Figure 10-12 illustrates the tree of dashboards that HawkEye AP provides, and highlights the location of the **System Status** dashboard.

Figure 10-12: Accessing the System Alerts Widget



The HawkEye AP System Alerts widget includes the following information for each alert:

- **timestamp**— date and time-of-day the alert was sent
- **Component Name**—name of the HawkEye AP component that is failing
- **Component Type**—Type of component.
- **Message**—text that describes the specific kind of failure
- **Reference Number**—unique identifier for the system alert

Additionally, the first column of every alert widget is the unnamed Acknowledged column. This column displays a select box for each row. Select the checkbox of desired row(s) to indicate that you have viewed the alert. The display of an acknowledged row changes in the following ways:

- The text is greyed.
- A line is drawn through the text.

- The row moves to the bottom of the alert-widget table.

For an illustration of acknowledged alerts, see the bottom rows in Figure 10-11.

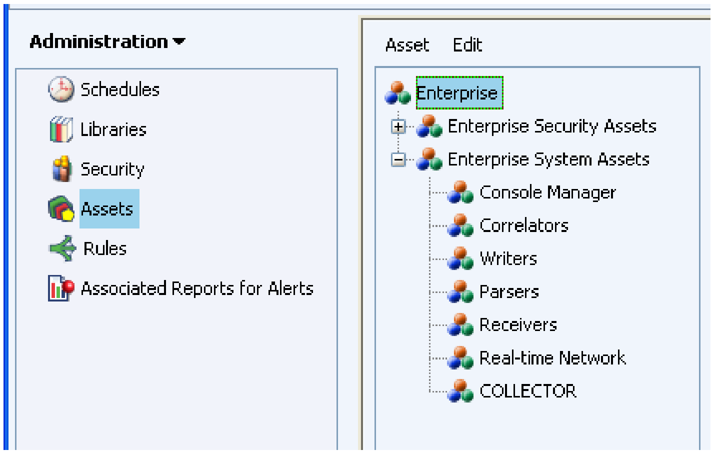
NOTE: To return an acknowledged row to unacknowledged status, deselect its checkbox. The row returns to its original location without the line through the text and without greyed text.

IMPORTANT: Application Manager system alerts display in the System Alerts widget and are recorded in log files called *application manager logs*. You can examine these files to see log entries that occurred before and after the system alert. Examining the surrounding entries can help you determine the root causes of system failures. For more information, see “Investigating Application Manager Problems and Events in Log Files”, on page 184.

Modifying System Assets

In addition to the user-defined security assets that you create and manage, the Asset Manager displays Enterprise System Assets. This group and the groups it contains are fixed and read-only. You cannot create, share, move, or delete system asset groups. You can only modify their notification recipients.

Figure 10-13: System Assets



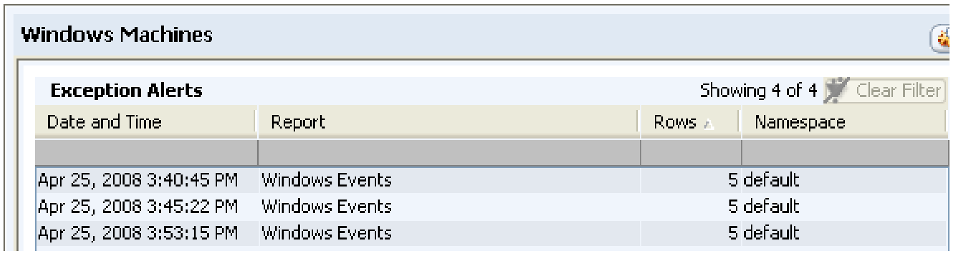
Monitoring Exception Alerts

You view Exception Alerts using Dashboard mode in HawkEye AP Console. The Exception Alerts widget displays the following information about the report that raised the alert:

- Date and Time— time the report ran
- Report Name— name of the report
- Rows— number of rows returned by the report

- Namespace—namespace against which the report ran

Figure 10-14: Exception Alerts Widget



Windows Machines

Exception Alerts

Showing 4 of 4

Date and Time	Report	Rows	Namespace
Apr 25, 2008 3:40:45 PM	Windows Events	5	default
Apr 25, 2008 3:45:22 PM	Windows Events	5	default
Apr 25, 2008 3:53:15 PM	Windows Events	5	default

For information see:

- Viewing Exception Alerts in Chapter 2, “Using Dashboards” in the *Reporting Guide*.
- Creating Schedules in Chapter 7, “Creating and Editing Schedules” in the *Reporting Guide*.

CHAPTER 11

Monitoring Source Health

Collection from log sources sometimes fails. If undetected, collection failures leave gaps in the historical event record stored in the Event Data Warehouse (EDW). A *source health failure* occurs when the volume of data loaded into the SLS from a particular log source varies from the expected amount.

The source health monitor tracks the health of specified log sources by periodically scanning the SLS to verify that it has recently received the expected volume of log records from each specified log source. When a monitored log source has not produced the expected amount of data, the source health monitor raises system alerts.

This chapter contains these sections:

- "How Monitoring Works", next
- "Configuring Source Health Monitoring", on page 296
- "Adding, Changing, Removing, & Listing Source Health Definitions", on page 306
- "Investigating Source Health Failures", on page 307

How Monitoring Works

After source health monitoring has been enabled, the monitor begins running periodic queries against recently loaded data. It aggregates the result data and stores it in the EDW. At each monitoring run, it determines whether sufficient data has been collected to run historic analysis. If the monitor determines that data is insufficient, it runs a query to collect the data. If this query fails to collect more data, it concludes the current run. Data may be insufficient because the monitor has just been enabled for a particular log source or because a query definition changed in a way that invalidates the historic data.

The monitor creates an expected log volume for each log source by examining historic data for the same period one, two, and three weeks previously. After accumulating this data, the source health monitor compares the volume of historic data against the volume of data in the current run.

If the current volume of data falls below the expected volume by a configured variance, the source health monitor generates a notification. It compares the notification against a master list of log sources to determine whether it has already raised alerts for the log source. If it has raised fewer alerts than a configured number, it sends an alert and increments the alert counter for the log source.

When data is received for a log source, the source health monitor examines the master list for that log source. When sufficient data arrives for a log source that has accumulated alerts, the monitor clears all alerts for that log source.

For information on enabling and changing source health monitoring, see "Adding, Changing, Removing, & Listing Source Health Definitions", on page 306.

Configuring Source Health Monitoring

HawkEye AP stores all log data it receives, both batch and streaming, in log tables. Each log table identifies the source of each data row. Some data is identified by a single column, such as the host name or IP address. Some data is identified by a set of columns, such as the application name and the IP address. To enable the source health system to monitor and raise system alerts for a specific log source, you must configure the source health monitor to recognize the source ID. The most important aspect of configuration is specifying the log table and columns that identify the log source.

To configure source health monitoring, you create a *source health definition* in a file that uses XML format. You specify the tables and columns that identify the log sources to query, the times to query, the percentage of variance, and the maximum number of times to raise an alert about a detected source-health failure. You create a separate file for each definition. Each definition specifies the name of the EDW table that stores the log data and the log source that generates the data.

The containing element for each source health definition is the `<SourceHealth>` element. You specify values for attributes of this element as well as for elements within this element. Each definition can contain only one `<SourceHealth>` element.

This section contains the following topics:

- “Configuring the Tables and Log Sources”, next
- “Configuring the Schedule Period”, on page 298
- “Configuring Variance and Maximum Alerts”, on page 300
- “Summary of Elements and Attributes in a Source Health Definition”, on page 300
- “Examples of Source Health Definitions”, on page 303
- “Source Health Definition DTD”, on page 303
- “Determining Source Health Monitoring Schedules”, on page 304

Configuring the Tables and Log Sources

To enable the source health monitor to identify your log sources, you must define one or two elements within the `<SourceHealth>` element. These elements define the event tables to monitor and the log sources within those tables.

If you are monitoring a single log table and do not need to limit data with a `WHERE` clause, you can use a simple syntax to identify the table and log source. If you require a more complex query, you must specify the full query in the definition.

Specifying a Simple Query and Single Event Table

To monitor a single event-log table, you specify values for two elements:

- `<Table>` element—specify the fully qualified table name

  You must include the namespace in which the table is located.

- `<Expression>` element—specify the column that identifies the log sources

The expression element contains either a VARCHAR column name or a Sensage SQL expression that evaluates to a VARCHAR. The VARCHAR value must identify the log sources, such as IP addresses, host names, application names, or a combination of these.

NOTE: When you use this syntax, you must include both the <Table> and <Expression> elements and the expression element must follow the table element.

EXAMPLES

The following simplified example illustrates a source health definition that specifies a table and expression.

```
<SourceHealth>
    <Table>myNamespace.syslog</Table>
    <Expression>log_source_id</Expression>
</SourceHealth>
```

The following simplified example illustrates a source health definition that computes the value of the log source in the expression.

```
<SourceHealth>
    <Table>myNamespace.snare</Table>
    <Expression>_strjoin('_',snare_user_name,host)</Expression>
</SourceHealth>
```

Specifying a Complex Query or Multiple Event Tables

To monitor more than one log table or to restrict the rows returned, you must define the query in the <Query> element. This element enables complex queries, such as those that combine several log tables with UNION ALL clauses or those that eliminate certain log sources with a WHERE clause.

SYNTAX

The syntax for the query element is:

```
<Query>
   SELECT ts, <expression> AS source
      FROM <namespace>.<log_table>
      [WHERE NOT IN <list_of_log_sources_to_eliminate>]
      DURING ALL

   [UNION ALL

   SELECT ts, <expression> AS source
      ...
   ]
</Query>
```

The first column, which must be named ts, contains the timestamp of the selected log records. The second column must be renamed to source and must be a VARCHAR.

The <expression> element contains either a VARCHAR column name or a Sensage SQL expression that evaluates to a VARCHAR. The VARCHAR value identifies the log sources, such as IP addresses, host names, application names, or a combination of these.

The *<namespace>*.*<log_table>* elements specify the fully qualified name of the monitored event-log table.

NOTE:

- When you use this syntax, you must not include the `<Table>` and `<Expression>` elements.
- The source health monitor processes the query that you configure as a subquery. The main query aggregates volume data by source ID. To enable this aggregation, the query you configure must return two result columns: the first column must be a timestamp column called `ts` and the second must be a `VARCHAR` column call `source`.

EXAMPLE

The following simplified example illustrates a source health definition that specifies a query.

```
<SourceHealth>
    <Query>
        WITH $logtable AS myNamespace.cdrlog
        WITH $source_id AS exchg_id
        SELECT ts, $source_id AS source
          FROM $logtable
          DURING ALL
    </Query>
</SourceHealth>
```

Configuring the Schedule Period

To schedule the period over which the source health monitor queries the event tables, you define three schedule points. You define these points as attributes of the `<SourceHealth>` element:

- **frequency point**—determines the number of times within a 24-hour period that monitoring should occur

  Specify this value in the `<DailyFrequency>` attribute of the source health definition. The default value is `24`, which causes monitoring to occur hourly.

  The value must be evenly divisible into `24` to ensure data is collected and analyzed at the same hour each day. If you specify a value of `6`, the historical record is examined every four hours, starting at midnight local time of the Application Manager.

- **run point**—determines when the monitoring run actually begins; configured to ensure that the monitoring run coincides with the log-roll points of batch data collection. You configure an offset that delays monitoring by a specified number of minutes.

  Specify this value in the `<DailyFrequencyOffset>` attribute of the source health definition. The default and the minimum values are `0`.

  For example, assume you want monitoring to occur three times within each 24-hour period; you set the `<DailyFrequency>` attribute to `3`. By default, monitoring would occur at midnight, 8 a.m., and 4 p.m. Assume further that you want to delay each run by ninety minutes to coincide with the log-roll points of your batch data collection. To configure the monitoring points to 1:30 a.m., 9:30 a.m., and 5:30 p.m., you set the `<DailyFrequencyOffset>` attribute to `90`.

  Although there is no maximum value for this attribute, there is no point in specifying an offset value that is greater than the interval between monitoring runs. For example, if you have scheduled hourly runs, there is no point in setting the offset greater than `60`. However, if you do provide an offset value that is greater than the run interval value, the source health monitor subtracts the smaller value from the larger. For example, if you have scheduled hourly runs and a daily frequency offset of `70`, the source health monitor sets the offset to `10`.

NOTE: The combined values of the `<DailyFrequency>` and `<DailyFrequencyOffset>` attributes determine exactly when the source health monitor definition runs.

- **monitoring point**—ensures that batch loading completes before examining the log records. You configure an offset that delays monitoring by a specified number of hours, which causes the source health monitor to examine only log records whose timestamp is older than the time specified by the offset.

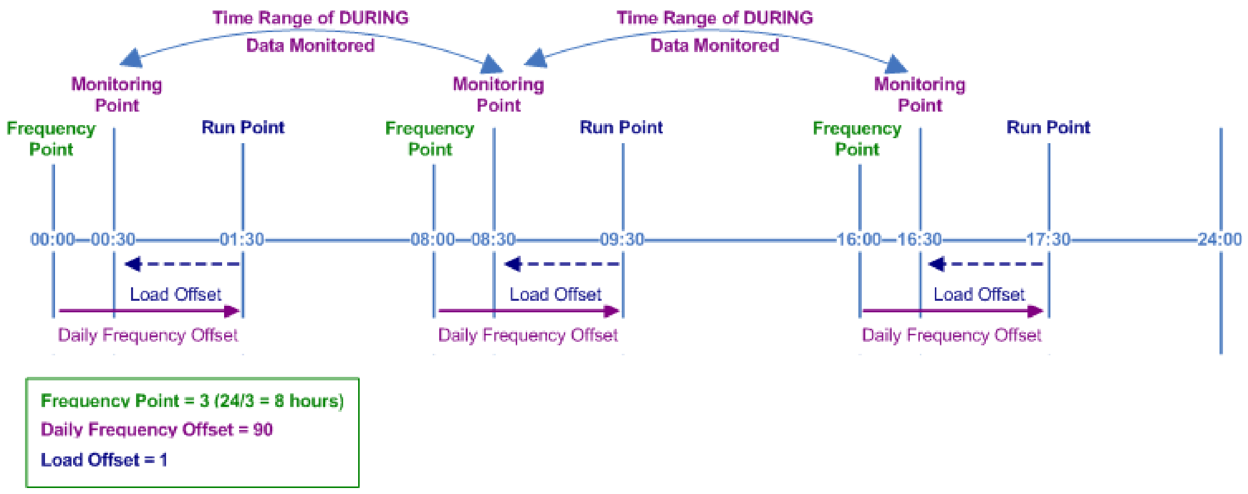
  Specify this value in the `<LoadOffset>` attribute of the source health definition. The default value is `1` and the minimum value is `0`. There is no maximum value but any value greater than `24` is interpreted as `24`.

  Each monitor run examines the data loaded into the EDW since the last run. When you configure the run, take into consideration that the most recent data may not be available because it has not yet been loaded into the event table in the EDW. The load offset attribute allows you delay when the source health monitor begins examining log records for the current run. For example, if your data takes four hours to load, you might want to set your load offset to 6 hours. This value ensures sufficient time for transmission and minor load delays.

  NOTE: The monitoring point determines the start and finish times of the monitored data. From a Sensage SQL perspective, the period over which the data is monitored, which is the period between monitoring points, sets the value of the `DURING ALL` clause.

Figure 11-1 illustrates a scheduling period in which the frequency has been set to three times each day, the frequency offset has been set to 90 minutes, and the load offset has been set to one hour.

Figure 11-1: Example Scheduling Period



For more information on scheduling, see “Determining Source Health Monitoring Schedules”, on page 304. For an example of setting these attributes, see “Specifying a Simple Query and Single Event Table”, on page 296.

Configuring Variance and Maximum Alerts

In addition to setting the scheduling period attributes, you can set values for the attributes that define the variance and maximum number of alerts.

Specifying Variance

Use the `<Variance>` attribute of the `<SourceHealth>` element to specify the maximum percentage by which the measured volume of loaded events for a log source varies from the historical rolling average for the source. The source health monitor raises a source health alert when the specified variance is reached or exceeded.

For example, if you specify a value of `20`, a source health alert is raised if the measured volume for a given hour of a specific day of the week is more than 20% less than historical rolling average for that hour and day of the week.

The default value is `10`. The maximum value is `100` and maximum value is `0`. Hexis Cyber Solutions recommends that you do not set the value below `10`. Setting low values can cause too many system alerts.

NOTE: Specifying a variance of `100` puts the source health monitor into a special operation mode in which it does not examine historic data. Instead, it compares the current source IDs with a list of previously received source IDs. If a source ID exists in the previous list but not in the current list, the source health monitor raises an alert for the previous source ID.

For more information on setting this attributes, see “Adjusting Attribute Values on a Running System”, on page 305. For an example of setting this attribute, see “Specifying a Simple Query and Single Event Table”, on page 296.

Specifying Maximum Alerts

Use the `<MaxAlerts>` attribute of the `<SourceHealth>` element to specify the maximum of number of times to raise an alert about a detected source health failure. After the specified number of alerts have been raised, the log source is marked as inactive and no further alerts are sent for that log source.

If log records subsequently arrive for the log source, the count is reset and, if the log source has been marked as inactive, it is reset to active.

Set a value of `unlimited` for this attribute to specify that there is no limit to the number of alerts to raise for a log source.

The default value is `unlimited`. Acceptable values are a non-negative integer or `unlimited`.

Summary of Elements and Attributes in a Source Health Definition

Source Health Definition Elements

The table below summarizes the elements of a source health definition.

Element or Attribute	Meaning
`<SourceHealth>`	The containing element for source health definitions. Each definition file can contain only one `<SourceHealth>` element.
`<Table>`	An element within a `<SourceHealth>` element that specifies an event table to monitor, including the namespace in which the table is located. If you include a `<Table>` element, you must follow it with an `<Expression>` element and must not include a `<Query>` element. For more information, see "Configuring the Tables and Log Sources", on page 296.
`<Expression>`	An element within a `<SourceHealth>` element that specifies a SQL `varchar` column expression that identifies log sources, such as by IP address, host name, application name, or a combination of them. If you include an `<Expression>` element, you must precede it with a `<Table>` element and must not include a `<Query>` element. For more information, see "Configuring the Tables and Log Sources", on page 296.
`<Query>`	An element within a `<SourceHealth>` element that allows you to define a complex query or to identify multiple event tables. If you include a `<Query>` element, do not include a `<Table>` or an `<Expression>` element. For more information, see "Configuring the Tables and Log Sources", on page 296.

Source Health Definition Attributes

The table below summarizes the attributes of a source health definition.

Element or Attribute	Meaning
LoadOffset	An attribute of a <SourceHealth> element that specifies the number of hours to allow to ensure that the data has been loaded into the SLS. The default value is "1" and the minimum value is "0". For more information, see "Configuring the Schedule Period", on page 298 and "Determining Source Health Monitoring Schedules", on page 304.
Variance	An attribute of a <SourceHealth> element that specifies the maximum percentage by which the measured volume of loaded events for a log source varies from the historical rolling average for the source. The default value is "10", the minimum value is "0", and the maximum value is "100". • Specify "100" to raise an alert only when the current monitoring period does not include log records for all source IDs previously recorded. In other words, if a source ID exists in the previous list but not in the current list, the source health monitor raises an alert for the previous source ID. • Sensage recommends that you do not set the value below "10" to avoid raising too many system alerts. For more information, see "Configuring Variance and Maximum Alerts", on page 300.
MaxAlerts	An attribute of a <SourceHealth> element that specifies the maximum of number of times to raise an alert about a detected source health failure. The default value is "unlimited" and the acceptable values are "unlimited" or a non-negative integer. For more information, see "Configuring Variance and Maximum Alerts", on page 300.
<Table>	An element within a <SourceHealth> element that specifies an event table to monitor, including the namespace in which the table is located. If you include a <Table> element, you must follow it with an <Expression> element and must not include a <Query> element. For more information, see "Configuring the Tables and Log Sources", on page 296.
<Expression>	An element within a <SourceHealth> element that specifies a SQL varchar column expression that identifies log sources, such as by IP address, host name, application name, or a combination of them. If you include an <Expression> element, you must precede it with a <Table> element and must not include a <Query> element. For more information, see "Configuring the Tables and Log Sources", on page 296.
<Query>	An element within a <SourceHealth> element that allows you to define a complex query or to identify multiple event tables. If you include a <Query> element, do not include a <Table> or an <Expression> element. For more information, see "Configuring the Tables and Log Sources", on page 296.

Examples of Source Health Definitions

This section presents the same source health definitions that were included as examples in "Configuring the Tables and Log Sources", on page 296. However, the examples above do not illustrate setting the values for schedule period, variance, or maximum alerts. The examples in this section more closely resemble a typical definition.

The first two examples illustrates the simple format, in which you identify the table and log source. The third example illustrates the more complex format, in which you specify the full query in the definition.

Simple Syntax

The following two examples illustrate how to configure monitoring of a single event-log table when the query does not require a WHERE clause.

The following example illustrates a source health definition that specifies a table and expression.

```
<SourceHealth DailyFrequency="6" DailyFrequencyOffset="90"
 LoadOffset="2" MaxAlerts="5">
    <Table>myNamespace.syslog</Table>
    <Expression>log_source_id</Expression>
</SourceHealth>
```

The following example illustrates a source health definition that computes the value of the log source in the expression.

```
<SourceHealth DailyFrequency="6" DailyFrequencyOffset="90"
 LoadOffset="2" MaxAlerts="5">
    <Table>myNamespace.snare</Table>
    <Expression>_strjoin('_',snare_user_name,host)</Expression>
</SourceHealth>
```

Complex Query or Multiple Event

The following example illustrates a source health definition that specifies a query.

```
<SourceHealth DailyFrequency="6" DailyFrequencyOffset="90"
 LoadOffset="2" MaxAlerts="5">
    <Query>
        WITH $logtable AS myNamespace.cdrlog
        WITH $source_id AS exchg_id
        SELECT ts, $source_id AS source
          FROM $logtable
          DURING ALL
    </Query>
</SourceHealth>
```

Source Health Definition DTD

The code below is the DTD of a source health definition.

```
<?xml version="1.0" standalone="no" ?>
<!DOCTYPE sourcehealth [
<!ATTLIST sourcehealth
name CDATA #REQUIRED
frequency CDATA #IMPLIED
start CDATA #IMPLIED
delay CDATA #IMPLIED
margin CDATA #IMPLIED
maxalerts CDATA #IMPLIED >
<!ELEMENT query (#PCDATA)>
<!ELEMENT table (#PCDATA)>
<!ELEMENT expression (#PCDATA)>
<!ELEMENT spec (table, expression)
<!ELEMENT sourcehealth (query | spec)>
]>
```

Determining Source Health Monitoring Schedules

The most important consideration when you schedule monitoring runs is how often the log data loads into the EDW. The two extremes are continuous loading or once-a-day loading. There is no need to monitor source health more frequently than data is loaded into the EDW.

- If data is loaded once each day, you can set the `<DailyFrequency>` attribute to `1`.
- If data is loaded continuously, you can keep the `<DailyFrequency>` attribute set to the default of `24` for hourly monitoring.

Scheduling the Monitoring of Infrequently Loaded Data

When scheduling the monitoring of infrequently loaded data, you will probably need to change the values of all three scheduling attributes. For example, assume your log records are loaded nightly between 10:00 p.m. and 2:00 a.m. You would want to set the monitoring run to begin at 3:00 a.m. and to examine only log records loaded before 9:00 p.m. You would set the three attributes as follows:

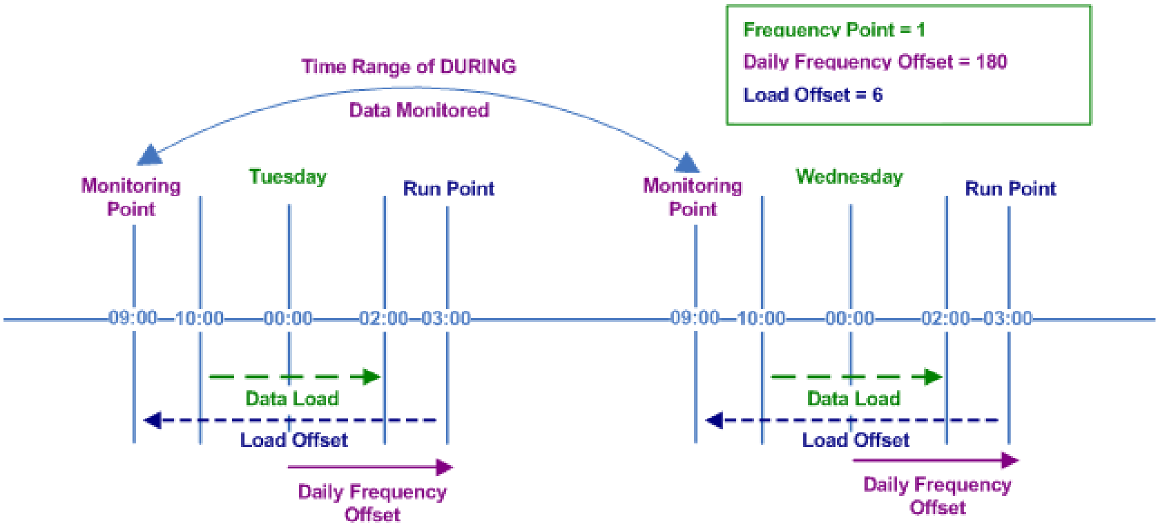
- `<DailyFrequency> = 1`
- `<DailyFrequencyOffset> = 180`
- `<LoadOffset> = 6`

Setting the load offset to `6` and the daily frequency offset to `180` allows for:

- 4 hours of load time
- 1 hour after the load before the monitoring run begins
- 1 hour before the load to allow for transmission and other minor delays in sending the log files

The setting above causes the source health monitor to begin running at 3 a.m. and to monitor data for the period beginning 24 hours before yesterday at 9 p.m., as illustrated in Figure 11-2 below.

Figure 11-2: Example of Once-a-Day Scheduling



Scheduling the Monitoring of Frequently Loaded Data

When scheduling the monitoring of frequently loaded data, you will probably need to set the value of `<DailyFrequencyOffset>`, but you may not need to change the values of the other two scheduling attributes. For example, if an important report runs every hour on the hour, you should set the source health monitor to begin late enough so that its monitoring does not interfere with the report run.

If you set the source health monitor to examine log records hourly, keeping the default value of `1` for `<LoadOffset>` should ensure that the monitor does not analyze log records before they have been loaded. Assume you keep the default value of `1` for `<LoadOffset>` and the default value of `0` for `<DailyFrequencyOffset>`. The monitoring run that begins at 11:00 examines log records loaded between 9:00 and 10:00; the monitoring run that begins at 12:00 examines log records loaded between 10:00 and 11:00.

Adjusting Attribute Values on a Running System

While the system is running, you can modify configuration values to tune system behavior. For example, if you are receiving too few system alerts, you can decrease the variance.

If you are receiving too many alerts, you should determine what is causing them. For batch loading, perhaps the data is being loaded too late to be examined by the source health query. Increasing the load offset might resolve this problem. For real-time loading, decreasing the daily frequency would spread the log-record volume over a longer period and might smooth out variability that can cause spurious alerts.

If you continue to see too much variability in the number of log records loaded, you can increase the variance to eliminate unwanted alerts. If you set the variance to `100`, you invoke a special mode of operation in which the source health monitor does not examine the volume of data. Instead it reports only on the sources that have not produced any records since the last run.

About the Time Zone

HawkEye AP stores log data in the EDW with a UTC (coordinated universal time) time zone in the timestamp column. Doing so keeps log record from different time zones comparable. The `<DailyFrequency>` and `<DailyFrequencyOffset>` values are defined with a baseline of midnight in the local time of the Application Manager. The source health monitor converts local time to UTC for its queries on log tables.

ADDING, CHANGING, REMOVING, & LISTING SOURCE HEALTH DEFINITIONS

Monitoring source health failures begins only after you add at least one source health definition to your deployment of HawkEye AP. Use the `clsetup` command to add, change, remove, and list source health definitions.

NOTE: Before you run any of the commands documented in this section, your HawkEye AP system must already be running. In other words, you must have already configured the EDW and other HawkEye AP components.

Adding or Changing a Source Health Definition

To add a new source health definition or change an existing definition, use the `clsetup` command with the following syntax:

```
clsetup [add | change] sourcehealth <shm_name> --definition=<filename>
```

NOTE:

- You can use the `change` option to modify any attribute value at any time. If you change any of the scheduling attributes, previously collected data remains unchanged.
- If you change a definition so that it refers to different tables or log sources, the system removes the current set of historical event data and the list of sources because they not relevant to the new definition.
- The `clsetup` command restarts the Application Manager when you add, change, or remove a source health definition.

Removing a Source Health Definition

To remove a source health definition and terminate the monitoring of the tables it specifies, use the `clsetup` command with the following syntax:

```
clsetup remove sourcehealth <shm_name>
```

When you remove a definition, the system removes all information about the definition.

Listing Source Health Definitions

To list all source health definitions or the text that defines one, use the `clsetup` command with the following syntax:

```
clsetup list sourcehealth [<shm_name>]
```

If you omit the source-health-monitor name, the command returns the name of every existing definition.

If you provide the source-health-monitor name, the command returns the text that defines the specified definition, if one exists.

Investigating Source Health Failures

The path that log events follow from the time that log sources write them and the EDW returns them in queries is long. Many processing steps intervene along the way, and the steps vary depending on whether you collect them in streams or batches. Log events written by particular log sources can get stuck at different steps, due to network, hardware, or system failures. If the situation is not resolved quickly, log events can be lost.

The "How Monitoring Works", on page 295 section of this chapter explains how monitoring works, how to configure it, and how to create and modify source health definitions. After you configure a source health definition and add it to your SLS instance, the system displays the following system alerts to help you identify potential problems:

- **No SLS upload in the last <*number*> hours**—indicates that no log events have been loaded into the SLS; raised only for batch loads. This may be normal for your system. If not, the situation can lead to follow-on source health alerts. Correct the problem as soon as possible.
- **No successful SLS upload in the last <*number*> hours**—indicates that no log events have been loaded into the SLS; raised only for real-time loads. This may be normal for your system. If not, the situation can lead to follow-on source health alerts. Correct the problem as soon as possible.
- **Source Health Alert for ID**—indicates that the source health monitor has discovered a gap in log events from a specified log source during the most recent monitoring run. The message contains the identity of the log source with a gap in its log events, along with the start time and duration during which the gap was discovered.

  Example:

  ```
  Source Health failure from monitor Monitor for ID: EXCH-11013. Period start:
  2007-04-23T19:00:00.000Z, duration: 1 hours. Expected volume 50, actual volume
  16
  ```

  This alert recurs for a log source, start time, and duration until one of the following occurs:

  - You fix the problem and the number of log events loaded into the EDW returns to normal.

  —or—

  - The number of "Source Health Alert for ID" alerts exceeds the threshold specified in the source health definition for the log source.

  To investigate the potential health failure, start with the log source itself. If the log source is active and online, ensure that the receiver or retriever through which its log events enter your HawkEye AP deployment is working correctly. Finally, ensure that the loader that stores log events from the source in the EDW is working correctly. If necessary, load the missing log events manually with the `atload` command to fill in the gap.

- **Source Health System Configuration Error**—indicates that a configuration error was encountered when the Application Manager started. This alert typically displays immediately after you add or change a source health definition. The configuration is read-only when Application Manager starts.

  Example:

  ```
  Source Health configuration error from monitor cdr_exchanges: Invalid value for
  Attribute: DailyFrequency
  ```

- **Source Health System Runtime Error**—indicates that an error occurred during a source health monitoring run. The monitoring run stops immediately, and it waits for the next scheduled monitoring run to begin. If runtime errors occur in three consecutive monitoring runs, monitoring runs for that definition are suspended until the Application Manager is restarted. Because the Application Manager is known internally as the *Controller*, see the Controller server log for details.

  Example:

  ```
  Source Health runtime error from monitor cdr_exchanges: Query of log for source
  IDs failed
  ```

CHAPTER 12

Troubleshooting

This chapter includes these sections:

- “Checking Space for the Data Store”, next
- “Monitoring for Inconsistent Loads”, on page 309
- “Handling Rendezvous Timeouts”, on page 310
- “Improving Query Performance”, on page 310

Checking Space for the Data Store

To determine the space and inode (i-number) usage for data store, use the `ds_stat` utility. Run the `ds_stat` command to make estimates and projections of disk space usage on the EDW.

```
ds_stat { <file> | <directory> }
```

The `ds_stat` utility takes a single parameter, which is the name of a directory or the name of a file. If the argument is a file, the utility generates disk usage statistics for that file. If the argument is a directory, utility generates a summary of the disk usage statistics for the entire directory tree.

The statistics that `ds_stat` generates are:

- Total inodes used
- Total file size
- Total disk blocks used
- Total disk space used
- Ration of disk space to inodes

Monitoring for Inconsistent Loads

Hexis Cyber Solution recommends that you regularly query the system.upload_info table for loads with *inconsistent* data. Data is inconsistent when it is not replicated correctly across all hosts in an EDW instance. Rows with the value `false` in the CONSISTENT column identify loads with inconsistent data.

When you discover loads with inconsistent data through a query of system.upload_info, run a separate query of the system.raw_upload_info table and capture the results in a file. Then, contact Hexis Cyber Solutions Technical Support for assistance.

For more information on the system tables, see:

- “system.upload_info”, on page 168
- “system.raw_upload_info”, on page 169

HANDLING RENDEZVOUS TIMEOUTS

If you see a “rendezvous timeout” error, contact Technical Support for assistance.

IMPROVING QUERY PERFORMANCE

This section documents two options for improving query performance:

- “Generating a SQL Query Plan for Query Tuning”, next
- “Enhancing Performance for Specific TOP Queries on Large Clusters”, on page 311

Generating a SQL Query Plan for Query Tuning

A query plan, or query execution plan, describes how the EDW Query Engine plans to run a given query. This information is useful primarily to trouble-shoot a query that is running too slowly. A knowledgeable database administrator can use the plan to tune query performance.

IMPORTANT: This section describes how to generate a query plan. However, because interpreting a query plan and using it to tune performance requires sophisticated EDW Query Engine knowledge, Hexis Cyber Solutions does not expect its system administrators to interpret these plans directly. The instructions in this section are provided only to make it easier for you to generate a plan that you can deliver to Technical Support when query performance would benefit from improvement.

Sensage SQL provides the EXPLAIN extension to enable you to create a query plan. The syntax for this extension is:

```
EXPLAIN <any_select_statement>
```

When you precede the SELECT keyword with the EXPLAIN keyword for any valid query, the EDW Query Engine generates a query plan instead of actually running the query. You can generate the plan for a query that you run on the command line or from a file.

The following query example selects four columns from a table named test between May, 2008 and June, 2009. It returns only the first 40 rows:

```
SELECT top 40 Host, ClientIP, Get, Code
   FROM test
   WHERE ClientIP = '167.29.33.430'
   DURING _time('2008-05-01T10:11:00'),_time('2009-06-01T10:11:00');
```

To return the query plan rather than the data for this query, you would modify the query as follows:

```
EXPLAIN SELECT top 40 Host, ClientIP, Get, Code
   FROM test
   WHERE ClientIP = '167.29.33.430'
   DURING _time('2008-05-01T10:11:00'),_time('2009-06-01T10:11:00');
```

The query above highlights the single change in **boldface** type.

Assume you run this query directly from the command line with a command like the following:

```
atquery --user=administrator --pass=changeme --namespace=default localhost:8072\
-e "EXPLAIN SELECT top 40 Host, ClientIP, Get, Code FROM test\
WHERE ClientIP = '167.29.33.430'\
DURING _time('2008-05-01T10:11:00'),_time('2009-06-01T10:11:00');
```

The EDW engine returns the query plan as a single varchar column. The output includes rows of dashed lines. To return only the query plan without the dashed lines, add the following `atquery` flags:

```
--format=tsv --verbose=0
```

The `--verbose=0` option forces the EDW engine to return only the query plan. The `--format=tsv` option forces the varchar field to display as tab-separated values. The varchar column contains several new lines, all represented by the `\n` character.

NOTE: The tsv format is preferable to the csv (comma-separated value) or psv (pipe-separated value) formats. The varchar output that represents the query plan contains no tabs, but does contain commas and could contain pipes. Therefore, csv-formatted output would include escaped commas and psv-formatted output could include escaped pipes. The most easily readable format is tsv.

To make the query plan more readable, you can run the EDW output through the Unix `sed` (stream editor) utility. This utility reads the text input, applies formatting that you specify, and outputs the formatted text. Add the following formatting to the `sed` utility to display the query plan in text appropriately broken into separate lines:

```
sed 's/\\n/\n/g'
```

As a final step, redirect the query plan into a text file that you can share with Technical Support.

In other words, to create a readable query plan in its own text file for the example query above, you could run the following:

```
atquery --user=administrator --pass=changeme --namespace=default --format=tsv\
--verbose=0 localhost:8072 -e "EXPLAIN SELECT top 40 Host, ClientIP, Get, Code\
FROM test WHERE ClientIP = '167.29.33.430'
DURING _time('2008-05-01T10:11:00'),_time('2009-06-01T10:11:00');"
| sed 's/\\n/\n/g' > plan.out
```

Enhancing Performance for Specific TOP Queries on Large Clusters

When a query that runs on a large cluster includes both the `TOP` or `FIRST` clause and an `ORDER BY`, `GROUP BY`, or `DISTINCT` clause, the EDW Query Engine must retrieve all relevant records from all hosts in the cluster before it can sort the data and return the specified top records.

However, run the same query without an `ORDER BY`, `GROUP BY`, or `DISTINCT` clause and performance improves significantly. Because such a query requires no sorting, the EDW Query Engine stops scanning nodes and terminates the query after it retrieves the number of records specified in the `TOP` or `FIRST` clause.

IMPORTANT:

- The value you specify in the `TOP` or `FIRST` clause should exceed the number of rows returned.

In other words, a query that specifies `TOP 1000` but returns fewer than 1000 rows does not benefit from this performance recommendation.

- Performance benefits are greatest when the query runs on a multi-node cluster. Do not expect performance enhancement if you run the query on a 1-node cluster.

Setting the TopQueryFastExit Flag

Above and beyond the performance improvement gained by running a `TOP` query without sorting the data, you may also be able to improve query performance by enabling the `TopQueryFastExit` flag in the `athttpd.conf` file. This file is located in:

```
<Sensage_Home>/latest/etc/sls/instance/<instance_name>
```

Although the EDW Query Engine automatically stops processing an unordered `TOP` query as soon as it retrieves the number of rows specified in the `TOP` clause, enabling the `TopQueryFastExit` flag may further improve performance.

Set `TopQueryFastExit` to `1` to cause an unordered `TOP` query to stop processing as soon as the EDW retrieves the number of rows specified in the `TOP` clause. You must set this flag in the `athttpd.conf` file on every node in the cluster and restart the EDW for the change to take effect.

By default the flag is set to `0` and the feature is not enabled.

NOTE: Although the EDW log file contains broken pipe error messages when you enable `TopQueryFastExit`, these messages indicate no actual problem. The only issue is that they clutter the log file. If running the query without enabling this flag performs sufficiently, there is no need to enable this flag as well.

APPENDIX A

LOG FILE APPENDIX

The following sections document log files generated by HawkEye AP components. The files are grouped alphabetically by components. This section documents the following log file components:

- “Analytics Installer”, next
- “Atpgsql”, next
- “Atslapd”, on page 314
- “Collector”, on page 315
- “Command-Line Utilities”, on page 315
- “Middle Tier (Application Server)”, on page 316
- “Nearline Storage”, on page 316
- “Real-Time Component”, on page 316
- “HawkEye AP”, on page 317
- “EDW”, on page 317

ANALYTICS INSTALLER

Log File Name	Category	Purpose	Log Rotate	PTL	Notes
<user’s_home_directory>/analytics_installer.log	Debug, Monitoring	ERR MON	N	N	log file for the Analytics installer
<user’s_home_directory>/logadapter_installation.log	Debug, Monitoring	ERR MON	N	N	log file for the log adapter installer
<user’s_home_directory>/apmgr.log	Debug, Monitoring	ERR MON	N	N	log file for exporting and importing reports, dashboards, and folders

ATPGSQL

Log File Name	Category	Purpose	Log Rotate	PTL	Notes
/<*Sensage_Home*>/latest/var/log/atpgsql/db_load_schema.log	Debug, Monitoring		N		loading the database
/<*Sensage_Home*>/latest/var/log/atpgsql/db_load_schema_rootfolder.log	Debug, Monitoring		N		seeding the database
/<*Sensage_Home*>/latest/var/log/atpgsql/db_schema.log	Debug, Monitoring		N		creating the database
/<*Sensage_Home*>/latest/var/log/atpgsql/db_test_schema.result	Debug, Monitoring		N		warnings from creating the database
/<*Sensage_Home*>/latest/var/log/atpgsql/initdb.log	Debug, Monitoring		N		configuring atpgsql
/<*Sensage_Home*>/latest/var/log/atpgsql/run.log	Debug, Monitoring		N		running atpgsql
/<*Sensage_Home*>/latest/var/log/atpgsql/shutdown.log	Debug, Monitoring		N		pgsql shutdown log

ATSLAPD

Log File Name	Category	Purpose	Log Rotate	PTL	Notes
/<*Sensage_Home*>/latest/var/log/atslapd/atslapd.log	Debug, Monitoring	ERR MON	Y		ATSLAPD message **NOTE**: Rotated by syslog-ng; (see /etc/logrotate.d/syslog-ng)

Collector

Log File Name	Category	Purpose	Log Rotate	PTL	Notes
<Sensage_Home>/latest/var/log/collector/activity.log	Debug, Monitoring	AUDIT	N	Y	All actions taken by Collector during retrieve and load varies logging level in the collector config.xml **NOTE**: By default, logs to syslog; must edit config.xml to enable logging to activity.log.
<Sensage_Home>/latest/var/log/collector/error.log	Debug, Monitoring	ERR MON	N		Only the error entries from the activity log
<Sensage_Home>/latest/var/log/collector/collector.output	Debug, Monitoring		N		
	Debug, Monitoring	ERR MON	N		Stderror prints from the activity log write process
	Debug, Monitoring		N		Deprecated
<Sensage_Home>/latest/var/log/collector/transaction.log	Monitoring	AUDIT	N	Y	All collector transactions can be loaded into the EDW for self audit
	Debug, Monitoring	ERR MON	N		Parse failers for loaders. These are defined in the config.xml
<Sensage_Home>/latest/var/log/collector/windowretriever.log	Debug, Monitoring	ERR MON	Not by default; can be set to rotate		Error and debugging information for the windows retriever

Command-Line Utilities

Log File Name	Category	Purpose	Log Rotate	PTL	Notes
<Sensage_Home>/*<version>*/var/log/clsetup/clsetup_calls.log	Debug, Monitoring	AUDIT	N		Functions called by clsetup
<Sensage_Home>/*<version>*/var/log/clsetup/clsync_log.gz	Debug, Monitoring		N		gzip of clsync_log
<Sensage_Home>/*<version>*/var/log/clsetup/clssh_log.gz	Debug, Monitoring		N		gzip of clssh_log
<Sensage_Home>/*<version>*/var/log/clsetup/clsetup_log.gz	Debug, Monitoring		N		gzip of clsetup_log

MIDDLE TIER (APPLICATION SERVER)

Log File Name	Category	Purpose	Log Rotate	PTL	Notes
<*Sensage_Home*>/latest/var/log/controller/app_manager.log	Debug, Monitoring	System Monitoring	Y, daily/ app restart	N	log file for middle-tier servlets
<*Sensage_Home*>/latest/var/log/controller/app_manager_audit.log	Audit Middletier calls	Audit trail	Y, daily/ app restart	Y	web service calls with key parameters
<*Sensage_Home*>/latest/var/log/controller/controller.stderr	Debug, Monitoring		Y, Sensage restart		stdout from the controller process
<*Sensage_Home*>/latest/var/log/controller/controller.stdout	Debug, Monitoring		Y, Sensage restart		stderr from the controller process

NEARLINE STORAGE

Log File Name	Category	Purpose	Log Rotate	PTL	Notes
<*Sensage_Home*>/latest/var/log/nss/nss_stderr.log	Debug, Monitoring	ERR MON	N		Stderr for the nss (Nearline storage server)
<*Sensage_Home*>/latest/var/log/nss/nss_activity.log	Debug, Monitoring	AUDIT	N		Activity log for the Nearline storage server
<*Sensage_Home*>/latest/var/log/nss/nss_error.log	Debug, Monitoring	ERR MON	N		nss error log

REAL-TIME COMPONENT

Log File Name	Category	Purpose	Log Rotate	PTL	Notes
<*Sensage_Home*>/latest/var/log/rt/parser_rule_lal_loads.stdout	Debug, Monitoring		N		stdout from loading HawkEye AP EPL realtime rules
<*Sensage_Home*>/latest/var/log/rt/parser_rule_lal_loads.stderr	Debug, Monitoring		N		stderr from loading HawkEye AP EPL realtime rules
<*Sensage_Home*>/latest/var/log/rt/<*parser_name*>.stdout	Debug, Monitoring		N		stdout from Parser (realtime only)
<*Sensage_Home*>/latest/var/log/rt/<*parser_name*>.stderr	Debug, Monitoring		N		stderr from the Parser process (realtime only)

HawkEye AP

Log File Name	Category	Purpose	Log Rotate	PTL	Notes
<user's_desktop>/ console.log	Debug, Monitoring	ERR MON	N	N	Log file for the HawkEye AP user interface for a user session. Logs debug, informational, warning, and error messages that occur during each user's session in HawkEye AP

EDW

Log File Name	Category	Purpose	Log Rotate	PTL	Notes
/var/log/messages	Debug, Monitoring	ERR MON	Y		"EDW Errors (default location of RedHat syslog data) **NOTE**: • To capture all EDW data (debug and info), change the syslog configuration file to load all Local 0 facility data to a named file. NOTE: Rotated by logrotate; see /etc/logrotate.d/syslog
<Sensage_Home>/ *<version>*/var/log/sls/ *<instance_name>*/ athttpd_log	Debug, Monitoring	AUDIT	N		Log of httpd communications. Mostly POST traffic
<Sensage_Home>/ *<version>*/var/log/sls/ *<instance_name>*/ stdout_log	Debug, Monitoring		N		stdout statements
<Sensage_Home>/ *<version>*/var/log/sls/ *<instance_name>*/ error_log	Debug, Monitoring	ERR MON	N		errors recorded by the EDW instance

APPENDIX B

ERROR CODES

This appendix documents error messages raised when you use HawkEye AP. The messages are grouped by component and, within each component, by category and code number. This chapter documents error messages for the following components:

- “Collector”, next
- “Real Time”, next
- “HawkEye AP Console”, on page 329
- “EDW”, on page 337
- “HawkEye Retriever”, on page 346

COLLECTOR

Error Code	Message
B1000	Unable to find configuration file %s (%s)
B1001	Unable to read configuration file %s (%s)
B1002	Unable to create file '%s' (%s)
B1003	Unable to delete file '%s' (%s)
B1004	Unable to rename file '%s' to '%s' (%s)
B1005	Unable to read file '%s' (%s)
B1006	Unable to append to file '%s' (%s)
B1007	Unable to link '%s' to '%s' (%s)
B1008	Unable to read directory '%s' (%s)
B1009	Unable to fork child process (%s)
B1010	Exec of '%s' failed (%s)
B1011	Unable to create directory '%s' (%s)
B1012	Error writing stream '%s' (%s)
B1013	Error reading stream '%s' (%s)
B1014	Socketpair failed (%s)
B1015	Program exiting due to thrown exception
B1016	Can't create socket '%s'->'%s' (%s)
B1017	Can't open SMTP server '%s' (%s)
B1018	SMTP server fail on '%s'
B1019	Unable to find configuration DTD file in directory %s
B1200	Found %s nodes matching xpath='%s' expected exactly 1
B1201	DTD parse failed file='%s' error=[%s]
B1202	XML parse failed file='%s' error=[%s], backup config file may be available in '%s'
B1203	DTD validation failed during load xml='%s' dtd='%s' error=[%s], backup config file may be available in '%s'
B1204	DTD validation failed during save xml='%s' dtd='%s' error=[%s]
B1205	XML Node not found xpath='%s'
B1206	XML string parse failed error=[%s] string=[%s]
B1207	XML empty element '%s', backup config file may be available in '%s'
B1208	Expected DTD validation to fail on save, it didn't
B1209	Expected DTD validation to fail on load, it didn't
B1300	Time range Bad value '%s'
B1301	Time range expected 0-%s but got %s
B1501	Bad token '%s' as array delimiter
B1502	Bad hash name/value seperator expected ':' got '%s'
B1503	Bad token '%s' as hash delimiter
B1504	Unexpected token '%s'

Error Code	Message
B1505	Residuals tokens %s != %s
B1506	Test failed, unexpected result
B1507	Can't set more binary '%s'
B1601	Test of '%s' failed, expected throw of error code '%s' got no error
B1700	Failed to create SSL_CTX (%s) [%s]
B1701	Failed to set CTX options [%s]
B1702	Failed to create SSL (%s) [%s]
B1703	Failed to connect SSL [%s]
B1704	Failed to read private key from '%s' (%s)
B1705	Failed to read certificate from '%s' (%s)
B1706	Failed to accept SSL [%s]
B1800	Unmapped log field name(s) '%s'
B1900	Months and/or years not supported in interval (%s)
B1901	Interval %s seconds is less than %s in (%s)
B1902	Interval %s seconds is greater than %s in (%s)
C1	Unexpected Exception (%s) in file %s at line %s
C1020	Unable to create PID file %s
C1021	Unable to delete PID file %s
C1022	Unable to daemonize Collector. Exiting.
C1023	I/O Error during Controller startup (%s)
C1024	Could not stop running Collector (PID %s) during startup
C1025	Collector did not remove PID file %s upon exit
C1026	Could not find PID file %s
C1027	Could not reap PID %s after trying for %s seconds
C1100	Failed to backout load from SLS instance='%s' uploadId='%s' exitCode='%s'
C1101	Preprocess failure prog='%s' file='%s' (%s)
C1102	Load failed prior to uploadId instance='%s' file='%s'
C1103	Load failed after uploadId instance='%s' file='%s' exitValue='%s' uploadId='%s'
C1104	Parse Failure instance='%s' file='%s' uploadId='%s' line=[%s]
C1105	Unable to parse time range '%s'
C1106	Loader stopped due to fatal error text=%s
C1107	Timeout waiting for sub-process, terminating
C1108	No PTL found which matchs file name '%s'
C1109	Load failed before 'final status' instance='%s' file='%s' uploadId='%s'
C1110	Load failed instance='%s' file='%s' uploadId='%s' with 'final status... %s'
C1111	Load failed broken pipe instance='%s' file='%s' uploadId='%s'
C1112	Load failed bad exit status instance='%s' file='%s' uploadId='%s' status='%s'

Error Code	Message
C1113	Load failed exit status 10 (double TERM or INT) instance='%s' file='%s' uploadId='%s' status='%s'
C1114	Load failed exit status 15 (insufficient space on server) instance='%s' file='%s' uploadId='%s' status='%s'
C1115	Load failed exit status 20 (bad command line) instance='%s' file='%s' uploadId='%s' status='%s'
C1116	Load failed exit status 98 (failed to contact SLS) instance='%s' file='%s' uploadId='%s' status='%s'
C1117	Load failed exit status 99 (internal error) instance='%s' file='%s' uploadId='%s' status='%s'
C1118	Tar for daisy chain returned bad exit value '%s'
C1119	Webui instance '%s' has no config.dat file
C1120	Bad or missing meta data file '%s'
C1121	Loader '%s' already running ($!)
C1400	Retriever stopped due to fatal error name=%s text=%s
C1401	Preprocessor exited with bad status on preprocessor='%s' file='%s' status='%s'
C1402	No retriever type named '%s'
C1403	Backup to '%s' failed status (%s), will retry in '%s' seconds.
C1404	Bad backup url '%s'
C1405	Can't load .meta file '%s' (%s)
C1406	Retriever name must contain alpha numeric and '_' '-' characters only '%s'
C1407	Bad exit status for md5sum command (%s)
C1408	No output from md5sum command
C1409	Bad output from md5sum command (%s)
C1410	Interval '%s' is less than 1 second
C1411	Months and years not supported in interval '%s'
C1412	FTP connection to host '%s' failed to put file '%s'
C1413	Size mismatch after download of '%s', %s != %s
C1414	No LogQueue specified for retriever '%s'
C1415	Retriever '%s' already running ($!)
C1416	Preprocessor failed to generate output file preprocessor='%s' file='%s'
C1417	Gap in queue probability coverage at '%s' in retriever '%s'
C1418	Unexpected failure to place file '%s' in queue
C1419	Low and high attribute on LogQueue must be from 0-99 in retriever '%s'
C1420	Illegal retriever name '%s'
C1500	FTP connection to host '%s' failed with message '%s'
C1501	FTP connection to host '%s' failed on login to '%s'
C1502	FTP connection to host '%s' failed on cwd to '%s'
C1503	FTP connection to host '%s' failed on data transfer

Error Code	Message
C1504	FTP connection to host '%s' failed to delete file '%s'
C1505	FTP connection to host '%s' failed to create file '%s'
C1506	FTP connection to host '%s' failed on nlist
C1507	FTP connection to host '%s' failed on pwd
C1508	FTP connection to host '%s' failed on get '%s'
C1509	FTP connection to host '%s' failed to set mode binary
C1600	SFTP connection to '%s' failed (%s)
C1601	SFTP ls of '%s' failed (%s)
C1602	SFTP get of '%s' failed (%s)
C1603	SFTP rm of '%s' failed (%s)
C1604	SFTP create of '%s' failed (%s)
C1605	SFTP connection lost (%s)
C1606	SFTP bad server version (%s)
C1607	SFTP sftp client exit pid: %s, exit code: %s
C1608	SFTP put of '%s' failed (%s)
C1609	SFTP unparseable line from ls '%s'
C1610	SFTP unable to find ssh in path
C1611	SFTP unable to cd to '%s'
C1612	SFTP unable to parse directory line '%s'
C1700	Bad request '%s' only PUT supported
C1701	Bad request '%s' expect HTTP/1.x
C1702	Can't parse request '%s'
C1703	Can't listen on '%s'
C1704	Missing Content-Length: header on '%s'
C1705	Oversized body %s > %s
C1706	Premature socket EOF or error '%s' in %s
C1707	Can't parse file record '%s', on '%s'
C1708	User/passwd required and not provided
C1801	Can't parse server status line '%s'
C1802	Unexpected EOF in server data
C1803	Server reports error (%s %s %s)
C1804	Unable to parse XML reply from server (%s)
C1805	Server reports error detail (%s)
C1806	Failed to load SSL libraries (%s)
C2000	Could not start Controller
C2001	Could not create Log Queue at %s
C2002	Could not read Log Queue %s. Removing from rotation.
C2003	Could not write Log Queue %s. Removing from rotation.
C2004	Could not execute Log Queue %s. Removing from rotation.

Error Code	Message
C2005	%s MB free space on %s Log Queue is below minimum %s MB. Disabling %s.
C2006	Could not disable Log Queue %s.
C2007	Could not enable Log Queue %s
C2008	Could not find PID of group %s in state. Could not stop it.
C2009	Could not start Loader %s because it is using the same Log Queue %s as Loader %s
C2010	User %s does not exist. Exiting.
C2011	No group id for user %s. Exiting.
C2012	Failed to change UID to %s (now: %s, %s)
C2013	Failed to change GID to %s (now: %s, %s)
C2014	Log Queue Root dir %s does not exist
C2015	Collector/FileRoot is not defined
C2016	FileRoot is not configured correctly. %s does not exist.
C2017	Collector/StateRoot is not defined
C2018	StateRoot is not configured correctly. %s does not exist.
C2019	A Shepherd has disappeared and will not be respawned. May have exited on error or been killed. (name: %s)
C2020	A Loader has disappeared and will not be respawned. May have exited on error or been killed. (name: %s)
C2100	Could not open dashboard file %s (%s)
C2200	Could not create state directory %s
C2201	General error when creating state dir (%s)
C2202	Could not instantiate DB File %s
C2203	Could not optimize statefile %s
C2301	SCP No SourceFile entries specified for '%s'
C2302	SCP non zero exit value from scp downloading '%s'
C2303	SCP no files retrieved '%s'
C2304	SCP STDERR: %s
C2400	WWW IP for host '%s' not found
C2401	WWW Failed to retrieve '%s'
C2402	WWW Body size mismatch Content-Length: %s != downloaded: %s
C2403	WWW No content length in headers
C2404	WWW Can't load SSLStream, open SSL possibly not installed? (%s)
C2405	WWW Unparseable status line from server (%s)
C2406	WWW bad status from server (%s)
C2501	Can't use Schedule and Period at the same time in '%s'
C2601	Host Glob of SourceHost '%s' results in 0 results
C2701	Daisy Chain untar of '%s' to '%s' failed exit code '%s'
C2702	Daisy Chain meta and logfile pair are bad, log: '%s' meta: '%s'

Error Code	Message
C2703	Daisy Chain Got '%s' files in tar file, expected 2
C2801	LEA Can't parse log line (%s) from '%s'
C2802	LEA return code: %s
C2803	LEA debug output: %s
C2804	LEA unsafe cma name '%s'
C2805	LEA Can't parse log line (%s)
C2806	LEA bad mode (%s)
C2900	Unable to find template variable '%s' in replacement hash
C2901	No such test '%s'
C3000	Unable to connect to '%s' ('%s')
C3001	Query failed: %s (SQL: %s)
C3002	Database fetch failed: %s
C3003	No state value for column '%s', and no default exists
C3004	No StateCol definition for parameter '%s'
C3005	No SQL query defined
C3006	State Column '%s' is missing from the SQL query
C3007	Received result type '%d', expected 4040

REAL TIME

Message
%d seconds have elapsed and %d rules have been received
%d seconds have elapsed and no rules have been received
: rtnet error %s\n
ASSERT() failed
Bad Cast attempted
Bad Map - no value following = in %s
Bad RPC size %u
Bad configuration file %s: %s”,”/etc/sensage/parserRules.config
Bad map - no = char in %s
Bad map - no string preceding = in %s
Bad month name string ‘%s’ in message ‘%s’ using current time
CTCPListen: Failed to bind socket: %d %d%s
CTCPSTream: connect to %s failed: %d %d%s
Can’t create properties file: %s
Can’t create rule cache ‘%s’
Can’t create scratch file: %s
Can’t open input file: %s
Can’t open load file: %s
Can’t read cached rule file ‘%s’
Cannot open %s\n”,”/etc/sensage/parserRules.config
Cannot open file ‘parserHosts.cache’ for dumping the hosts cache.
Cannot open hosts file ‘%s’
Cannot stat IP file ‘%s’
Config too large: %s\n”,”/etc/sensage/parserRules.config
Conflicting page size sysconf=%ld, getpagesize=%d
Could not obtain PID
Decode done but NULL LRWorld
Decode of message didn’t use the expect amount
Dispatcher: Event %d not deregistered
END TEST: %s - %s EXCEPTION=%s
END TEST: %s - %s EXCEPTION=?
EOF on input channel with no rules loaded, unable to process pending data
ERROR: Rule missing signature
ERROR: Rules code not compatible -- check compiler versions
Error Rule file: %s - %s
Error Rule file: %s - Sensage EPL rule send by Controller when RuleLoader is configured

Message
Exception interrupted persist attempt, trying again
Expected a numeric field, got a string field: %s
Expected a string field, got a numeric field: %s
FSMs (%llu+%d) exhausted -- dropping event, rule %s
Fatal: %s (%s)
ITCPConnect: Failed to bind socket: %d %d%s
ITCPConnect: Failed to shutdown write: %d
Ignoring corrupt ptl file: %s
Ignoring corrupt upload file: %s
Ignoring row for just completed block: %s
Ignoring upload message for completed block: %s
LARGEFILE support not compiled in
Lost Sensage EPL sequence got %u expected %u, dropping rest of stream
Module %d:%s tries to register twice, original=%d.%d.%d.%d:%u, new=%d.%d.%d.%d:%u
Module ‘%s’ clock is off by more than 5 minutes
Module ‘%s’ deregistering (leaving) the system
Node %s closed network connection
Node %s, no heartbeat in %u sec
Odd page size %ld vs %d
PCRE compilation failed at offset %d: %s\n
Packet length %u greater than max %llu
Parser shutdown while data left to process
Persisted data embedded version mismatch %llu vs. %llu
Persistor replied with NAK on data version=%llu
Rename %s->%s failed
Rule error: %s with rule %s
Rule load failure: %s
Shutting down socket: %d due to RT exception
Shutting down socket: %d due to std::exception: %s
Shutting down socket: %d due to unknown exception
Sort skipped due to low disk
Test doesn’t exist: %s
ThreadEnd due to uncaught RTException: %s
ThreadEnd due to uncaught std::exception: %s
ThreadEnd due to uncaught unknown exception
Too many rules (%llu)

Message
Two minutes have passed and the parser is still waiting for rules from: %s%s
Unable to delete expired file '%s'
Uncaught RTException during persisted action
Uncaught RTException during rtnet_receive
Uncaught std::exception during persisted action: %s
Uncaught std::exception during rtnet_receive: %s
Uncaught unknown exception during persisted action
Uncaught unknown exception during rtnet_receive
Unexpected exception in rule compile: %s
Unhandled deferred message: %s
Write failed in flush buffer to '%s' amt=%d
Write to file failed: %s
\tFailed to read %d bytes from %s: %s
\tFile %s too large: %d vs %d
\tOpen %s failed: %s
\tRule load from %s failure: %s
\tRulename = %s
\tRulenum = %d
\tStatenum = %d
atload failed: table='%s', message=%s
bogus data on socket can't decode, closing: %d [%s]
can't decode message because size: %u > %u
commitToNet commit timed out
delete ruleChain failed: %s
empty atload failed: table=%s
execv() failed
lock contention on SPtr2
malloc(%llu) Arena failed %d (%s)
mlock -- memory lock of %llu failed %d (%s)
netctrl: Could not obtain PID
netctrl: couldn't send die command %d/%s
netctrl: couldn't send to node %s
netctrl: rtnet error (%d) %s\n
pcre error %d on %s s(for %s %s)\n
preprocess failed status=%d, block: %s
preprocess failed, no output was produced, block: %s
read error on socket: %d
read error on socket: %d [%s]

Message
read nvram iErr = %d, version #: %llu
read zero bytes from %s\n”,”/etc/sensage/parserRules.config
rtnet vendor mismatch dropping connection: %s/%s doesn’t match %s/%s
rtnet(%d) error %s
rtnet: bad vendor name %s
rule compile failure: %s
rv=%d
setPopulateConst: no event field ‘%s’
sort failed, skipping straight to load status=%d, block: %s
thread not running: %s
waitpid error pid=%d
write error on socket: %d
write error on socket: %d [%s]
wrote nvram iErr = %d, version #: %llu (%u-%u)

HawkEye AP Console

This section documents error messages raised when you use HawkEye AP Console:

- “General Errors”, next
- “Security Errors”, on page 330
- “Dashboard Errors”, on page 331
- “Report Errors”, on page 332
- “Schedule Errors”, on page 334
- “User Errors”, on page 335
- “Journal Errors”, on page 336
- “Role Errors”, on page 336

General Errors

Error codes for General Errors, which range between 1000 - 1999, are listed below.

Error Code	Message
1000	Error creating exception
1001	Illegal argument, session ID may not be null
1002	Error getting current session
1003	Error begin service call failure
1004	Error committing service call
1005	Error ending service call
1006	Null object type not allowed
1007	Class not found

Error Code	Message
1008	Unable to access database
1009	User not logged in
1010	Login denied for user: {0}
1011	Connection failed for user:{0}@{1}:{2}/{3}
1012	Secure connection failed for user:{0}@{1}:{2}/sls
1013	User session has expired
1014	Caused by:
1015	Unable to connect to {0}
1016	Database connection lost
1017	Properties file must contain either shared secret or username/password
1018	Not implemented

Security Errors

Error codes for Security Errors, which range between 2000 - 2999, are listed below.

Error Code	Message
2000	Access denied
2001	Unable to set role permission
2002	Unable to check permission
2003	Unable to get permission
2004	Unable to get user permissions on object
2005	Unable to clear role permission
2006	Unable to get role permissions
2007	Unable to convert action to a permission
2008	Unable to convert permission to actions
2009	Unable to set role permissions on object {0}
2010	Unable to add permission
2011	Unable to remove permission
2012	Unable to add permission to role
2013	Unable to remove permission from role
2014	Unable to enable role {0}
2015	Unable to disable role {0}
2016	Permission ID: {0} Not found in SLS
2017	Unable to get a role's permissions

Dashboard Errors

Error codes for Dashboard Errors, which range between 3000 - 2999, are listed below

Error Code	Message
.3000	Create dashboard folder failed
3001	Create dashboard failed
3002	Create dashboard page failed
3003	Create dashboard widget failed
3004	get dashboard items failed
3005	get dashboard folder {0} failed
3006	get dashboard {0} failed
3007	get dashboard page {0} failed
3008	get dashboard widget {0} failed
3009	get dashboard folder hierarchy failed
3010	Update of dashboard folder {0} failed
3011	Update of dashboard {0} failed
3012	Update of dashboard page {0} failed
3013	Update of dashboard widget {0} failed
3014	Delete of dashboard folder {0} failed
3015	Delete of dashboard {0} failed
3016	Delete of dashboard page {0} failed
3017	Delete of dashboard widget {0} failed
3018	Illegal argument, dashboard folder ID may not be null
3019	Illegal argument, dashboard folder may not be null
3020	Illegal argument, dashboard ID may not be null
3021	Illegal argument, dashboard may not be null
3022	Illegal argument, dashboard page ID may not be null
3023	Illegal argument, dashboard page may not be null
3024	Illegal argument, dashboard widget ID may not be null
3025	Illegal argument, dashboard widget may not be null
3026	Unexpected item in dashboard folder, Item Name: {0}, Type: {1}
3027	Cannot get report for dashboard with ID: {0}
3028	Error getting history for dashboard: {0}
3029	Errors while emailing dashboard
3030	Run of dashboard not found
3031	Delete of dashboard folder {0} failed, folder is not empty
3032	Illegal argument, dashboard item name may not be null
3033	Illegal Arguments, dashboard parameter does not contain the page parameter
3034	Get Dashboard Item {0} Failed

Error Code	Message
3035	Illegal Argument, Report Id may not be null
3036	Unable to determine if the Report is used
3037	Unable to get the Dashboards using the Report
3038	Unable to move dashboard item

Report Errors

Error codes for Report Errors, which range between 4000 - 4999, are listed below

Error Code	Message
4000	Unable to get reports
4001	Unable to get report with ID: {0}
4002	Unable to create report definition
4003	Update of report with ID: {0} failed
4004	Unable to delete report with ID: {0}
4005	Illegal argument, report may not be null
4006	Unable to get report roles
4007	Unable to get report with ID: {0} Not found
4008	Unable to create report folder
4009	Unable to get report folder with ID: {0}
4010	Unable to get report folder with ID: {0} Not found
4011	Update of report folder with ID: {0} failed
4012	Unable to delete report folder with ID: {0}
4013	Unable to create report link
4014	Update of report link with ID: {0} failed
4015	Unable to delete report link with ID: {0}
4016	Unable to run report
4017	Unable to get namespaces
4018	Unable to get tables
4019	Unable to get columns
4020	Unable to get report items
4021	Unable to get report item with ID: {0}
4022	Report must have a valid id and name
4023	ID: {0} was not found or is neither a report folder nor report link
4024	Unable to run report, report does not exist for ID: {0}
4025	Unable to run report with ID: {0}, Date range unavailable
4026	Unable to run report with ID: {0} Namespace unavailable
4027	Unable to run report with ID: {0}, Namespace is Null
4028	Cannot update metadata for report with ID: {0}
4029	Report folder ID:{0} - Name: {1} does not match name: {2}

Error Code	Message
4030	Can't get report from link
4031	Unexpected item in report folder, Item name: {0}, Type: {1}
4032	Invalid date range used to run report with ID: {0}
4033	A date must be specified before which cache entries will be deleted
4034	Deleting report cache entries failed for report with ID: {0}
4035	Unable to fetch report cache entries for report with ID: {0}
4036	Unable to fetch list of running queries
4037	Unable to fetch cache data for report with ID: {0}
4038	Unable to create library
4039	Unable to delete library with ID: {0}
4040	Unable to update library with ID: {0}
4041	Unable to get library with ID: {0}
4042	Unable to get libraries
4043	Unable to link library {0} to report {1}
4044	Unable to remove library link {0}
4045	Unable to get libraries for library with ID: {0}
4046	Error parsing multiple response when getting libraries
4047	Error parsing multiple response when getting report definitions
4048	Unable to link library {0} to library {1}
4049	Unable to move library link {0}
4050	Library is in use - unable to delete
4051	Errors while emailing report
4052	Unable to create or rename library - name must be unique
4053	Unable to fetch objects
4054	Unable to get drilldown reports for report {0}
4055	Unable to add drilldown report {0} to report {1}
4056	Unable to remove drilldown report {0} from report {1}
4057	Unable to create display object for report {0}
4058	Unable to get display object {0}
4059	Unable to get display objects
4060	Unable to delete display object {0}
4061	Unable to delete display objects for report {0}
4062	Unable to update display object {0}
4063	Unable to move object {0}
4064	Unable to parse multiple responses while getting disk usage
4065	Unable to get metadata for object with ID: {0}
4066	Unable to delete a report that is in use
4067	Illegal Argument, Library may not be null
4068	Unable to Add Alert Column {0} to Drilldown Report {1}

Error Code	Message
4069	Unable to Remove Alert Column {0} from Drilldown Report {1}
4070	Unable to Get Drilldown Reports for Alert Column {0}
4071	Invalid time interval used to run report with ID: {0}
4072	Unable to get cache entry {0}
4073	For cache entry with ID {0}, unable to cast column named {1} to type {2}
4074	Unable to get display object for report ID: {0}
4075	Unable to complete operation - a circular reference would result
4076	Report run failed ({0})
4077	Unable to get report name ({0})
4078	Unable to create report link ({0})
4079	Unable to create folder ({0})
4080	Unable to create report definition ({0})
4081	Unable to delete report definition ({0})
4082	Unable to create library ({0})
4083	Unable to delete one or more cache entries
4084	An error occurred while getting cache data (cause : {0})
4085	An invalid type was specified for casting
4086	Unable to parse multiple responses while getting status
4999	Data is not yet available

Schedule Errors

Error codes for Schedule Errors, which range between 5000 - 5999, are listed below.

Error Code	Message
5000	Error getting schedules status
5001	Error creating schedule
5002	Unable to get schedule with ID: {0}
5003	Error getting schedule
5004	Error updating schedule
5005	Error deleting schedule
5006	Database error while deleting schedule
5007	Name: {0} already used
5008	Incomplete schedule definition: {0}
5009	Cannot run schedule, it is already enabled
5010	Start date not set for backfill
5011	Error running schedule
5012	Cannot change schedule, someone else has changed it
5013	Cannot find class for job Type: {0}

Error Code	Message
5014	Parse error with cron expression {0}
5015	Illegal argument to scheduler {0}
5016	Schedule rejected with cause {0}
5017	Scheduler exception when deleting job {0}
5018	Exception creating file {0}
5019	Exception exporting report {0}
5020	Exception exporting dashboard {0}
5021	Exception writing file {0}
5022	Email not configured
5023	Errors sending email: {0}
5024	Cannot backfull running schedule
5025	Schedule {0} not found
5026	Cannot change schedule while it is running
5027	The Schedule Manager is not ready
5028	Enable schedule failed {0}
5029	Disable schedule failed {0}

User Errors

Error codes for User Errors, which range between 6000 - 6999, are listed below.

Error Code	Message
6000	Unable to get users
6001	Unable to get user with ID: {0}
6002	Unable to create user
6003	Update of user with ID: {0} failed
6004	Unable to delete user {0}
6005	Illegal argument, user may not be null
6006	Unable to get a user's roles
6007	User ID: {0} Not found in SLS
6008	Illegal argument, user preference may not be null
6009	Only one user may have their preferences set at a time.
6010	Unable to add user preferences to user with ID: {0}
6011	Unable to remove user preferences from user with ID: {0}
6012	Unable to get user preference for user with ID: {0}
6013	Unable to disable user with ID: {0}
6014	Unable to enable user with ID: {0}
6015	Unable to change password for user with ID: {0}
6016	Syncing of User/Role Information from SLS failed: {0}
6017	Error updating an user: {0}

Error Code	Message
6018	Unable to change password, old password incorrect

Journal Errors

Error codes for Journal Errors, which range between 7000 - 7999, are listed below.

Error Code	Message
7000	Illegal argument, journal entries may not be null
7001	Execute journal failed, invalid reference name
7002	Execute journal failed, invalid journal service call
7003	Execute journal failure, method "{0}" not found with {1} parameter(s)
7004	Execute journal failed, illegal access
7005	Execute journal failed, invocation error
7006	Execute journal failed, empty service constructor not found
7007	Execute journal failed, error constructing service

Role Errors

Error codes for Role Errors, which range between 8000 - 8999, are listed below.

Error Code	Message
8000	Unable to get roles
8001	Unable to get role with ID: {0}
8002	Unable to create role
8003	Update of role with ID: {0} failed
8004	Unable to delete role {0}
8005	Illegal argument, role may not be null
8006	Illegal argument, role ID may not be null
8007	Role ID: {0} Not found in SLS

User/Role Errors

Error codes for User/Role Errors, which range between 9000 - 9999, are listed below.

Error Code	Message
9000	Error Removing Users From Role: {0}
9001	Error Adding Users To Role: {0}

Dashboard Data Access Object Errors

Error codes for Dashboard DAO Errors, which range between 10000 - 10999, are listed below

Error Code	Message
10001	Move dashboard item failed, item not in source folder
10002	A dashboard item with this name already exists: {0}
10003	Moving a folder to one of its sub folders not allowed

Report Data Access Object Errors

Error codes for Report DAO Errors, which range between 11000 - 11999, are listed below

Error Code	Message
11001	Unable to get Associated Report with id: {0}
11002	Unable to get Report Display Object with id: {0}

EDW

The errors listed below represent approximately one quarter of the exceptions issued by the EDW.

Error Code	Message
0010004	Shim not implemented
0010012	FILE* is NULL
0010014	Parameter name is NULL
0010019	FILE* for output is NULL
0010020	FILE* for output is NULL
0010022	malformed bucket chain in Tcl_DeleteHashEntry
0010023	called Tcl_FindHashEntry on deleted table
0010024	Name is NULL
0010025	Value is NULL
0010026	called Tcl_CreateHashEntry on deleted table
0010042	Attempting to process data with no schema
0010043	Column Index out of bounds
0010053	Unknown P_UNIT_T
0010057	Unknown Data Type: $1
0010058	Internal Logic Error: Please contact the Development Team
0010062	HTTP data ended before complete content was received
0010063	Stream has received data of an unknown type
0010064	HTTP data ended before complete HTTP header was received
0010066	Errors while reading file

Error Code	Message
0010067	Unknown Data Type: $1
0010071	Problems compiling statement
0010072	pullDataSource should never be called for CPushedValueStream
0010073	‘processHttpData’ called after ‘getLeftOverData’
0010074	Shim not implemented
0010088	EOS has already been sent
0010089	EOS has already been sent
0010091	EOS has already been sent
0010093	EOS has already been sent
0010094	EOS has already been sent
0010098	Shim not implemented
0010102	Child Process Died
0010124	Internal Error: Unknown parsing state
0010134	Failed to write all the Sort Data
0010140	Multiple Schema Specifiers in Stream
0010141	Incorrect Data Type Requested
0010142	Unknown Internal State
0010158	NULL Byte found in InitialData
0010159	NULL Byte found in InitialData
0010168	Unexpected error code: $1
0010176	Unknown STATE Code
0010178	Invalid !DOCTYPE block
0010179	Invalid !ELEMENT block
0010184	Unknown Parameter Type in Table
0010186	Argument is not an XMLRPC Fault Response
0010187	Argument is not a standardized XMLRPC Fault Response
0010210	No Such Channel: $1
0010216	Invalid length on hexadecimal string: $1
0010217	Non-hexadecimal digit: $1
0010230	Non-hexadecimal digit: $1
0010241	Too many fields in input data
0010242	Too few fields in input data
0010243	Multiple !DOCTYPE’s in document
0010252	Corrupted BZIP2 data stream
0010260	Unknown Data Type Code
0010261	XMLRPC_BINARY Not Implemented Yet
0010262	No Regex found
0010265	Shim not implemented
0010268	Shim not implemented

Error Code	Message
0010274	Couldn’t find chunk-size in stdin stream
0010284	Invalid Regex Type: $1
0010286	All Network Communication Processes have died
0010290	Internal Error: Received incomplete rows
0010292	‘parseData’ called after ‘getLeftOverData’
0010333	Method should never be called
0010338	‘start’ called on an uninitialized object
0010339	‘init’ called on a non-empty object
0010340	Shim not implemented
0010347	Internal Error: Stream Iter is Empty
0010353	Stream Iter is Empty
0010355	Stream Iter is Empty
0010361	Shim not implemented
0010375	Invalid Regex Type: $1
0010383	Shim not implemented
0010385	Shim not implemented
0010386	Empty Passwords are not allowed
0010398	Unknown Permission Type: $1
0010399	The ‘$1’ role cannot be deleted
0010400	The ‘$1’ user cannot be deleted
0010422	Index out of bounds
0010427	The ‘$1’ user cannot be deleted
0010453	Read did not fetch an integral number of records
0010455	Encountered EOF while reading Data File
0010456	Unable to read complete row from Data File
0010459	Internal Logic Error: Please contact the Development Team
0010467	The ‘$1’ user cannot be disabled
0010468	The ‘$1’ role cannot be disabled
0010469	The ‘$1’ user cannot be removed from the ‘$2’ role
0010470	The ‘$1’ role cannot be deleted
0010478	Internal Error: Child sent a METADATA packet
0010479	Internal Error: Child sent a SCHEMA packet
0010497	The ‘$1’ permission cannot be removed from the ‘$2’ role
0010498	The ‘$1’ permission cannot be removed from the ‘$2’ role
0010499	Unable to send WAIT message to Application Server
0010526	Internal Logic Error: Please contact the Development Team
0010528	Problems compiling statement
0010532	Invalid SKIP-QUEUE Credentials supplied
0010547	Invalid language/country specifier for locale: $1/$2

Error Code	Message
0010548	Invalid UTF-8 Encoding in ‘$1’ at position $2
0010549	Incomplete UTF-8 Encoding in ‘$1’ at position $2
0010550	Invalid UTF-8 Encoding in ‘$1’ at position $2
0010551	Incomplete UTF-8 Encoding in ‘$1’ at position $2
0010556	Invalid UTF-8 Encoding in ‘$1’ at position $2
0010557	Incomplete UTF-8 Encoding in ‘$1’ at position $2
0010558	Invalid UTF-8 Encoding in ‘$1’ at position $2
0010561	Incomplete UTF-8 Encoding in ‘$1’ at position $2
0010562	Invalid UTF-8 Encoding in ‘$1’ at position $2
0010563	Incomplete UTF-8 Encoding in ‘$1’ at position $2
0010564	Index Out of Bounds: $1 < 0
0010565	Invalid UTF-8 Encoding in ‘$1’ at position $2
0010566	Incomplete UTF-8 Encoding in ‘$1’ at position $2
0010567	Index Out of Bounds: $1 > $2
0010568	Invalid UTF-8 Encoding in ‘$1’ at position $2
0010569	Invalid UTF-8 Encoding in ‘$1’ at position $2
0010570	Incomplete UTF-8 Encoding in ‘$1’ at position $2
0010571	Internal Error: Incomplete ROLES Definition
0010574	Invalid UTF-8 Encoding in ‘$1’ at position $2
0010575	Incomplete UTF-8 Encoding in ‘$1’ at position $2
0010587	Internal Error: Incomplete row
0010619	Parameter $1 is not a constant string
0010708	Failure while reading from ‘$1’
0010713	Invalid Regex Type: $1
0010715	Invalid Regex Type: $1
0010717	Could not open shared secret file
0020033	Invalid table name
0020050	columns array requires at least one column
0020053	Problems compiling statement
0020057	internal error: identifier not compiled
0020069	Must have either/or sql/plan request
0020095	Not a create-table command
0020103	Incorrect Data Type Requested
0020108	Expected non-zero-length field name
0020109	Expected non-zero-length field name (2)
0020118	Expected XMLRPC Structure
0020129	Expected non-zero-length field name
0020130	Expected non-zero-length field name
0020135	Query plan not parsable

Error Code	Message
0020140	Bad partition column node
0020141	Expected partition column name
0020143	Execution plan is NULL
0020144	For now we must have partition column
0020146	Didn’t find any incoming links
0020147	Don’t know muxer type
0020154	Execution plan is NULL
0020173	OOB isn’t a failure OOB
0020183	Unexpected out-of-space
0020222	Expected to catch exception
0020224	Expected to catch exception (2)
0020225	Too many value streams
0020226	Bad data type
0020241	Missing +/- on sorting column
0020251	Unexpected domain
0020255	Non-addamark exception caught
0020256	Cannot use function in this case
0020260	Cannot use function in this case
0020268	Bad state
0020273	Can’t send OOB onto stream
0020277	Can’t shutdown output
0020302	Muxer not currently sending data
0020322	Setting up slave nodes again
0020345	Pushing column without values
0020347	Metadata not sent yet
0020365	No clients
0020366	Index out of range
0020367	Request already set
0020368	Index out of range
0020369	Index out of range
0020370	Muxer already set
0020371	Index out of range
0020372	Muxer not currently sending data
0020376	Expecting one client only
0020377	Expecting one client only
0020394	Problems compiling statement
0020903	Expected to catch exception (2)
0020908	Expected to catch exception
0020981	Expecting one client only

Error Code	Message
0020983	Expecting one client only
0020989	Expecting one client only
0025001	Non-null fatalFunc
0030001	Index out of bounds
0030007	Perl tried to die!
0030010	Invalid data found in INT32 field
0030011	Integer overflow/underflow
0030012	Invalid data found in DOUBLE field
0030013	Floating Point overflow/underflow
0030014	Invalid data found in BOOLEAN field
0030015	BASE64 not implemented
0030016	CDATA found in non-scalar data type
0030017	Unknown XML-RPC Datatype
0030018	Incorrect Data Type Requested
0030019	Incorrect Data Type Requested
0030020	Incorrect Data Type Requested
0030021	Incorrect Data Type Requested
0030023	Incorrect Data Type Requested
0030024	Incorrect Data Type Requested
0030026	Incorrect Data Type Requested
0030028	The ETL ‘SELECT’ Statement must be FROM ‘stdin’
0030046	Internal Error: Unknown XMLRPC Parsing State
0030048	Bad XMLRPC: No
0030049	Internal Error: Unexpected State Stack
0030050	Bad XMLRPC: No
0030051	Internal Error: Unknown XMLRPC Parsing State
0030052	Bad XMLRPC: Unexpected CDATA found
0030058	Internal Error: State Stack is empty
0030059	Internal Error: Index out of bounds
0030067	Unexpected XML tag (was expecting ‘param’)
0030068	Unexpected XML tag (was expecting ‘value’)
0030070	Unexpected XML tag (was expecting ‘member’)
0030074	Unexpected XML tag (was expecting ‘data’)
0030075	Found two ‘data’ tags for the same array
0030076	Unexpected XML tag (was expecting ‘value’)
0030077	Unexpected XML tag (was expecting ‘value’)
0030078	Unexpected XML tag (wasn’t expecting any tags)
0030079	Unknown XMLRPC Parsing State
0030080	Unexpected end-tag

Error Code	Message
0030082	Unexpected State Stack
0030083	Missing required subparts for ‘member’ block
0030084	Unknown XMLRPC Parsing State
0030085	Unexpected CDATA found
0030091	State Stack is empty
0030092	Index out of bounds
0030097	!IS_OP
0030108	Regex not compiled
0030118	!IS_CONST
0030128	Bad XMLRPC: No the
0030129	Bad XMLRPC: No the
0030136	Ran out of data
0030143	Unknown DataType
0030149	!IS_OP
0030150	!IS_CONST
0030159	Perl interpreter not initialized
0030171	Unknown DataType
0030173	Cannot open file
0030216	Invalid ORIENTATION code
0030218	Unknown FORMAT code
0030220	Unknown FORMAT code
0030221	XML Encoding not implemented for COLUMN orientation
0030223	Unknown FORMAT code
0030224	BINARY Encoding not implemented for ROW orientation
0030225	Invalid FORMAT code
0030226	Invalid SQLTYP code
0030228	GZIP Compression not implemented
0030229	Unknown COMPRESSION code
0030450	Unimplemented Data Type
0030464	invalid type code
0030563	Attempt to consume more data than exists
0030585	Bad state, cannot send message
0030593	Application server is NULL
0030650	Content type is NULL
0030668	Internal Error: Unable to read MemTotal from meminfo
0030669	Internal Error: Unable to read MemFree from meminfo
0030670	Internal Error: Unable to read Cached from meminfo
0030671	Internal Error: Unable to read Buffers from meminfo
0030707	No manager

Error Code	Message
0030708	No manager
0030709	No manager
0030710	No manager
0030711	Consuming too much or negative
0030712	Weird state
0030713	Unconsuming negative bytes
0030719	Not first vs. first
0030720	Should not be called when no buffer
0030721	Already managed
0030984	!IS_OP
0030985	!IS_CONST
0030993	Only select from stdin allowed
0030994	Problems compiling statement
0031012	!IS_OP
0031014	!IS_CONST
0031050	name must not be null
0031051	value must not be null
0031067	pullDataSource should never be called
0031102	order info may not be NULL
0031160	No Channels
0031341	val == NULL
0031413	Incorrect Data Type Requested
0031417	Regex not compiled
0031418	Regex not compiled
0031471	ORDER by not allowed
0031473	UNION not allowed
0031586	Internal Error: Incomplete row
0031587	Internal Error: Incomplete row
0040001	first argument must be a constant
0060012	Bad channel index
0070011	Empty UserNames are Invalid
0070024	Could not open file $1 for reading
0070025	Out of memory
0070026	Guest role entry not found
0070036	Administrator role entry not found
0070049	Empty role names are invalid
0070051	Empty Permission Names are invalid
0070067	Could not open shared secret file
0070072	Out of memory

Error Code	Message
0070091	Administrator user entry not found
0070092	Guest user entry not found
0070110	Cannot delete built-in permission ‘$1’
0070203	Out of memory
0080002	Unix socket file name too long: $1
0080004	No near line storage socket specified
0080005	Bad reply from near line server
0080006	Got reply from server before Hello
0080008	Bad hello response from server
0080009	Bad StartTime
0080010	Bad EndTime
0080013	No matching pending store request
0080014	Internal Logic Error: Please contact the Development Team
0080018	Problems compiling statement
0080021	Connection to Near Line Server lost during setNSI command
0080023	Connection to nearline storage server lost
0080024	Failed to complete near line storage server operations
0080036	The ETL ‘SELECT’ Statement must be FROM ‘stdin’
0080038	Only select from stdin allowed
0080039	Problems compiling statement
0080186	No table specified
0080199	Bad keytype
0080207	Bad string format
0080221	Attempt to write read only DB file ‘$1’
0080228	Attempt to write read only DB file ‘$1’
0080229	Bad operation ‘$1’
0080233	UPLOADID must be first CSV column
0080234	gzopen failure
0080237	No table specified
0080238	Auth type mismatch $1 vs. $2
0080239	$1 was invoked on multiple requests
0080242	Shim not implemented
0080243	Shim not implemented
0080244	Shim not implemented
0080245	Shim not implemented
0080246	No active master node(s) found for query
0080253	Unexpected EOF talking to Atalla at ‘$1’
0080267	Bad response code to LA GUID request ‘$1’
0080268	LA GUID cound mismatch in reply ‘$1’

Error Code	Message
0080277	Attempt to sign digest failed
0080278	Attempt to get binary value of NULL
0080286	Can’t find host ‘$1’
0080296	Can’t find upload ID entry for $1
0080297	TASCMConfig should be of the form LAIP:PORT/MGRIP:PORT
0090022	Error querying system.upload_info table
0090023	For table ‘$1’ upload id not found: ‘$2’

HawkEye Retriever

HawkEye Retriever displays messages without error codes. The table below lists the full set of messages.

Message
error: Auto-discovery encountered an exception: {0}
fatal: Required config property not set: {0}
error: Unable to construct reader: {0}. {1}
warn: No logging level specified. Defaulting to: {0}
warn: No logging file specified. Defaulting to: {0}
error: Unknown log type {0} specified for Windows host: {1}
error: Simple config property: {0} requires {1}
error: Required config property not set: {0}
warn: Optional config property not set: {0}, using default: {1}
error: Required config property not set: {0}
warn: Auto-discovery of event sources disabled, missing config property: {0}
error: Required config property not set: {0} , no event types will be read from discovered sources.
warn: Config property not set: {0}
info: Auto-discovered host: {0}
warn: Config property not set: {0} , using default: {1}
warn: Error recovering stale log files in {0}
error: Required config property not set: {0}
warn: Config property not set: {0} , using default: {1}
warn: Config property not set: {0} , using default: {0}
error: Unable to open a TCP socket connection to {0}:{1}
error: Missing required config property: {0}, no records will be filtered by ID.
warn: Config property not set: {0} , using default: {1}
error: Unable to initialize StateManager database, no state will be persisted {0}
error: Error closing StateManager database. {0}
error: Unable to load the state of pipeline[{0}

Message
error: Unable to save the state of pipeline[{0}
debug: EventPipelineState::writeObject
debug: EventPipeline::writeObject pipelineDetails: {0}
info: EventPipeline:readObject started, version {0}
info: EventPipeline:readObject::Upgrading state for key = {0}
info: EventPipeline:readObject::During upgrading state, offset null for {0} using zero.
error: EventPipeline:readObject::Unknown version of EventPipelineState: {0}
info: EventPipeline:readObject pipelineDetails: {0}
info: EventPipeline:readObject this.toString: {0}
error: Unknown version of EventPipelineStateDetails: {0}
info: SMBEventReader:init:: eventTypeStr = {0}
debug: SMBEventReader:isRemoveLogReset:: pipeline = {0}
warn: Pipeline {0}
trace: SMBEventReader:read:: handleName = {0} , readOffset = {1}
info: Using skipAheadMap to populate handle {0} with offset {1}
info: SMBEventReader:read:: For pipe = {0}. Using oldest record number = {1}
trace: SMBEventReader:read:: records.readOffset = {0}
trace: SMBEventReader:read:: {0}
trace: SMBEventReader:read:: skipping recordNumber {0}
warn: SMBEventReader:read:: record number was -1 for handle = {0}
warn: SMBEventReader:prepareForRestart:: skipAheadMap already in use!
info: SMBEventReader:prepareForRestart:: handleName = {0} , readOffset = {1}
error: SMBEventReader:prepareForRestart:: skipping due to null log handles
error: SMBEventReader:close:: null logHandles
error: SMBEventReader:close:: null adminSession
info: SMBEventReader:setDebug:: adminSession is null. enableDebug = {0}.
warn: Config property not set: {0} , using default: {1}
warn: Config property not set: {0} , defaulting to:
info: Remote shell server listening on port: {0}
info: Remote shell accepting localhost connections only: {0}
error: Unable to start a remote shell running on port: {0}, e
info: Received client connection: {0}
info: Refusing non-localhost connection from: {0}
info: Verified localhost connection.
info: Remote client disconnected: {0}
fatal: Uncaught exception in thread ‘ {0}
info: RTNet enabled: {0}
error: Required config property not set: {0} , using default: {1}
debug: Initializing RTNet: {0} = {1} , {2} = {3} ,

Message
fatal: Unable to initialize RTNet: {0}
error: Unable to create pipeline [{0}
error: Uncaught exception in Remote Shell thread.{0}
error: Attempting to restart Remote Shell thread in {0} seconds.
error: Uncaught exception in Pipeline Monitor thread.{0}
error: Attempting to restart Pipeline Monitor thread in {0} seconds.
error: Unable to load config file: {0}
trace: getPipelineThread threw = {0}
error: Pipeline[{0}] has error with reader: {1}
error: Unable to establish pipeline[{0}] due to bad readers.
debug: Pipeline[{0}]: assigning filter: {1}
error: Pipeline[{0}]: Unable to construct filter.{1}
warn: Polling interval property not set: {0} , using default: {1}
warn: Invalid polling interval: {0} , using default: {1}
warn: Invalid block size: {0} , using default: {1}
warn: Delete Check property not set: {0} , using default: {1}
warn: Invalid delete check: {0} , using default: {1}
debug: Unable to write records {0}
info: Pipeline Restarter: attempting to start broken pipeline[{0}]
info: Pipeline Restarter: stopping pipeline thread[{0}]
info: Pipeline Restarter: waiting for pipeline to stop[{0}]
info: Pipeline Restarter: pipeline[{0}] is not stopped yet
warn: Pipeline Restarter: Can’t restart pipeline[{0}]. Pipeline would not stop.
info: Pipeline Restarter: preparing reader for restart[{0}
info: Pipeline Restarter: resetting state on pipeline[{0}]
info: Pipeline Restarter: State reset for pipeline[{0}]
warn: Pipeline Restarter: Pipeline not found: pipeline[{0}]
info: Pipeline Monitor: checking for broken pipelines every {0} seconds.
info: Pipeline Monitor: pipeline[{0}] appears to be broken, will attempt restart.
info: Pipeline Monitor: all pipelines appear to be working.
fatal: RTNet Native Implementation: winretriever stub was not loaded, unknown operating system
warn: Missing config property: {0} , using default: {1}

APPENDIX C

TIME ZONES

This appendix lists every time zone supported by the EDW and the HawkEye AP Console. When you enter a time value in HawkEye AP Console, use one of the time-zone formats listed below in “Supported Time Zones”, on page 349.

Because the EDW handles raw data from logs, which does not always use the supported time zones, it must accommodate some shortenings of time zone indicators found in that data (such as PDT for Pacific Daylight Time and CST for Central Standard Time). For more information, see “Time-Zone Conversion”, next.

TIME-ZONE CONVERSION

Sensage SQL provides functions that recognize specific time-zone shortenings and maps them to standard time-zone strings, as shown in the table below.

Shortened Time Zone	Standard Time Zone	Description
`PST`	`PST8PDT`	Pacific Standard Time and Pacific Daylight Time are interpreted as either standard time or daylight savings depending on the time of year of the actual date.
`PDT`	`PST8PDT`	
`MDT`	`MST7MDT`	Mountain Daylight Time is interpreted as either standard time or daylight savings depending on the time of year of the actual date.
`CST`	`CST6CDT`	Central Standard Time and Central Daylight Time are interpreted as either standard time or daylight savings depending on the time of year of the actual date.
`CDT`	`CST6CDT`	
`EDT`	`EST5EDT`	Eastern Daylight Time is interpreted as either standard time or daylight savings depending on the time of year of the actual date.

NOTE:

- Because Arizona does not support MDT, Mountain Standard Time is NOT converted to Mountain Daylight Time.
- Because Indiana does not consistently support EDT, Eastern Standard Time is NOT converted to Eastern Daylight Time.

SUPPORTED TIME ZONES

```
Africa/Addis_Ababa
Africa/Algiers
Africa/Asmera
Africa/Bangui
Africa/Blantyre
Africa/Brazzaville
Africa/Bujumbura
Africa/Cairo
```

```
Africa/Ceuta
Africa/Dar_es_Salaam
Africa/Djibouti
Africa/Douala
Africa/Gaborone
Africa/Harare
Africa/Johannesburg
Africa/Kampala
Africa/Khartoum
Africa/Kigali
Africa/Kinshasa
Africa/Lagos
Africa/Libreville
Africa/Luanda
Africa/Lubumbashi
Africa/Lusaka
Africa/Malabo
Africa/Maputo
Africa/Maseru
Africa/Mbabane
Africa/Mogadishu
Africa/Nairobi
Africa/Ndjamena
Africa/Niamey
Africa/Porto-Novo
Africa/Tripoli
Africa/Tunis
Africa/Windhoek
America/Adak
America/Anchorage
America/Anguilla
America/Antigua
America/Araguaina
America/Argentina/Buenos_Aires
America/Argentina/Catamarca
America/Argentina/ComodRivadavia
America/Argentina/Cordoba
America/Argentina/Jujuy
America/Argentina/La_Rioja
America/Argentina/Mendoza
America/Argentina/Rio_Gallegos
America/Argentina/San_Juan
America/Argentina/Tucuman
America/Argentina/Ushuaia
America/Aruba
America/Asuncion
America/Atka
America/Bahia
America/Barbados
America/Belem
America/Belize
America/Boa_Vista
America/Bogota
America/Boise
America/Buenos_Aires
America/Cambridge_Bay
America/Campo_Grande
America/Cancun
America/Caracas
```

```
America/Catamarca
America/Cayenne
America/Cayman
America/Chicago
America/Chihuahua
America/Coral_Harbour
America/Cordoba
America/Costa_Rica
America/Cuiaba
America/Curacao
America/Dawson
America/Dawson_Creek
America/Denver
America/Detroit
America/Dominica
America/Edmonton
America/Eirunepe
America/El_Salvador
America/Ensenada
America/Fort_Wayne
America/Fortaleza
America/Glace_Bay
America/Godthab
America/Goose_Bay
America/Grand_Turk
America/Grenada
America/Guadeloupe
America/Guatemala
America/Guayaquil
America/Guyana
America/Halifax
America/Havana
America/Hermosillo
America/Indiana/Indianapolis
America/Indiana/Knox
America/Indiana/Marengo
America/Indiana/Vevay
America/Indianapolis
America/Inuvik
America/Iqaluit
America/Jamaica
America/Jujuy
America/Juneau
America/Kentucky/Louisville
America/Kentucky/Monticello
America/Knox_IN
America/La_Paz
America/Lima
America/Los_Angeles
America/Louisville
America/Maceio
America/Managua
America/Manaus
America/Martinique
America/Mazatlan
America/Mendoza
America/Menominee
America/Merida
America/Mexico_City
```

```
America/Miquelon
America/Monterrey
America/Montevideo
America/Montreal
America/Montserrat
America/Nassau
America/New_York
America/Nipigon
America/Nome
America/Noronha
America/North_Dakota/Center
America/Panama
America/Pangnirtung
America/Paramaribo
America/Phoenix
America/Port-au-Prince
America/Port_of_Spain
America/Porto_Acre
America/Porto_Velho
America/Puerto_Rico
America/Rainy_River
America/Rankin_Inlet
America/Recife
America/Regina
America/Rio_Branco
America/Rosario
America/Santiago
America/Santo_Domingo
America/Sao_Paulo
America/Shiprock
America/St_Johns
America/St_Kitts
America/St_Lucia
America/St_Thomas
America/St_Vincent
America/Swift_Current
America/Tegucigalpa
America/Thule
America/Thunder_Bay
America/Tijuana
America/Toronto
America/Tortola
America/Vancouver
America/Virgin
America/Whitehorse
America/Winnipeg
America/Yakutat
America/Yellowknife
Antarctica/Casey
Antarctica/Davis
Antarctica/DumontDUrville
Antarctica/Mawson
Antarctica/McMurdo
Antarctica/Palmer
Antarctica/Rothera
Antarctica/South_Pole
Antarctica/Syowa
Antarctica/Vostok
Arctic/Longyearbyen
```

```
Asia/Aden
Asia/Almaty
Asia/Amman
Asia/Anadyr
Asia/Aqtau
Asia/Aqtobe
Asia/Ashgabat
Asia/Ashkhabad
Asia/Baghdad
Asia/Bahrain
Asia/Baku
Asia/Bangkok
Asia/Beirut
Asia/Bishkek
Asia/Brunei
Asia/Calcutta
Asia/Choibalsan
Asia/Chongqing
Asia/Chungking
Asia/Colombo
Asia/Dacca
Asia/Damascus
Asia/Dhaka
Asia/Dili
Asia/Dubai
Asia/Dushanbe
Asia/Gaza
Asia/Harbin
Asia/Hong_Kong
Asia/Hovd
Asia/Irkutsk
Asia/Istanbul
Asia/Jakarta
Asia/Jayapura
Asia/Jerusalem
Asia/Kabul
Asia/Kamchatka
Asia/Karachi
Asia/Kashgar
Asia/Katmandu
Asia/Krasnoyarsk
Asia/Kuala_Lumpur
Asia/Kuching
Asia/Kuwait
Asia/Macao
Asia/Macau
Asia/Magadan
Asia/Makassar
Asia/Manila
Asia/Muscat
Asia/Nicosia
Asia/Novosibirsk
Asia/Omsk
Asia/Oral
Asia/Phnom_Penh
Asia/Pontianak
Asia/Pyongyang
Asia/Qatar
Asia/Qyzylorda
```

```
Asia/Rangoon
Asia/Riyadh
Asia/Riyadh87
Asia/Riyadh88
Asia/Riyadh89
Asia/Saigon
Asia/Sakhalin
Asia/Samarkand
Asia/Seoul
Asia/Shanghai
Asia/Singapore
Asia/Taipei
Asia/Tashkent
Asia/Tbilisi
Asia/Tehran
Asia/Tel_Aviv
Asia/Thimbu
Asia/Thimphu
Asia/Tokyo
Asia/Ujung_Pandang
Asia/Ulaanbaatar
Asia/Ulan_Bator
Asia/Urumqi
Asia/Vientiane
Asia/Vladivostok
Asia/Yakutsk
Asia/Yekaterinburg
Asia/Yerevan
Atlantic/Bermuda
Atlantic/Canary
Atlantic/Cape_Verde
Atlantic/Faeroe
Atlantic/Jan_Mayen
Atlantic/Madeira
Atlantic/South_Georgia
Atlantic/Stanley
Australia/ACT
Australia/Adelaide
Australia/Brisbane
Australia/Broken_Hill
Australia/Canberra
Australia/Currie
Australia/Darwin
Australia/Hobart
Australia/LHI
Australia/Lindeman
Australia/Lord_Howe
Australia/Melbourne
Australia/NSW
Australia/North
Australia/Perth
Australia/Queensland
Australia/South
Australia/Sydney
Australia/Tasmania
Australia/Victoria
Australia/West
Australia/Yancowinna
Brazil/Acre
```

```
Brazil/DeNoronha
Brazil/East
Brazil/West
CET
CST6CDT
Canada/Atlantic
Canada/Central
Canada/East-Saskatchewan
Canada/Eastern
Canada/Mountain
Canada/Newfoundland
Canada/Pacific
Canada/Saskatchewan
Canada/Yukon
Chile/Continental
Chile/EasterIsland
Cuba
EET
EST
EST5EDT
Egypt
Eire
Europe/Amsterdam
Europe/Andorra
Europe/Athens
Europe/Belfast
Europe/Belgrade
Europe/Berlin
Europe/Bratislava
Europe/Brussels
Europe/Bucharest
Europe/Budapest
Europe/Chisinau
Europe/Copenhagen
Europe/Dublin
Europe/Gibraltar
Europe/Helsinki
Europe/Istanbul
Europe/Kaliningrad
Europe/Kiev
Europe/Lisbon
Europe/Ljubljana
Europe/London
Europe/Luxembourg
Europe/Madrid
Europe/Malta
Europe/Mariehamn
Europe/Minsk
Europe/Monaco
Europe/Moscow
Europe/Nicosia
Europe/Oslo
Europe/Paris
Europe/Prague
Europe/Riga
Europe/Rome
Europe/Samara
Europe/San_Marino
Europe/Sarajevo
```

```
Europe/Simferopol
Europe/Skopje
Europe/Sofia
Europe/Stockholm
Europe/Tallinn
Europe/Tirane
Europe/Tiraspol
Europe/Uzhgorod
Europe/Vaduz
Europe/Vatican
Europe/Vienna
Europe/Vilnius
Europe/Warsaw
Europe/Zagreb
Europe/Zaporozhye
Europe/Zurich
GB
GB-Eire
GMT
HST
Hongkong
Indian/Antananarivo
Indian/Chagos
Indian/Christmas
Indian/Cocos
Indian/Comoro
Indian/Kerguelen
Indian/Mahe
Indian/Maldives
Indian/Mauritius
Indian/Mayotte
Indian/Reunion
Iran
Israel
Jamaica
Japan
Kwajalein
Libya
MET
MST
MST7MDT
Mexico/BajaNorte
Mexico/BajaSur
Mexico/General
Mideast/Riyadh87
Mideast/Riyadh88
Mideast/Riyadh89
NZ
NZ-CHAT
Navajo
PRC
PST8PDT
Pacific/Apia
Pacific/Auckland
Pacific/Chatham
Pacific/Easter
Pacific/Efate
Pacific/Enderbury
Pacific/Fakaofo
```

```
Pacific/Fiji
Pacific/Funafuti
Pacific/Galapagos
Pacific/Gambier
Pacific/Guadalcanal
Pacific/Guam
Pacific/Honolulu
Pacific/Johnston
Pacific/Kiritimati
Pacific/Kosrae
Pacific/Kwajalein
Pacific/Majuro
Pacific/Marquesas
Pacific/Midway
Pacific/Nauru
Pacific/Niue
Pacific/Norfolk
Pacific/Noumea
Pacific/Pago_Pago
Pacific/Palau
Pacific/Pitcairn
Pacific/Ponape
Pacific/Port_Moresby
Pacific/Rarotonga
Pacific/Saipan
Pacific/Samoa
Pacific/Tahiti
Pacific/Tarawa
Pacific/Tongatapu
Pacific/Truk
Pacific/Wake
Pacific/Wallis
Pacific/Yap
Poland
Portugal
ROK
Singapore
Turkey
US/Alaska
US/Aleutian
US/Arizona
US/Central
US/East-Indiana
US/Eastern
US/Hawaii
US/Indiana-Starke
US/Michigan
US/Mountain
US/Pacific
US/Samoa
UTC
W-SU
WET
```

INDEX