

IBM Operations Analytics - Log Analysis
Version 1.3.3

*Installation, Configuration, and
Administration Guide*



Note

Before using this information and the product it supports, read the information in [Appendix A, “Notices,”](#) on page 365.

Edition notice

This edition applies to IBM® Operations Analytics - Log Analysis and to all subsequent releases and modifications until otherwise indicated in new editions.

References in content to IBM products, software, programs, services or associated technologies do not imply that they will be available in all countries in which IBM operates. Content, including any plans contained in content, may change at any time at IBM's sole discretion, based on market opportunities or other factors, and is not intended to be a commitment to future content, including product or feature availability, in any way. Statements regarding IBM's future direction or intent are subject to change or withdrawal without notice and represent goals and objectives only. Please refer to the [developerWorks terms of use](#) for more information.

© **Copyright International Business Machines Corporation 2015.**

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Chapter 1. About this publication.....	1
Audience.....	1
Publications.....	1
Accessing terminology online.....	1
Accessibility.....	1
Tivoli technical training.....	1
Providing feedback.....	1
Conventions used in this publication	2
Typeface conventions	2
Chapter 2. Installing.....	3
Hardware and software requirements.....	3
Downloading Log Analysis.....	6
Prerequisites.....	6
IBM Prerequisite Scanner.....	6
Create a non-root user.....	7
Recommended library installers.....	7
Verifying the operating system version.....	8
Verifying the 64-bit library requirement.....	8
Disabling Security-Enhanced Linux (SELinux).....	8
Verifying KornShell library.....	9
Verifying the Python version.....	9
Install python-simplejson package for Python.....	9
Installing unzip utility.....	10
Verifying the host server IP address and names.....	10
Number of open files and virtual memory limits.....	10
Verifying the IP address and host name configurations.....	12
Time server.....	12
Installing on IBM Power8 servers.....	12
Installing on Linux on System z and Linux on System x based servers.....	13
Installing.....	15
Installing with the IBM Installation Manager UI.....	15
Installing with the IBM Installation Manager command-line interface.....	17
Silently installing Log Analysis.....	18
Installing and configuring the IBM Tivoli Monitoring Log File Agent.....	21
Verifying the version information.....	22
Removing IBM Operations Analytics - Log Analysis.....	22
Removing IBM Operations Analytics - Log Analysis.....	22
Using the console to remove IBM Operations Analytics - Log Analysis.....	23
Silently removing IBM Operations Analytics - Log Analysis.....	23
Installing reference.....	25
Configuration properties file.....	25
Default ports.....	26
install.sh command.....	26
backup_restore.sh script.....	27
Chapter 3. Upgrading, backing up, and migrating data.....	29
Backup and migration limitations.....	31
Backing up data.....	31
Restoring data.....	32

Updating remote Logstash installations.....	33
Chapter 4. Configuring.....	35
Postinstallation configuration.....	35
Configuring secure communication and authentication.....	35
Users and roles.....	58
System.....	64
Configuring the time zone for indexing.....	72
Managing alerts.....	73
Creating alerts.....	73
Manage Alerts UI.....	76
Configuring alerts for SNMP and EIF events.....	79
Configuring email alert actions.....	84
Configuring auditability.....	85
Audit parameters.....	85
Viewing the audit file.....	86
Configuring Expert Advice Custom Search Dashboard.....	86
Customizing the default Expert Advice custom Custom Search Dashboard.....	86
Configuring a custom Expert Advice app.....	88
Configuring launch in context.....	91
Search UI launch-in-context.....	91
Custom Search Dashboard launch-in-context.....	92
Configuring the DSV toolkit.....	92
What's new.....	93
Create an Insight Pack using the DSV toolkit.....	93
Specifying properties for a log file type.....	93
Generate a properties file.....	97
Generate an Insight Pack.....	98
Troubleshooting.....	99
Supported formats.....	100
DSV formats.....	100
Configuring aliases.....	101
Configuring an alias.....	101
Configuration reference.....	102
ldapRegistryHelper.properties.....	102
ldapRegistryHelper.sh command.....	103
unity command.....	104
LFA configuration file parameters.....	104
eifutil.sh command.....	107
lfautil.sh command.....	108
Data Collector properties.....	108
unity_securityUtility.sh command.....	109
securityUtility.sh utility.....	109
eif.conf file.....	110
unity.conf file.....	111
install.sh command.....	112
ssh-config.properties.....	113
Audit parameters.....	113
Version utility.....	114
Supported time zone names.....	119
Supported languages.....	134
Chapter 5. Data tiering and storage.....	135
Configuring the data archive.....	135
Configuring long-term data storage.....	136
Supported versions of Hadoop.....	137
Prerequisite tasks.....	138

Configuring long term data storage automatically.....	138
Manually configuring long term data storage.....	140
Testing the Hadoop integration.....	143
Hadoop Integration tab.....	144
Managing the Log Analysis server.....	144
Maintaining automatically configured connections.....	144
Editing a NameNode server connection.....	144
Editing a DataNode server connection.....	145
Deleting a NameNode server connection.....	145
Deleting a DataNode server connection.....	145
Maintaining manually configured DataNode connections.....	145
Sharing a Hadoop cluster across multiple Log Analysis instances.....	146
DataNode server connection editor reference.....	147
NameNode server connection editor reference.....	147
 Chapter 6. Loading and streaming data.....	 149
Data collection tools.....	149
Supported operating systems for data collection.....	150
Deploying scalable data collection architecture.....	151
Planning your deployment.....	151
Configuring scalable data collection.....	158
Configuring Insight Packs.....	169
Managing your data collection components.....	176
Increasing the volume of data.....	178
Loading batches of data.....	178
Data Collector client.....	178
Generic Receiver.....	181
Loading batches of historic data with the IBM Tivoli Monitoring Log File Agent.....	183
Loading a batch of log files with the LFA.....	183
Streaming data with the IBM Tivoli Monitoring Log File Agent.....	186
Considerations when using the LFA.....	187
Configuring the LFA.....	187
Regular expression support for the LFA.....	209
Troubleshooting data loading.....	210
Streaming data with the IBM Performance Management OS agent.....	212
Configuring the IBM Performance Management OS agent to stream data.....	212
Streaming data from multiple remote sources across a network.....	213
Configuring secure shell (SSH) communication for multiple remote hosts.....	214
Deploying the LFA or EIF on remote servers.....	215
Streaming data with logstash.....	218
Dependencies.....	218
Removing logstash 1.5.3.....	219
Upgrading to logstash 2.2.1.....	219
Installing logstash on a remote node.....	220
logstash configuration.....	221
Installing Logstash on Windows based servers.....	228
Logstash configuration file reference.....	229
Logstash operations.....	231
logstash best practices.....	231
References.....	231
Known issues.....	232
Configuring the EIF Receiver.....	233
Configuring receiver buffer size and timeout.....	233
Configuring the EIF receiver user account.....	234
Configuring the number of events in the EIF Receiver.....	235
Configuring the EIF Receiver memory clean up interval	236
Changing the default password for the Data Collector and EIF Receiver.....	237

Changing the default EIF Receiver or Data Collector password.....	237
unity_securityUtility.sh command.....	238
Ingestion of non-English-language log files.....	238
Ingesting non-English-language log files with the internal IBM Tivoli Monitoring Log File Agent..	239
Streaming non-English-language log files from a remote, internal IBM Tivoli Monitoring Log File Agent.....	239
Ingesting non-English-language log files from an external IBM Tivoli Monitoring Log File Agent..	240
Ingesting non-English-language log files with the Data Collector.....	241
Chapter 7. Managing Insight Packs.....	243
Supported software.....	244
Downloading an Insight Pack.....	244
Installing Insight Packs.....	244
Upgrading an Insight Pack.....	245
Cloning source types for Insight Packs.....	245
Out of the box Insight Packs.....	246
DB2 Insight Pack.....	246
Generic Annotation Insight Pack.....	256
Javacore Insight Pack.....	267
Syslog Insight Pack.....	272
Web Access Logs Insight Pack.....	280
WebSphere Application Server Insight Pack.....	290
Windows OS Events Insight Pack.....	299
Custom Insight Packs.....	305
Chapter 8. Administrating.....	307
Getting started with Log Analysis.....	307
Logging in to IBM Operations Analytics - Log Analysis	307
Installing sample files.....	307
Enabling the GUI search history.....	308
Defining a default search.....	308
Creating and updating the data model.....	309
Data Sources workspace.....	309
Data Types workspace.....	311
Administrative tasks.....	323
Configuring automatic refreshes for new dashboards.....	323
Configuring automatic refreshes for existing dashboards.....	323
Configuring the timeout for the Log Analysis server.....	324
Adding or removing IBM Tivoli Monitoring Log File Agent from an existing installation.....	325
Changing the IP address for Log Analysis.....	325
Monitoring and configuring data ingestion statistics.....	325
Configuring the data limit for bundled versions.....	326
Server Statistics workspace.....	327
export_statistics command.....	327
Deleting data.....	329
Limitations and prerequisites for data deletion.....	330
Deleting data.....	330
Administering reference.....	332
export_statistics command.....	332
delete.properties.....	334
API guide.....	335
Manually configuring authentication for the REST API.....	335
Using the Data Collector client to authenticate and invoke the REST API.....	336
Search REST API overview.....	338
Search query API.....	347
Using the REST API to administer the Log Analysis data model.....	349
REST API for asynchronous searches.....	360

Appendix A. Notices.....	365
Trademarks.....	366
Terms and conditions for product documentation.....	366
IBM Online Privacy Statement.....	367
.....	368
Trademarks.....	368

Chapter 1. About this publication

This guide contains information about how to use IBM Operations Analytics - Log Analysis.

Audience

This publication is for users of the IBM Operations Analytics - Log Analysis product.

Publications

This section provides information about the IBM Operations Analytics - Log Analysis publications. It describes how to access and order publications.

Accessing terminology online

The IBM Terminology Web site consolidates the terminology from IBM product libraries in one convenient location. You can access the Terminology Web site at the following Web address:

<http://www.ibm.com/software/globalization/terminology>.

Accessibility

Accessibility features help users with a physical disability, such as restricted mobility or limited vision, to use software products successfully. In this release, the IBM Operations Analytics - Log Analysis user interface does not meet all accessibility requirements.

Accessibility features

This information center, and its related publications, are accessibility-enabled. To meet this requirement the user documentation in this information center is provided in HTML and PDF format and descriptive text is provided for all documentation images.

Related accessibility information

You can view the publications for IBM Operations Analytics - Log Analysis in Adobe Portable Document Format (PDF) using the Adobe Reader.

IBM and accessibility

For more information about the commitment that IBM has to accessibility, see the IBM Human Ability and Accessibility Center. The IBM Human Ability and Accessibility Center is at the following web address: <http://www.ibm.com/able> (opens in a new browser window or tab)

Tivoli technical training

For Tivoli® technical training information, refer to the following IBM Tivoli Education Web site at <http://www.ibm.com/software/tivoli/education>.

Providing feedback

We appreciate your comments and ask you to submit your feedback to the IBM Operations Analytics - Log Analysis community.

Conventions used in this publication

This publication uses several conventions for special terms and actions, operating system-dependent commands and paths, and margin graphics.

Typeface conventions

This publication uses the following typeface conventions:

Bold

- Lowercase commands and mixed case commands that are otherwise difficult to distinguish from surrounding text
- Interface controls (check boxes, push buttons, radio buttons, spin buttons, fields, folders, icons, list boxes, items inside list boxes, multicolumn lists, containers, menu choices, menu names, tabs, property sheets), labels (such as **Tip:**, and **Operating system considerations:**)
- Keywords and parameters in text

Italic

- Citations (examples: titles of publications, diskettes, and CDs)
- Words defined in text (example: a nonswitched line is called a *point-to-point line*)
- Emphasis of words and letters (words as words example: "Use the word *that* to introduce a restrictive clause."; letters as letters example: "The LUN address must start with the letter *L*.")
- New terms in text (except in a definition list): a *view* is a frame in a workspace that contains data.
- Variables and values you must provide: ... where *myname* represents....

Monospace

- Examples and code examples
- File names, programming keywords, and other elements that are difficult to distinguish from surrounding text
- Message text and prompts addressed to the user
- Text that the user must type
- Values for arguments or command options

Chapter 2. Installing

This section outlines the procedure that you must follow to install IBM Operations Analytics - Log Analysis. Details of prerequisites, the installation procedure, and additional post-installation tasks are outlined in this section.



Warning: You cannot install IBM Operations Analytics - Log Analysis on a Network File System (NFS) based, shared resource.

Note:

If you want to install IBM Operations Analytics - Log Analysis on a system where the locale is not set to English United States, you need to set the locale of the command shell to export `LANG=en_US.UTF-8` before you run any IBM Operations Analytics - Log Analysis scripts. For example, when you install IBM Operations Analytics - Log Analysis you need to run the following command to change the locale before you run the `unity.sh` script:

```
export LANG=en_US.UTF-8
```

Hardware and software requirements

The hardware and software requirements vary according to the type of deployment that you want to set up. Before you install IBM Operations Analytics - Log Analysis, review the requirements that are listed here.

Supported operating systems

Table 1. Supported operating systems		
Operating system	Red Hat Enterprise Linux version	SUSE Linux Enterprise Server version
IBM System z®	6, 7, or 7.1	11 and 12
IBM System x	5, 6, 7, or 7.1	11 and 12
IBM Power8 Little Endian	7.1	-

Log Analysis supports both System z and x86 hardware. Operating systems that are installed on System z based hardware are referred to as Linux on System z. Operating systems that are installed on x86 based hardware are referred to as Linux on System x.

Supported browsers

- Mozilla Firefox Extended Support Release (ESR) version 38
- Microsoft Internet Explorer 10 and 11
- Google Chrome versions 44, 45, and 47
- Microsoft EDGE



Warning: If you use Internet Explorer 11, the user interface might not render properly or at all. To fix the issue, you need to disable the compatibility lists. For more information about how to do disable the compatibility lists, see the *Cannot display the UI in Microsoft Internet Explorer 11* topic in the *Troubleshooting Guide*.

Software requirements

- KornShell

- Python Version 2.4.3 with simplejson module (RHEL v5) or Python Version 2.6.6 to 2.6.8 (RHEL v6 or higher, SLES v11 or higher)
- Perl Version 5.8.8 or later is required to load sample data
- Unzip utility is required to install the Indexing Server.
- Sed utility is required to install the Indexing Server.

Tip: To install the required libraries on Red Hat Enterprise Linux Server Edition, YUM is recommended. On SUSE Linux Enterprise Server, YaST is recommended. For more information about installing these libraries and additional information about setting up your environment, see the *Installing* section of the documentation.

Hardware requirements

The following hardware requirements are representative of what may be required for an installation of IBM Operations Analytics - Log Analysis. The hardware requirements can change depending on the amount of data that you stream.

The specifications assume that you install IBM Operations Analytics - Log Analysis, the Indexing Engine component, and IBM InfoSphere® on separate servers:

Test configuration

The test configuration assumes that you intend to stream about 1 Gigabyte (GB) of data daily and that you plan to retain data for seven days.

IBM Operations Analytics - Log Analysis server:

- Four 2.8 Gigahertz (GHz) core processors
- 16 gigabytes (GB) of RAM
- 60 GB of hard disk space

Indexing Engine server

- Four 2.8 GHz core processors
- 16 GB of RAM
- 10 GB of hard disk space

Commodity configuration

The commodity configuration assumes that you intend to stream about 250 GBs of data daily and that you plan to retain data for 15 days.

IBM Operations Analytics - Log Analysis server:

- Eight 2.8 Gigahertz (GHz) core processors
- 16 GB of RAM
- 60 GB of hard disk space

Indexing Engine server

- Eight 2.8 GHz core processors
- 64 GB of RAM
- 4 Terabytes (TB) of hard disk space

Production configuration

The production configuration assumes that you intend to stream about 500 GBs of data daily and that you plan to retain data for 30 days.

IBM Operations Analytics - Log Analysis server:

- Sixteen 2.8 Gigahertz (GHz) core processors
- 16 GB of RAM
- 60 GB of hard disk space

Indexing Engine server

- Sixteen 2.8 GHz core processors
- 64 GB of RAM
- 16 TB of hard disk space

You can use IBM Operations Analytics - Log Analysis to stream up to two TBs of data daily.

Processor, disk, and memory types

IBM Operations Analytics - Log Analysis is designed to be compatible with a wide range of processors, hard disks and memory types.

Software requirements for Linux on System z and Linux on System x

You can install IBM Operations Analytics - Log Analysis on Linux on System z and Linux on System x. For more information, see the [“Installing on Linux on System z and Linux on System x based servers”](#) on page 13 in topic the *Installing* section of the documentation.

Software requirements:

- KornShell is required for the IBM Tivoli Monitoring Log File Agent.
- Python Version 2.4.3 with simplejson module (RHEL v5) or Python Version 2.6.6 to 2.6.8 (RHEL v6 or higher, SLES v11 or higher).
- Perl Version 5.8.8 or later is required to load sample data.
- The Unzip utility is required to install the Indexing Server.
- The Sed utility is required to install the Indexing Server.
- The following RPMs are required for both Red Hat Enterprise Linux on System z and SUSE Linux on System z:
 - libstdc++ (32-bit and 64-bit)
 - gtk2 (32-bit and 64-bit)
 - libXpm (32-bit and 64-bit)
 - libXtst (32-bit and 64-bit) is required if you want to run the Installation Manager in GUI mode.

You also need to complete the prerequisite tasks, which include disabling the Security Enhanced Linux module. For more information, see the *Prerequisite* topic in the *Installing* section of the documentation.

To tune the Linux operating system, ensure that the following resource configuration settings are made:

Number of concurrent files: 4096

Use the `ulimit -n` command to change this setting to 4096.

Virtual memory: Unlimited

Use the `ulimit -v` command to change this setting to unlimited.

Hardware for Linux on System z and Linux on System x

The following hardware requirements are representative of what may be required for an installation of IBM Operations Analytics - Log Analysis on Linux on System z and Linux on System x. The hardware requirements can change depending on the amount of data that you stream.

This configuration assumes that you intend to stream about 10 GB of data daily. This scenario assumes that you install two logical partitions (LPARs) and that the LPARs stream 5 GBs of data each day. It is assumed that data is retained for 10 days:

- One Central Processing Unit (CPU)
- 16 GB of RAM
- 150 GB of hard disk space

Supported file systems

The following file systems are supported for both operating systems:

- ext2/3/4
- reiser3
- XFS
- NFS 3/4

Hardware and software requirements for Hadoop

Standard Edition users can integrate Hadoop for long term data storage.

Several different versions of Hadoop are supported. For more information see the *Supported versions of Hadoop* topic in the *Data tiering and storage* section of the documentation.

Downloading Log Analysis

Read this document to get an overview of how to download Log Analysis from Passport Advantage.

1. Plan your installation. For more information, see [About IBM Operations Analytics - Log Analysis](#)
2. Review the prerequisites, including the hardware and software requirements. For more information, see [“Prerequisites”](#) on page 6.
3. Go to Passport Advantage at <https://www.ibm.com/developerworks/servicemanagement/iaa/log/downloads.html>.
4. Choose the type of installation.

IBM Operations Analytics - Log Analysis for Linux on System x86_64 or Linux on System z. For more information, see [“Installing on Linux on System z and Linux on System x based servers”](#) on page 13.

To install IBM Operations Analytics - Log Analysis on Linux on System x86_64, download the **IBM Operations Analytics - Log Analysis Linux 64 bit (CN8IEN)** package.

To install IBM Operations Analytics - Log Analysis on Linux on System z, download the **IBM Operations Analytics - Log Analysis Linux on System z 64 bit (CN8IJEN)** package.

To install IBM Operations Analytics - Log Analysis on IBM Power8 Little Endian for Linux based servers, download the **IBM Operations Analytics - Log Analysis 1.3.3 Power8 ppc64le ALL editions (CN8K9EN)** package from <https://www.ibm.com/developerworks/servicemanagement/iaa/log/downloads.html>.

For more information about how to install the product, see [Chapter 2, “Installing,”](#) on page 3.

5. Complete the post-installation configuration. For more information, see [“Postinstallation configuration”](#) on page 35.
6. After you install IBM Operations Analytics - Log Analysis, you can configure it to meet your needs. For more information, see [Chapter 4, “Configuring,”](#) on page 35.

Prerequisites

Before you install IBM Operations Analytics - Log Analysis, ensure that the system meets the hardware and software requirements and complete the prerequisite tasks.

IBM Prerequisite Scanner

IBM Prerequisite Scanner is a prerequisite checking tool that analyzes system environments before you install or upgrade IBM Operations Analytics - Log Analysis.

The IBM Prerequisite Scanner compressed file is packaged with IBM Operations Analytics - Log Analysis.

Run the IBM Prerequisite Scanner as part of your installation planning before you install IBM Operations Analytics - Log Analysis.

Installing the IBM Prerequisite Scanner

1. To install the IBM Prerequisite Scanner, select one of the following options:
 - Locate the IBM Prerequisite Scanner compressed file in the IBM Operations Analytics - Log Analysis package.
 - Download the most recent scanner version for your operating system from IBM Fix Central: <http://www-933.ibm.com/support/fixcentral/swg/selectFixes?parent=ibm~Tivoli&product=ibm/Tivoli/Prerequisite+Scanner&release=All&platform=All&function=all>
2. Extract the contents of the IBM Prerequisite Scanner compressed file to a preferred location as specified by `ips_root` on each target system that you intend to install IBM Operations Analytics - Log Analysis.

Note: You must have write permissions to the root directory in which you extract the contents of the compressed file.

Running the IBM Prerequisite Scanner

1. Open the command window.
2. Change the `ips_root` directory to the directory where the scanner is expanded.
3. Set the value for the relevant environment variable to **True**. For example:

```
export ENV_NAME=True
```

4. Run the IBM Prerequisite Scanner script file, `prereq_checker`. For example:

```
prereq_checker.sh "ILA 01320000" detail outputDir=<output_directory>
```

5. Check the output results of the scanner and resolve any failed checks.

For more information about the IBM Prerequisite Scanner, see the IBM Prerequisite Scanner wiki page: <https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Documentation%20Central/page/IBM%20Prerequisite%20Scanner>

Create a non-root user

You must use a non-root user to install IBM Operations Analytics - Log Analysis.

Procedure

If you do not have a non-root user defined on your system, you must create one. To create this user, log in as a root user and run the command:

```
useradd -m -d /home/<username> <username>
```

where `-m` creates a home directory for your user if one does not exist, `-d` is the path to the home directory, and `<username>` is the user name that you want to create.

Ensure that you have the necessary access rights so that you can add files to the location where you want to install IBM Operations Analytics - Log Analysis.

Recommended library installers

The following libraries are recommended for use with IBM Operations Analytics - Log Analysis.

The YUM (Yellow dog update, modified) package manager is recommended for installing the required libraries on Red Hat Enterprise Linux (RHEL).

The YaST (Yet another Setup Tool) tool is recommended for installing the required libraries on SUSE Linux Enterprise Server (SLES).

Verifying the operating system version

IBM Operations Analytics - Log Analysis requires Red Hat Enterprise (RHEL) for Linux or SUSE Linux Enterprise Server (SLES).

About this task

For information about Red Hat Enterprise (RHEL) for Linux and SUSE Linux Enterprise Server (SLES) versions, see <https://developer.ibm.com/itoa/docs/log-analysis/documentation/hardware-software-requirements/>.

Procedure

- To verify that the correct version of RHEL is installed, log in as a root user and enter the following command:

```
cat /etc/redhat-release
```

Note: If your version is not supported, you must upgrade to one of the supported versions.

- To verify that the correct version of SLES is installed, log in as a root user and enter the following command:

```
cat /etc/SuSE-release
```

Note: If your version is not supported, you must upgrade to one of the supported versions.

Verifying the 64-bit library requirement

For Red Hat Enterprise Linux, IBM Operations Analytics - Log Analysis requires the 64-bit compat-libstdc++ library:

Procedure

- To determine if the required libraries are installed, run the command:

```
sudo /usr/bin/yum --noplugins list installed "libstdc++"
```

If this command indicates that the required libraries are installed, no additional action is required. For example:

```
libstdc++.x86_64
```

- If the required libraries are not installed, search the Red Hat Network repository to list the libraries that are available for your operating system:

```
sudo /usr/bin/yum --noplugins search libstdc++
```

- To install the required libraries, run this command:

```
sudo /usr/bin/yum --noplugins install libstdc++.x86_64
```

Disabling Security-Enhanced Linux (SELinux)

If SELinux is in enforcing mode, an exception occurs during the installation of IBM Operations Analytics - Log Analysis. Ensure that the SELinux policy is set to a permissive or disabled state. To disable the SELinux:

Procedure

- Log in as a root user.
- Edit the config file that is in the `/etc/selinux/` directory.
- Change the SELINUX value to a permissive or disabled state. The possible values are:

permissive

SELinux prints warnings instead of enforcing them.

disabled

SELinux is fully disabled.

For example, to disable the kernel, change the value to disabled:

```
SELINUX=disabled
```

4. Save your changes.
5. Restart the operating system.

Verifying KornShell library

Verify that the KornShell library is part of your operating system.

Procedure

1. To verify that KornShell is part of your operating system, enter one of the following commands:

```
usr/bin/ksh
```

or

```
/bin/ksh
```

2. If these commands do not work, enter the following command to confirm that KornShell is installed:

```
rpm -qa | grep ksh  
ksh-20100202-1.el5
```

3. If KornShell is not installed, download it and use the following command to install it:

```
rpm -ivh ksh-<version>.rpm
```

where <version> is the version that you downloaded.

Verifying the Python version

Python Version 2.4.3 and 2.6.6 to 2.6.8 are supported by IBM Operations Analytics - Log Analysis.

If you did not use an rpm package to install Python, the IBM Operations Analytics - Log Analysis installer might not recognize it and a warning message might be displayed. This warning can be ignored and the installation continues.

To verify that you are using the correct version of Python, enter the following command:

```
rpm -qa | grep "^python-2"
```

If successful, the command returns the version of Python. For example:

```
python-2.4.3-27.el5
```

If you are not using the correct version of Python, download and install it.

Install python-simplejson package for Python

IBM Operations Analytics - Log Analysis requires Python Version 2.4.3 including the python-simplejson rpm or Python Version 2.6.6 to 2.6.8. If you use Python Version 2.4.3, you must also install the simple JSON rpm, python-simplejson. Python Version 2.6.6 to 2.6.8 include the required rpm.

Procedure

1. Download the python-simplejson package from http://pkgs.org/centos-5-rhel-5/centos-rhel-x86_64/python-simplejson-2.0.9-8.el5.x86_64.rpm/download/.

2. Log in to the IBM Operations Analytics - Log Analysis server as a root user.
3. Change directory to the folder that contains the package.
4. Run the following command:

```
rpm -i python-simplejson-2.0.9-8.el5.x86_64.rpm
```

Installing unzip utility

You must install the unzip utility on any servers where you install IBM Operations Analytics - Log Analysis or remote Indexing Engine.

Procedure

- To install the utility on Red Hat Enterprise Linux (RHEL), log in as a root user and run the following command:

```
yum install unzip
```

- To install the utility on SUSE Enterprise Linux Server (SELS), log in as a root user and run the following command:

```
zypper install unzip
```

Verifying the host server IP address and names

Before you install IBM Operations Analytics - Log Analysis, you must ensure that the details for each host server are maintained correctly in the `etc/hosts` directory on the target system.

About this task

If you do not complete this task, you may encounter issues when you log in or run a Custom Search Dashboard. For more information, see *Cannot run Custom Search Dashboards after IP address change* in the *Troubleshooting Guide*.

Procedure

- For a server that uses a static IP address, define the static IP address and the required values in the following format:

IP	LONG-HOSTNAME	SHORT-HOSTNAME
----	---------------	----------------

For example:

9.124.111.162	scaserver1.example.com	scaserver1
---------------	------------------------	------------

- For a Dynamic Host Configuration Protocol (DHCP) server that uses a loop back IP address, define the loop back IP address and the required values in the following format:

LOOPBACK-IP	LONG-HOSTNAME	SHORT-HOSTNAME
-------------	---------------	----------------

For example:

127.0.0.1	ibmscala.example.com	ibmscala
-----------	----------------------	----------

Number of open files and virtual memory limits

Update your open files and virtual memory limits to match the recommended values.

The recommended value of the `ulimit -n` setting, which governs number of open files that are allowed for a process, is 4096.

The recommended value of the `ulimit -v` setting, which governs the virtual memory limit for a process, is unlimited.

For more information, see the Performance and Tuning Guide at: <https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBMLogAnalyticsBeta/page/PerformanceandTuningGuide>

Changing the default number of open files limit

The recommended `ulimit -n` setting, which governs the number of open files that are allowed for a process, is 4096 on each of the servers where a Indexing Engine node is installed. To modify the number of files open limit, complete the following steps:

Procedure

1. Log in to the server where a Indexing Engine node is installed using the Indexing Engine **user id** for that server.
2. Run the following command in a terminal window:

```
ulimit -n
```

If the returned value is less than 4096, proceed to step 3. If the returned value is more than 4096, proceed to step 7.

3. Log in to the server as the **root user**.
4. Open the `/etc/security/limits.conf` file.
5. Modify or add the following lines:

```
<user-id> hard nfile 4096
<user-id> soft nfile 4096
```

where `<user-id>` is the `user id` used to install the Indexing Engine on the server.

To modify the value for all users on this server, modify or add the following lines:

```
* hard nfile 4096
* soft nfile 4096
```

6. Save your changes.
7. Repeat for each Indexing Engine instance.

What to do next

To ensure that the changes are updated, restart IBM Operations Analytics - Log Analysis.

Changing the virtual memory limit

The recommended `ulimit -v` setting, which limits the virtual memory for processes, is unlimited on each of the servers where a Indexing Engine node is installed. To modify the limit, complete the following steps:

Procedure

1. Log in to the server where a Indexing Engine node is installed using the Indexing Engine **user id** for that server.
2. Run the following command in a terminal window:

```
ulimit -v
```

If the returned value is not unlimited, proceed to step 3. If the returned value is unlimited, proceed to step 7.

3. Log in to the server as the **root user**.
4. Open the `/etc/security/limits.conf` file.
5. Modify or add the following lines:

```
<user-id> hard as unlimited
<user-id> soft as unlimited
```

where <user-id> is the user id used to install the Indexing Engine on the server.

To modify the value for all users on this server, modify or add the following lines:

```
* hard as unlimited
* soft as unlimited
```

6. Save your changes.
7. Repeat for each Indexing Engine instance.

What to do next

To ensure that the changes are updated, restart IBM Operations Analytics - Log Analysis.

Verifying the IP address and host name configurations

You need to verify that the IP address and host name settings are configured correctly.

Procedure

1. To verify that the host name is configured correctly, enter the following command:

```
hostname
```

If it is configured correctly, the command returns the host name. For example:

```
example
```

2. To verify that the host name uses the fully qualified host name, enter the following command:

```
hostname -f
```

If successful, the command returns the fully qualified host name. For example:

```
example.ibm.com
```

3. To confirm that the IP address is configured correctly, ping the host name:

```
ping example
```

If successful, the IP address is returned.

Time server

To ensure that some of the features of IBM Operations Analytics - Log Analysis work correctly, you need to set up a time keeping server.

Some features of IBM Operations Analytics - Log Analysis use relative time such as days, weeks, or months. This time is calculated from the time on the server on which IBM Operations Analytics - Log Analysis is installed. To ensure that these features generate correct results, you must ensure that the time on the IBM Operations Analytics - Log Analysis server is correct. You can use a time server to synchronize the time on the IBM Operations Analytics - Log Analysis server with the correct time. You must do this before you install IBM Operations Analytics - Log Analysis.

Installing on IBM Power8 servers

Before you install Log Analysis on a IBM Power8 based server, read and complete the following prerequisites.

The supported version is IBM Power8 Little Endian for Linux (ppc64le). This is referred to as IBM Power8 in the this documentation.

Prerequisites

- Before you install Log Analysis, ensure that you install at least one font package. You can check the `/usr/share/fonts` directory to see what packages are installed. One font package that you can use is called `gnu-free-fonts-common-20120503-8.ael7b.noarch`.
- You must use Red Hat Enterprise for Linux 7.1 as your operating system.

Supported features

If you install Log Analysis on IBM Power8 based hardware, you cannot use the internal IBM Tivoli Monitoring Log File Agent that is installed with other versions of Log Analysis. It is not installed as part of the installation on IBM Power8 based servers.

However, you can still configure an external IBM Tivoli Monitoring Log File Agent to stream data from a remote source to your installation of Log Analysis. For more information, see [Chapter 6, “Loading and streaming data,”](#) on page 149.

All the other features of Log Analysis are available.

Installing on Linux on System z and Linux on System x based servers

Before you can install IBM Operations Analytics - Log Analysis on a Linux on System z or Linux on System x based operating system, read and complete the prerequisites.

Hardware requirements

Linux on System z runs on System z based hardware. Linux on System x runs on Intel or AMD-based hardware. Both types of hardware are supported but there are some minor differences in the software requirements.

Note: If you install IBM Operations Analytics - Log Analysis on Intel or AMD-based hardware, you must install IBM Operations Analytics - Log Analysis components like the Indexing Engine server on Intel or AMD-based hardware. You cannot install IBM Operations Analytics - Log Analysis components on Linux on System z based hardware. Likewise, if you install IBM Operations Analytics - Log Analysis on Linux on System z based hardware, you must install IBM Operations Analytics - Log Analysis components like the Indexing Engine server on System z based hardware.

For more information about the hardware requirements, see [Hardware and software requirements](#).

Prerequisites

For more information about the prerequisite tasks that you must complete, see [“Prerequisites”](#) on page 6.

Supported operating systems for cross-platform data ingestion

You can choose one of the following methods for loading data into IBM Operations Analytics - Log Analysis:

1. Using the internal IBM Tivoli Monitoring Log File Agent that is bundled with IBM Operations Analytics - Log Analysis to stream data.
2. Using an external IBM Tivoli Monitoring Log File Agent which you install separately to stream data.
3. Using the z/OS® Log Forwarder that is bundled with the z/OS Insight Packs.
4. Using the data collector client or FTP to load a batch of data.

Each of these scenarios offers varying cross-platform support as outlined in the following table:

Table 2. Supported operating systems for data loading scenarios	
Data loading scenario	Supported operating systems
1	See the <i>Supported Operating Systems</i> section in Chapter 6, “Loading and streaming data,” on page 149
2	See the <i>Requirements for the monitoring agent</i> topic documentation for your version of IBM Tivoli Monitoring at: https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/Tivoli%20Documentation%20Central/page/Tivoli%20Monitoring
3	See the <i>System requirements for the z/OS Log Forwarder</i> topic in the product documentation that is linked in the <i>Insight Packs</i> section
4	See the <i>Supported Operating Systems</i> section in Chapter 6, “Loading and streaming data,” on page 149

Note:

The remote installer that you use to install instances of the IBM Tivoli Monitoring Log File Agent and the Tivoli Event Integration Facility does not support cross operating system integration. You must use the remote installers to install remote instances on servers that use the same operating system. For example, if you install IBM Operations Analytics - Log Analysis on Linux on System z, you must install the remote instances on Linux on System z. In this example, you cannot install remote instances on Linux on System x.

Insight Packs for z/OS

Domain insights for z/OS subsystems are available separately. You can install z/OS Insight Packs on both Linux on System z and Linux on System x based servers. For more information, see [Documentation for IBM Operations Analytics for z Systems](#).

Linux on System z logstash support

logstash 1.5.3 is bundled with the IBM Operations Analytics - Log Analysis Linux on System z 64-bit installation package. However, logstash is not supported on Linux on System z. You can only use logstash to facilitate log aggregation on Linux x86 based hardware.

Cross-platform installation of logstash is not supported. For example, you cannot install logstash remotely from a Linux on System z server to another Linux on System z or Linux on System x based server.

Installing on non-English-language systems

If you want to install IBM Operations Analytics - Log Analysis on a system where the locale is not set to English United States, you need to set the locale of the command shell to export `LANG=en_US.UTF-8` before you run any IBM Operations Analytics - Log Analysis scripts. For example, when you install IBM Operations Analytics - Log Analysis you need to run the following command to change the locale before you run the `unity.sh` script:

```
export LANG=en_US.UTF-8
```

Tuning the operating system

To tune the Linux operating system, ensure that the following resource configuration settings are made:

Number of concurrent files: 4096

Use the `ulimit -n` command to change this setting to 4096.

Virtual memory: Unlimited

Use the `ulimit -v` command to change this setting to unlimited.

Installing

Before you use the command-line interface, the Installation Manager UI, or a silent installation to install IBM Operations Analytics - Log Analysis, read the prerequisites.

Log Analysis includes IBM Installation Manager Version 1.8.2. You can use this version to install Log Analysis immediately. You can also use an existing version of IBM Installation Manager to install Log Analysis. For more information, see the product documentation at http://www-01.ibm.com/support/knowledgecenter/SSDV2W/im_family_welcome.html.

Prerequisites

- Ensure that you complete all the prerequisites. For more information, see “Prerequisites” on page 6.
- Ensure that your user has the access rights that are required to add files to the location where you want to install IBM Operations Analytics - Log Analysis.
- If you previously installed IBM Tivoli Monitoring Log File Agent 6.3 or lower, the installation fails. To solve this problem, stop the existing IBM Tivoli Monitoring Log File Agent installation or rename the folder that was created when it was installed. For detailed information, see the topic about the installation failure if IBM Tivoli Monitoring Log File Agent was installed in the *Troubleshooting IBM Operations Analytics - Log Analysis* guide.
- Before you install IBM Operations Analytics - Log Analysis, you must ensure that the details for each host server are maintained correctly in the `/etc/hosts` directory on the target system. Failure to complete this task can result in OAuth errors when you run a Custom Search Dashboard. For more information, see “Verifying the IP address and host name configurations” on page 12.
- Ensure that IBM Installation Manager is not configured to search the service repositories. If the server is not connected to the internet, this setting can cause the installation to fail or stall.
- For more information about IBM Installation Manager, see the IBM Knowledge Center at https://www-01.ibm.com/support/knowledgecenter/SSDV2W/im_family_welcome.html.

Installing with the IBM Installation Manager UI

You must complete the following steps to install Log Analysis using the IBM Installation Manager User Interface (UI).

Before you begin

- Ensure that the **Search service repositories during installation and updates** check box is not selected. If you do select this check box and the server is not connected to the internet, the installation can stall or fail.

About this task

When you run the installation script, Log Analysis and IBM Installation Manager are installed. IBM Installation Manager Version 1.8.2 is installed when you install Log Analysis. Where necessary, Log Analysis upgrades the currently installed version of IBM Installation Manager.

Note: The installation path accepts ASCII characters only. Other characters, such as native language characters are not supported.

Procedure

1. Copy and extract the installation archive to a location on your server.

2. From the directory to which you extracted the installation files, run the command:

```
./install.sh
```

Note: Add a `-c` parameter to start the installation in console only mode. For more information, see [“Installing with the IBM Installation Manager command-line interface”](#) on page 17.

Note: To install Log Analysis on a remote server using GUI mode, ensure that virtual desktop software is installed on the server that you want to install Log Analysis on.

3. The Install Packages screen is displayed.
4. To install Log Analysis in the default directory, click **Next**. To install Log Analysis to a different location, click **Browse**, select an alternative location, and click **Next**.
5. To continue, accept the license agreement, and click **Next**.
6. To accept the default IBM Tivoli Monitoring Log File Agent, Indexing Engine, and Log Analysis options, click **Next**.

Note: If you cannot install the IBM Tivoli Monitoring Log File Agent or the check box is grayed out, you are either missing some of the prerequisites or the IBM Tivoli Monitoring Log File Agent is already installed on the same server. You need to complete one of the following actions:

- Check that you have meet all the required prerequisites.
 - If you do not want to use the existing IBM Tivoli Monitoring Log File Agent, you can stop the install and remove it.
 - If you want to use the IBM Tivoli Monitoring Log File Agent to load data into Log Analysis, you can continue with the installation, after you complete it, you need to configure the IBM Tivoli Monitoring Log File Agent.
7. The default ports that are used by Log Analysis are displayed. Accept the default option if these ports are not in use by any other application or change them if necessary. Click **Next**.
 8. If you want to install a local Indexing Engine instance, ensure that the **Apache Solr** check box is selected. The check box is selected by default. If you want to use a local Indexing Engine installation, you must install it now. You cannot install it after the Indexing Engine is installed. However, you can install instances of Indexing Engines on remote servers after the installation is completed. To enable search and indexing, you must install at least one Indexing Engine instance locally or remotely.
 9. Review the summary information that is provided for the installation and click **Install**.
 10. To complete the installation click **Finish**.

What to do next

Download a license for IBM Operations Analytics - Log Analysis from Passport Advantage® at <http://www-01.ibm.com/software/lotus/passportadvantage/> and complete the steps that are outlined in the readme file.

To verify that the installation is complete, log in to Log Analysis.

To install Indexing Engines on remote servers, you can create them after you install Log Analysis.

If you do not verify the server details as described in the *Prerequisites* topic, the following error message is displayed when you try to run a Custom Search Dashboard:

```
Failed to launch. Could not retrieve OAuth access token.
```

To correct this error, you must ensure that the server details are correct and start the installation again. For more information, see *Could not retrieve OAuth access token* in the *Troubleshooting* section.

Related concepts

[“Prerequisites”](#) on page 6

Before you install IBM Operations Analytics - Log Analysis, ensure that the system meets the hardware and software requirements and complete the prerequisite tasks.

[“Installing and configuring the IBM Tivoli Monitoring Log File Agent”](#) on page 21

When you install IBM Operations Analytics - Log Analysis, you can also choose to install the IBM Tivoli Monitoring Log File Agent (LFA) that is delivered with the product.

[“Logging in to IBM Operations Analytics - Log Analysis” on page 307](#)

This topic outlines how to log in to IBM Operations Analytics - Log Analysis and how to change the default username and password.

Related tasks

[“Installing Apache Solr on remote machines” on page 64](#)

After you install IBM Operations Analytics - Log Analysis, you can use the Apache Solr remote installer to install Apache Solr on a remote machine.

Installing with the IBM Installation Manager command-line interface

You can use the IBM Installation Manager command-line interface. to install Log Analysis.

Before you begin

- Do not enter `Control -C` to cancel the installation because this setting can cause the installer to behave inconsistently. Instead, to cancel the installation, enter `c` when prompted.

About this task

When you run the installation script, Log Analysis and IBM Installation Manager are installed. IBM Installation Manager Version 1.8.2 is installed when you install Log Analysis. Where necessary, Log Analysis upgrades the currently installed version of IBM Installation Manager.

Note: The installation path accepts ASCII characters only. Other characters, such as native language characters are not supported.

Procedure

1. Copy and extract the installation archive to a location on your server.
2. To install Log Analysis:

- a) From the directory location of the extracted installation files, run the command:

```
./install.sh -c
```

- b) Select the default IBM Log File Agent, Indexing Engine, and Log Analysis packages.

Note: If you cannot install the IBM Tivoli Monitoring Log File Agent or the check box is grayed out, you are either missing some of the prerequisites or the IBM Tivoli Monitoring Log File Agent is already installed on the same server. You need to complete one of the following actions:

- Check that you have meet all the required prerequisites.
 - If you do not want to use the existing IBM Tivoli Monitoring Log File Agent, you can stop the install and remove it.
 - If you want to use the IBM Tivoli Monitoring Log File Agent to load data into Log Analysis, you can continue with the installation, after you complete it, you need to configure the IBM Tivoli Monitoring Log File Agent.
- c) Accept the license agreement.
 - d) If required, change the installation location. Otherwise, accept the default location.
 - e) Choose the Log Analysis feature and, if required, the IBM Tivoli Monitoring Log File Agent feature.
 - f) If necessary, change the default port numbers. Otherwise, accept the default ports.
 - g) To install a local Indexing Engine instance, ensure that **Apache Solr** is selected. **Apache Solr** is selected by default. To use a local Indexing Engine installation, you must install it now. You cannot install it after the IBM Operations Analytics - Log Analysis is installed. However, you can install Indexing Engine instances on remote machines after the installation is completed. To enable search and indexing, you must install at least one Indexing Engine instance locally or remotely.

- h) If you want to generate an installation response file for future silent installations, select **Generate an Installation Response File**.
 - i) When the installation is complete, select **Finish**.
3. To check the details of the installation, choose **View Installed Packages**.

What to do next

Download a license for IBM Operations Analytics - Log Analysis from Passport Advantage at <http://www-01.ibm.com/software/lotus/passportadvantage/> and complete the steps that are outlined in the readme file.

To verify that the installation is complete, log in to Log Analysis.

To install Indexing Engines on remote servers, you can create them after you install Log Analysis.

If you do not verify the server details as described in the *Prerequisites* topic, the following error message is displayed when you try to run a Custom Search Dashboard:

```
Failed to launch. Could not retrieve OAuth access token.
```

To correct this error, you must ensure that the server details are correct and start the installation again. For more information, see *Could not retrieve OAuth access token* in the *Troubleshooting* section.

Related concepts

[“Prerequisites” on page 6](#)

Before you install IBM Operations Analytics - Log Analysis, ensure that the system meets the hardware and software requirements and complete the prerequisite tasks.

[“Installing and configuring the IBM Tivoli Monitoring Log File Agent” on page 21](#)

When you install IBM Operations Analytics - Log Analysis, you can also choose to install the IBM Tivoli Monitoring Log File Agent (LFA) that is delivered with the product.

[“Logging in to IBM Operations Analytics - Log Analysis” on page 307](#)

This topic outlines how to log in to IBM Operations Analytics - Log Analysis and how to change the default username and password.

Related tasks

[“Installing Apache Solr on remote machines” on page 64](#)

After you install IBM Operations Analytics - Log Analysis, you can use the Apache Solr remote installer to install Apache Solr on a remote machine.

Silently installing Log Analysis

You can install Log Analysis silently by using the sample response file that is provided with the product. This automates the installation procedure.

Before you begin

- Download and extract the Log Analysis installation archive. The archive contains the product files and a sample response file, `sample_smcl_silent_install.xml`, that is required for silent installation.

Note: A silent installation can fail if the IBM Installation Manager repository changed since the last installation or uninstall. This problem can occur even when you update the response file with the correct repository location. If you are installing from a new repository, remove or close any old repository connections.

Note: If you cannot install the IBM Tivoli Monitoring Log File Agent or the check box is grayed out, you are either missing some of the prerequisites or the IBM Tivoli Monitoring Log File Agent is already installed on the same server. You need to complete one of the following actions:

- Check that you have met all the required prerequisites.
- If you do not want to use the existing IBM Tivoli Monitoring Log File Agent, you can stop the install and remove it.

- If you want to use the IBM Tivoli Monitoring Log File Agent to load data into Log Analysis, you can continue with the installation, after you complete it, you need to configure the IBM Tivoli Monitoring Log File Agent.

About this task

When you run the installation script, Log Analysis and IBM Installation Manager are installed. IBM Installation Manager Version 1.8.2 is installed when you install Log Analysis. Where necessary, Log Analysis upgrades the currently installed version of IBM Installation Manager.

Note: The installation path accepts ASCII characters only. Other characters, such as native language characters are not supported.

Silently installing Log Analysis involves modifying the sample response file and then calling IBM Installation Manager from the command line or from a script to install the product. IBM Installation Manager obtains the installation settings from the response file.

Procedure

To silently install Log Analysis:

1. Copy the sample response file to a suitable local directory (for example, your home directory). Use a suitable name for the local response file, for example:
smcl_silent_install.xml
2. Modify the response file to suit your environment:
 - a) Locate the line and edit this line to reflect your home directory:

```
<preference name='com.ibm.cic.common.core.preferences.eclipseCache'
value='/home/MYUSERID/IBM/IBMIMShared' />
```

where /home/MYUSERID is the location of your home directory.

- b) Locate and edit these lines to specify the directory where you extracted the installation image:

```
<repository location='/home/MYUSERID/IMAGEDIRECTORY/im.linux.x86' />
<repository location='/home/MYUSERID/IMAGEDIRECTORY' />
```

where /home/MYUSERID is your home directory and IMAGEDIRECTORY is the name of the directory to which you extracted the installation package.

Note: The paths that are given assume that you extracted the installation package to your home directory.

- c) Locate and edit these lines to reflect your home directory:

```
<profile id='IBM Installation Manager' installLocation=
'/home/MYUSERID/IBM/InstallationManager/eclipse' kind='self'>
<data key='eclipseLocation' value='/home/MYUSERID/IBM/
InstallationManager/eclipse' />
```

and

```
<profile id='IBM Log Analytics'
installLocation='/home/MYUSERID/IBM/LogAnalysis'>
```

and

```
<data key='eclipseLocation' value='/home/MYUSERID/IBM/
LogAnalysis' />
```

where /home/MYUSERID is the location of your home directory.

- d) If necessary, change the following default port numbers:

Note: Default port numbers are used by Log Analysis, only modify the values if necessary.

```

<!-- Application WebConsole Port -->
<data key='user.unity.port.number,com.ibm.tivoli.scloganalytics'
value='9988' />
<!-- Application WebConsole Secure Port -->
<data key='user.unity.secureport.number,com.ibm.tivoli.scloganalytics'
value='9987' />
<!-- Database Server Port -->
<data key='user.database.port.number,com.ibm.tivoli.scloganalytics'
value='1627' />
<!-- Data Collection Server Port -->
<data key='user.eif.port.number,com.ibm.tivoli.scloganalytics'
value='5529' />
<!-- ZooKeeper Port -->
<data key='user.zookeeper.port.number,com.ibm.tivoli.scloganalytics'
value='12181' />
<!-- Apache Solr Search Port -->
<data key='user.searchengine.port.number,com.ibm.tivoli.scloganalytics'
value='9983' />
<!-- Apache Solr Stop Port -->
<data key='user.searchengineQS.port.number,com.ibm.tivoli.scloganalytics'
value='7205' />

```

e) Save your changes.

3. To exclude IBM Tivoli Monitoring Log File Agent from the installation, remove the LOG_FILE_AGENT parameter from the offering id element.

For example, change the following default entry:

```

<offering id='com.ibm.tivoli.scloganalytics'
profile='IBM Log Analytics'
features='IBM Log Analytics,LOG_FILE_AGENT'
installFixes='none' />

```

to:

```

<offering id='com.ibm.tivoli.scloganalytics'
profile='IBM Log Analytics'
features='IBM Log Analytics'
installFixes='none' />

```

4. To exclude Indexing Engine, remove the Data Explorer Application parameter from the offering id element.

For example, change the following default entry:

```

<offering id='com.ibm.tivoli.scloganalytics'
profile='IBM Log Analytics'
features='IBM Log Analytics,LOG_FILE_AGENT,
Data Explorer Application'
installFixes='none' />

```

to:

```

<offering id='com.ibm.tivoli.scloganalytics'
profile='IBM Log Analytics'
features='IBM Log Analytics, LOG_FILE_AGENT,'
installFixes='none' />

```

5. If you already have IBM Installation Manager installed, use this command to start the silent installation:

```

~/IBM/InstallationManager/eclipse/tools/imcl -s -input <HOME_DIR>/
smcl_silent_install.xml -sVP -acceptLicense

```

Where <HOME_DIR> is the directory where you stored the response file.

If you do not have IBM Installation Manager installed, use the `install.sh` command to install both IBM Installation Manager and Log Analysis. From the directory to which you extracted the installation archive, run the command:

```
./install.sh -s <HOME_DIR>/smcl_silent_install.xml
```

where `<HOME_DIR>` is your home directory. This command silently installs Log Analysis and IBM Installation Manager Version 1.8.2, if no other version of IBM Installation Manager is installed.

Results

The progress of the installation is displayed in an IBM Installation Manager console. To install without displaying the console, leave out the `-sVP` option (which shows Verbose Progress).

What to do next

Download a license for IBM Operations Analytics - Log Analysis from Passport Advantage at <http://www-01.ibm.com/software/lotus/passportadvantage/> and complete the steps that are outlined in the readme file.

To verify that the installation is complete, log in to Log Analysis.

To install Indexing Engines on remote servers, you can create them after you install Log Analysis.

If you do not verify the server details as described in the *Prerequisites* topic, the following error message is displayed when you try to run a Custom Search Dashboard:

```
Failed to launch. Could not retrieve OAuth access token.
```

To correct this error, you must ensure that the server details are correct and start the installation again. For more information, see *Could not retrieve OAuth access token* in the *Troubleshooting* section.

Related concepts

[“Prerequisites” on page 6](#)

Before you install IBM Operations Analytics - Log Analysis, ensure that the system meets the hardware and software requirements and complete the prerequisite tasks.

[“Installing and configuring the IBM Tivoli Monitoring Log File Agent” on page 21](#)

When you install IBM Operations Analytics - Log Analysis, you can also choose to install the IBM Tivoli Monitoring Log File Agent (LFA) that is delivered with the product.

[“Logging in to IBM Operations Analytics - Log Analysis ” on page 307](#)

This topic outlines how to log in to IBM Operations Analytics - Log Analysis and how to change the default username and password.

Related tasks

[“Installing Apache Solr on remote machines” on page 64](#)

After you install IBM Operations Analytics - Log Analysis, you can use the Apache Solr remote installer to install Apache Solr on a remote machine.

Installing and configuring the IBM Tivoli Monitoring Log File Agent

When you install IBM Operations Analytics - Log Analysis, you can also choose to install the IBM Tivoli Monitoring Log File Agent (LFA) that is delivered with the product.

After you install the LFA during the Log Analysis installation, you need to configure it. This type of LFA is referred to as an internal LFA.

If you chose not to install the LFA during the Log Analysis installation because you want to use an existing LFA to load data into Log Analysis, you need to configure your LFA to work with Log Analysis. This type of LFA is referred to as an external LFA.

For more information about how to configure the LFA, see [“Streaming data with the IBM Tivoli Monitoring Log File Agent” on page 186](#).

Related concepts

[“IBM Tivoli Monitoring Log File Agent configuration scenarios” on page 188](#)

You can use the internal IBM Tivoli Monitoring Log File Agent that is installed with IBM Operations Analytics - Log Analysis or you can use an external IBM Tivoli Monitoring Log File Agent to stream data from local or remote servers.

Verifying the version information

To display the version information for your installation, run the `unity_VersionInfoUtility.sh` utility.

Procedure

Run the following command:

```
<HOME>/IBM/LogAnalysis/utilities/unity_VersionInfoUtility.sh
```

Results

The version information is displayed along with some other useful information about your install.

Removing IBM Operations Analytics - Log Analysis

You can use the command-line, the IBM Installation Manager UI or a silent removal process to remove IBM Operations Analytics - Log Analysis.

Removing IBM Operations Analytics - Log Analysis

This topic outlines the steps that you must complete to remove IBM Operations Analytics - Log Analysis.

About this task

This procedure outlines how to remove IBM Operations Analytics - Log Analysis and IBM Installation Manager from your environment. Both of these components are installed when you install IBM Operations Analytics - Log Analysis. To complete the uninstallation, remove IBM Operations Analytics - Log Analysis and then, if required, complete the procedure to remove IBM Installation Manager.

Note: If you have remote installations of Indexing Engines, you must remove these before you remove IBM Operations Analytics - Log Analysis. For more information about removing Apache Solr Indexing Engines, see [“Removing Apache Solr instances” on page 65](#)

Procedure

1. To remove IBM Operations Analytics - Log Analysis:

- a) Navigate to the `<HOME>/IBM/InstallationManager/eclipse/launcher` directory and execute the command:

```
./launcher
```

Note: If you are accessing the installation environment remotely, ensure that your virtual desktop software is configured to allow you to view the graphical user interface for the IBM Installation Manager.

- b) Click **Next**.
 - c) Select the IBM Operations Analytics - Log Analysis package and click **Next**.
 - d) Click **Uninstall**. Allow the removal to proceed and when complete, click **Finish**.
2. (Optional) To remove IBM Installation Manager:

- a) From the <HOME>/var/ibm/InstallationManager/uninstall directory, execute the command:

```
./uninstall
```

where <HOME> is the directory to which you have installed IBM Operations Analytics - Log Analysis

- b) Complete the uninstallation steps and click **Finish**.

Using the console to remove IBM Operations Analytics - Log Analysis

This topic outlines the steps that you must complete to remove IBM Operations Analytics - Log Analysis using the console.

About this task

This procedure outlines how to remove IBM Operations Analytics - Log Analysis and IBM Installation Manager from your environment using the console. Both of these components are installed when you install IBM Operations Analytics - Log Analysis. To complete the uninstallation, remove IBM Operations Analytics - Log Analysis and then, if required, complete the procedure to remove IBM Installation Manager.

Note: If you have remote installations of Indexing Engines, you must remove these before you remove IBM Operations Analytics - Log Analysis. For more information about removing Apache SolrIndexing Engines, see [“Removing Apache Solr instances”](#) on page 65.

Procedure

1. To remove IBM Operations Analytics - Log Analysis:

- a) Navigate to the <HOME>/InstallationManager/eclipse/tools directory and execute the command:

```
./imcl -c
```

- b) Enter 5 and press Enter.
- c) Enter 1 to select the IBM Operations Analytics - Log Analysis package group and press N to proceed.
- d) Enter 1 to select the IBM Operations Analytics - Log Analysis package, press Enter.
- e) Enter N to proceed.
- f) Enter U to start the uninstallation.
- g) After the uninstallation has completed, enter F to complete the uninstallation.
- h) Enter X to close the IBM Installation Manager.

2. (Optional) To remove IBM Installation Manager:

- a) From the <HOME>/var/ibm/InstallationManager/uninstall directory, execute the command:

```
./uninstallc
```

- b) The uninstallation proceeds and completes.

Silently removing IBM Operations Analytics - Log Analysis

You can remove IBM Operations Analytics - Log Analysis silently by using the sample response file that is provided with the product.

Before you begin

You require the sample response file <HOME>/IBM/LogAnalysis/work_files/removal/sample_silent_loganalytics_removal.xml.

Note: A silent removal can fail if the IBM Installation Manager repository changed since the last installation or removal. This problem can occur even when you update the response file with the correct repository location. If the repository changed, remove or close the old repository connections before you remove the product.

Note: If you have remote installations of Apache Solr, remove them before you remove IBM Operations Analytics - Log Analysis. For more information about how to do this, see [“Removing Apache Solr instances”](#) on page 65.

About this task

Silently removing IBM Operations Analytics - Log Analysis involves modifying the sample response file and then calling IBM Installation Manager from the command line or from a script to remove the product. IBM Installation Manager obtains the required settings from the response file.

Procedure

To silent removal the product:

1. Copy the sample response file to a suitable local directory (for example, your home directory). Use a suitable name for the local response file, for example:
smcl_removal.xml
2. Modify the response file to suit your environment:
 - a) Specify the IBM Installation Manager repository location for your environment. For example:
repository location='/home/smcl/smcl_build/'
 - b) If you changed the default IBM Operations Analytics - Log Analysis port numbers during or after installation, update the following entries with the new port numbers:

```
<!-- IBM Log Analytics WebConsole Port Number -->
<data key='user.unity.port.number,com.ibm.tivoli.scloganalytics'
value='9988' />

<!-- IBM Log Analytics WebConsole Secure Port Number -->
<data key='user.unity.secureport.number,com.ibm.tivoli.scloganalytics'
value='9987' />

<!-- IBM Log Analytics Database Port Number -->
<data key='user.database.port.number,com.ibm.tivoli.scloganalytics'
value='1627' />

<!-- IBM Log Analytics DataCollection (EIF) Server Port Number -->
<data key='user.eif.port.number,com.ibm.tivoli.scloganalytics'
value='5529' />

<!-- IBM Log Analytics Search Engine WebConsole Server Port Number -->
<data key='user.searchengine.port.number,com.ibm.tivoli.scloganalytics'
value='9989' />

<!-- IBM Log Analytics Distributed Application Management Server
Port Number -->
<data key='user.zookeeper.port.number,com.ibm.tivoli.scloganalytics'
value='12181' />
```

- c) Save your changes.
3. Use the following command to removal the product:
~/IBM/InstallationManager/eclipse/tools/imcl -s -input <HOME_DIR>/
smcl_removal.xml -sVP -acceptLicense

Where <HOME_DIR> is the directory where you stored the response file.

Results

The progress of the removal is displayed in an IBM Installation Manager console. To run the removal script displaying the console, omit the -sVP option (which shows Verbose Progress).

Installing reference

Read the information about the scripts and properties that you can configure when you install Log Analysis.

Configuration properties file

After you complete the installation, to modify the configuration properties edit the `unitysetup.properties` file.

Modify only the properties that are listed in the table.

Note: You cannot change any of the other parameters in the file. These parameters are identified as such in the file. If you do change one of these parameters, IBM Operations Analytics - Log Analysis may not work correctly or at all.

Table 3. <i>unitysetup.properties</i> parameters	
Parameter	Description
MAX_SEARCH_RESULTS=1000	The maximum number of search results that can be returned by the search query.
MAX_DATA_FACETS_IN_CHART=1000	The maximum number of facets or data points that can be included in returned search results.
#SEARCH_QUERY_FOR_DEFAULT_SEARCH=*	Uncomment this parameter to enable the default search.
MAX_WORK_QUEUE_SIZE=100000	Determines the maximum size of the queue of documents that are being held for annotation and indexing.
NUM_DOCUMENT_PROCESSOR_THREADS=30	Determines the thread pool that is used to select, annotate, and index a document.
batchsize=500000	Determines the maximum number of records that can be loaded by the Generic Receiver in single batch.
UNITY_APP_TIME_OUT=180	Determines the default time-out value in seconds for the apps. The default value is 180 seconds.
ENABLE_SOLR_FACET_CACHE=false	To enable the facet cache for wildcard searches, set this parameter to true. Use this setting if you run wildcard searches on data that is older than 1 day. When it is enabled, the facets are counted before they are indexed by Log Analysis.
MAX_NON_INCREMENTAL_WINDOWS	Determines the number of asynchronous queries that can be run simultaneously for a percentile facet request query. The default value is 2.
COLLECTION_ASYNC_WINDOW	Determines the size of the asynchronous collection window. The default value is 1 day, 1d. Do not change this value unless you are certain that your installation can ensure that the performance of Log Analysis is not adversely affected.

Default ports

IBM Operations Analytics Log Analysis uses a number of default ports during the installation process.

Default ports

Table 4. The default ports for IBM Operations Analytics Log Analysis are as follows:		
Ports	Default Value	Description
Application WebConsole Port	9988	Use this port for unsecured http communication with the web application of IBM Operations Analytics Log Analysis.
Application WebConsole Secure Port	9987	Use this port for secured http communication with the web application of IBM Operations Analytics Log Analysis.
Database Server Port	1627	This port is used by IBM Operations Analytics Log Analysis for its internal database.
Data Collection Server Port	5529	This port is used by IBM Operations AnalyticsLog Analysis to collect data.
ZooKeeper Port Number	12181	This port is used by ZooKeeper service of IBM Operations AnalyticsLog Analysis to manage its Apache Solr nodes
Apache Solr Search Port	8983	This port is used by the Apache Solr server to listen for search queries.
Apache Solr Stop Port	7205	This port is used by IBM Operations AnalyticsLog Analysis to stop the Apache Solr server.

Note: The default ports listed in Table 1 apply to local IBM Operations AnalyticsLog Analysis installations. Remote IBM Operations AnalyticsLog Analysis installations may require different port numbers.

install.sh command

Use the `install.sh` command to install IBM Operations Analytics - Log Analysis.

The `install.sh` command is in the `<HOME>/IBM/LogAnalysis/remote_install_tool/` directory on the local installation of IBM Operations Analytics - Log Analysis.

install.sh command parameters

To install IBM Operations Analytics - Log Analysis with IBM Installation Manager, run the command:

```
./install.sh
```

This command installs IBM Operations Analytics - Log Analysis and installs or upgrades, IBM Installation Manager if no other version is installed. For more information, see [“Installing with the IBM Installation Manager UI”](#) on page 15.

To install IBM Operations Analytics - Log Analysis with the console, run the command:

```
./install.sh -c
```

This command installs IBM Operations Analytics - Log Analysis and installs or upgrades IBM Installation Manager, if no other version of IBM Installation Manager is installed. For more information, see [“Installing with the IBM Installation Manager command-line interface”](#) on page 17.

To silently install IBM Operations Analytics - Log Analysis, run the command:

```
./install.sh -s <HOME_DIR>/smcl_silent_install.xml
```

where *<HOME_DIR>* is your home directory. This command silently installs IBM Operations Analytics - Log Analysis and installs or upgrades IBM Installation Manager Version 1.8.2. For more information, see [“Silently installing Log Analysis”](#) on page 18.

To install the Tivoli Event Integration Facility Receiver or the IBM Tivoli Monitoring Log File Agent on a remote server, run the `remote_deploy.sh` script. For more information, see [“Installing Apache Solr on remote machines”](#) on page 64.

backup_restore.sh script

To back up or restore data, use the `backup_restore.sh` script.

Syntax

```
backup_restore.sh <backup_directory> -backup| -restore
```

where *<backup_directory>* is the path for the directory that you create to store the backed up files.

Parameters

<backup_directory>

The path for directory that you create to store the backed up files

backup

Backs up the current data to the backup directory.

restore

Restores the data stored in the backup directory.

Chapter 3. Upgrading, backing up, and migrating data

You use the same script to back up, restore, and migrate your data during an upgrade.

Before you begin

- Read about the limitations that apply to this procedure. For more information, see [“Backup and migration limitations”](#) on page 31
- If you do not configure a key-based Secure Shell (SSH) authentication as part of the installation, you are prompted for the password during the restoration. For more information about setting up SSH, see *Setting up Secure Shell to use key-based authentication..*
- **Fix Pack 1** Before you migrate to Log Analysis 1.3.3 FP001, complete the following steps for each saved search with an absolute time filter.
 1. To display a list of saved searches, click the **Saved Searches** icon.
 2. Select the saved search that you want to update.
 3. Right-click the saved search and select **Edit**.
 4. Edit the time filter to the Coordinated Universal Time.
 5. To save the updated search filter, click **Search**

About this task

Note: **Fix Pack 1** To migrate from Log Analysis 1.3.2 to Log Analysis 1.3.3 FP001, you must first migrate to Log Analysis 1.3.3. After you migrate to Log Analysis 1.3.3, you can install Log Analysis 1.3.3 FP001.

These items are backed up and migrated by this procedure:

- Saved searches, tags, and Data Sources
- Data Types including Source Types, Rule Sets, File Sets, and Collections.
- Topology configuration files
- Usage statistics
- LDAP configuration files
- Custom Search Dashboards
- Insight Packs
- Log File Agent (LFA) configuration files
- License configuration files
- All chart specifications, including custom chart specifications

In addition to these files, a number of files that are not required for a new installation are also backed up and maintained for reference purposes. These files include:

- Log files that are generated by IBM Operations Analytics - Log Analysis
- Log files that were uploaded in batch mode

Data other than the types of listed previously are not backed up or restored. Any customization that is made outside the files that are listed here must be migrated manually. For example, user information and changes to passwords for default users are not migrated to the target server.

LDAP information for one LDAP server is migrated automatically. If you have more than one DAP server, you must migrate and configure the information from the other LDAP server manually. The migrated information is stored in the `ldapRegistry.xml` file.

Procedure

To back up and restore data in IBM Operations Analytics - Log Analysis 1.3.3:

- a. Back up your data. For more information, see *Backing up data*.
- b. Restore your data. For more information, see *Restoring data*.

To migrate data from IBM Operations Analytics - Log Analysis 1.3.2 to 1.3.3:

- a. Back up your data in 1.3.2.
- b. Move the backed up, compressed files to the <Backup_dir> on the 1.3.3 server.
- c. Restore the data on the 1.3.3 server.

What to do next

You must update any existing Logstash plugins for IBM Operations Analytics - Log Analysis 1.3.2 after you migrate to IBM Operations Analytics - Log Analysis 1.3.3 to ensure that Logstash installations function correctly. For more information about updating the Logstash, see *Updating the Logstash plugin after migration* in the *Upgrading, backing up, and migrating data* section.

If you manually configured Hadoop in IBM Operations Analytics - Log Analysis 1.3.2, you must run the following command after you migrate to IBM Operations Analytics - Log Analysis 1.3.3.

```
<HOME>/IBM/LogAnalysis/utilities/migration/hadoop_config_migration.sh
```

Submit the Name Node and Data Node details to ensure that configuration and status details are updated and visible in the **Hadoop Integration** screen.

Note: Hadoop configuration migration from 1.3.2 to 1.3.3 is only supported for **IBM Open Platform Hadoop** installations. The Hadoop configuration migration is required to view details in the **Hadoop Integration** screen. Other Hadoop configurations continue to function without this feature.

Note: You must use the same Log Analysis user for both backup and restore. This is a Hadoop configuration requirement.

If you migrate data to a new target server, you must complete the following steps to update the data source with the relevant information for the server:

1. Log in to IBM Operations Analytics - Log Analysis and open the **Data sources** workspace.
2. Identify the data sources that use **Local file** as the location in their configuration. For each of these data sources, open the data source and complete the steps in the wizard without changing any data. This action updates the data sources with the relevant information for the new server.
3. Identify the data sources that use **Custom** as the location in their configuration. For each of these data sources, open the data source and complete the steps in the wizard, updating the host name to match the new one.

Related tasks

[“Setting up Secure Shell to use key-based authentication” on page 53](#)

Secure Shell (SSH) is a cryptographic network protocol for secure data communication between different computers. You set up key-based authentication between the IBM Operations Analytics - Log Analysis servers and the remote computers to which it connects.

[“Backing up data” on page 31](#)

To back up data, complete the procedure.

[“Restoring data” on page 32](#)

To restore backed up data, complete the following procedure.

[“Updating remote Logstash installations” on page 33](#)

After you migrate to IBM Operations Analytics - Log Analysis 1.3.3 , you must update any remote installations of Logstash that you created with IBM Operations Analytics - Log Analysis 1.3.2.

Backup and migration limitations

Before you back up, restore or migrate data between the same or different versions, read the limitations.

The following restrictions apply:

- To restore backed up data from Indexing Engines, you must set up at least the same number of Indexing Engines in the target system as existed previously in the source system.
- During data restoration, backed up data is not merged with existing data on the server. Backed up data must be restored on a target system immediately after IBM Operations Analytics - Log Analysis is installed.
- Restoring backed up data must not be completed more than once on a target server. If errors occur restoring backed up data, you must attempt the restoration after you remove and install IBM Operations Analytics - Log Analysis.
- Only data that is contained in the IBM Operations Analytics - Log Analysis default directories is backed up and restored. Any customization and modifications that are completed outside these directories are not backed up or restored.
- If you create a Custom Search Dashboard that points to the Derby database and, which has an encrypted user ID and password, you must update the application to reflect changes to the encryption of the Derby database user ID and password when you migrate to a new version. The Custom Search Dashboard cannot run until the application is edited to reflect the encryption that is used by the updated installer.
- When you restore a system that was extended with extra Insight Packs (for example, Insight Packs created with DSV toolkit), the log files and data source directories are not restored. To resolve this issue, you must manually add these directories, using the appropriate LFA configuration file as a reference.
- To migrate IBM Operations Analytics - Log Analysis 1.3.0, or earlier versions, to IBM Operations Analytics - Log Analysis 1.3.3, you must first migrate to IBM Operations Analytics - Log Analysis 1.3.2.
- Any customizations that you make to the server configuration in the `server.xml` file are not restored after you migrate the data. However, the data is stored in the `server.xml` file that is stored in the `backup.zip` that you create when you back up your data. You can use this file to restore the customizations.
- The `ltpekeys` files are not copied during the backup and restore. You must back up and restore these files manually.
- Only the default certificates for the Liberty server are migrated. If you use custom certificates, you must back up and restore these files manually.

Backing up data

To back up data, complete the procedure.

Procedure

1. To stop IBM Operations Analytics - Log Analysis, use the `unity.sh` script that is in the `<HOME>/IBM/LogAnalysis/utilities` directory:

```
./unity.sh -stop
```

2. Create a backup directory where you will store the backup files. Do not create this directory in the same directory in which IBM Operations Analytics - Log Analysis is installed. This directory is called the backup directory or `<backup_dir>` in this procedure.

3. From the <HOME>/IBM/LogAnalysis/utilities/migration directory, run the following command:

```
./backup_restore.sh <backup_dir> backup
```

where *<backup_dir>* is the directory that you created in step 2. The backup directory that you specify must be empty.

The backup command creates a set of archive files in the backup directory.

Results

When you run the backup command, IBM Operations Analytics - Log Analysis creates a zipped file that contains the archived data.

The zipped files have the word *restore* in the file name and are numbered sequentially. For example, *LogAnalytics_30Jul2014_Restore_001.zip*. These are the files that you can use later to restore the data.

The command also generates reference files. These files are stored in a separate archive file. The file name contains the words *BackupOnly*, for example *LogAnalytics_30Jul2014_BackupOnly_001.zip*. You do not need to restore these.

Restoring data

To restore backed up data, complete the following procedure.

Procedure

1. To stop IBM Operations Analytics - Log Analysis, use the *unity.sh* script that is in the <HOME>/IBM/LogAnalysis/utilities directory:

```
unity.sh -stop
```

2. From the <HOME>/IBM/LogAnalysis/utilities/migration directory, run the command:

```
./backup_restore.sh <Backup_dir> restore
```

where *<Backup_dir>* is the path to the directory containing your backed up files.

If you migrate Indexing Engine data, you must create the same number of Indexing Engines in the target system. If you specified a password when you created the nodes, you are prompted for the password.

3. If LDAP is configured on your source IBM Operations Analytics - Log Analysis server, you are prompted for the LDAP bind password during the restoration of your data on the target server.
4. **Fix Pack 1** If an LDAP registry file is in the Log Analysis 1.3.3 FP001 backup, select one of the following options.

Use existing *ldapRegistry.xml* from backup

Use the existing *ldapRegistry.xml* from the Log Analysis backup.

Generate new *ldapRegistry.xml* using *ldapRegistryHelper.properties*

Create a new *ldapRegistry.xml*. For more information about creating the *ldapRegistry.xml*, see [“ldapRegistryHelper.properties” on page 37](#)

5. If you have configured IBM Operations Analytics - Log Analysis to stream data from the same server on which it has been installed, and have migrated IBM Operations Analytics - Log Analysis to a new target server, review and update the host name setting in your data sources to reflect the host name of the target server.
6. To start IBM Operations Analytics - Log Analysis, use the *unity.sh* script that is in the <HOME>/IBM/LogAnalysis/utilities directory:


```
unity.sh -start
```

What to do next

Verify that the data migration and restoration has been successful. Confirm that all artifacts and data have been restored before you delete the back up archive. Progress information and errors recorded during back up and restore are stored in the <HOME>/IBM/LogAnalysis/utilities/migration/logs directory.

Updating remote Logstash installations

After you migrate to IBM Operations Analytics - Log Analysis 1.3.3 , you must update any remote installations of Logstash that you created with IBM Operations Analytics - Log Analysis 1.3.2.

Procedure

To import the `client.crt` file from <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/resources/security on the IBM Operations Analytics - Log Analysis 1.3.3. server to <JAVA_HOME> on the remote machine, use the following command.

```
<JAVA_HOME>/jre/bin/keytool -import -file <path> -keystore <JAVA_HOME>/jre/  
lib/security/cacerts -storepass changeit
```

Where

- <JAVA_HOME> is the directory where Java is installed on the remote machine, <logstash_install_dir>/LogAnalysis/ibm-java/
- <path> is the `client.crt` location.

Chapter 4. Configuring

You can configure IBM Operations Analytics - Log Analysis through the user interface and command-line interfaces. You can also administer and manage application security and single sign-on. This section outlines how to configure IBM Operations Analytics - Log Analysis.

Postinstallation configuration

After you install IBM Operations Analytics - Log Analysis, you must complete the required postinstallation tasks.

The postinstallation configuration includes the following tasks:

- Installing the sample data. This is optional.
- Creating users and assigning roles

After you complete the postinstallation configuration, you can scale your installation. For more information, see [“Streaming data from multiple remote sources across a network”](#) on page 213.

If you want to install Indexing Engines on remote servers, you can create them after IBM Operations Analytics - Log Analysis is installed. For more information, see [“Installing Apache Solr on remote machines”](#) on page 64.

If you do not install a local instance of an Indexing Engine, you must install an Indexing Engine on one of the remotely connected servers. If you do not install at least one Indexing Engine instance either locally or on a remote server, the search and indexing features do not work.

Configuring secure communication and authentication

After you install Log Analysis, you can configure secure communications and user authentication.

You can configure Lightweight Directory Access Protocol (LDAP) for user authentication. You can also configure secure sockets layer (SSL) and Secure Shell (SSH) protocols for secure communication, and configure single sign-on (SSO) with Tivoli Integrated Portal or Jazz® for Service Management.

Security considerations

A number of files in the IBM Operations Analytics - Log Analysis environment contain encoded or encrypted passwords. Access to these files must be controlled either through controlling access to the file system or through access rights or file permissions. This can be achieved by limiting the groups and users that have access to these files. The files that contain this information are:

- <HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/unity.conf
- <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/ldapRegistry.xml
- <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/unityUserRegistry.xml
- <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/keystore/unity.ks
- <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/keystore/unity_statistics.ks
- <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties
- <HOME>/IBM/LogAnalysis/eif_remote_install_tool/config/rest-api.properties
- <HOME>/IBM/LogAnalysis/solr_install_tool/scripts/register_solr_instance
- <HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh

LDAP configuration

You can implement an LDAP user registry in place of the database-managed custom user registry that is provided in Log Analysis.

Log Analysis uses database-managed custom user registry as the default setting for user authentication. After you configure LDAP authentication, you cannot revert to database-managed custom user registry.

Configure LDAP immediately after Log Analysis is installed.

The Log Analysis user registry is case sensitive. However, the LDAP user registry is not.

For a worked example of how to configure LDAP and SSL authentication, see [“LDAP and SSL configuration example” on page 46](#).

Log Analysis supports Tivoli Directory Server (TDS) and Microsoft Active Directory (AD) LDAP servers.

To configure LDAP authentication, complete the following steps:

1. Configure Log Analysis. Choose from one of the following configurations:
 - Use the `ldapRegistryHelper.sh` utility to help you to configure and enable LDAP authentication. For more information, see [“Configuring LDAP authentication with the ldapRegistryhelper.sh script” on page 36](#).
 - Alternatively, you can create the required `ldapRegistry.xml` file and manually enable LDAP authentication. For more information, see [“Manually configuring LDAP authentication” on page 39](#).
 - If you want to use multiple LDAP servers, you must create the `ldapRegistry.xml` file and configure LDAP manually. You cannot use the `ldapRegistryHelper.sh` utility. For more information, see [“Configuring multiple LDAP servers” on page 41](#).
2. Map any custom groups in LDAP to the security roles in Log Analysis and add any custom groups to the OAuth security role. If you use the default group names, `UnityAdmins` and `UnityUsers`, you can skip this step.

The default group names are mapped to security roles in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/unityConfig.xml` file in Log Analysis. If you do not use these default groups, for example if you cannot create these groups in your LDAP repository, you can add custom groups in Log Analysis.

For more information, see [“Mapping LDAP groups to the security role” on page 41](#)
3. Map your LDAP users in Log Analysis. For more information, see [“Mapping LDAP users in Log Analysis” on page 42](#).
4. Update the passwords in the configuration files. For more information, see [“Updating passwords in the configuration files” on page 44](#).

If your LDAP environment is configured to interact with other security software, Log Analysis might have to interact with this software to complete authentication. You must consider any user ID or password restrictions that this other security software imposes. For example, if one of the other LDAP applications requires an eight character password, the password that you specify in Log Analysis must meet that requirement.

Configuring LDAP authentication with the `ldapRegistryhelper.sh` script

You can use the `ldapRegistryHelper.sh` command to help you to create and enable an LDAP authentication with IBM Tivoli Directory Server or Microsoft Active Directory server.

About this task

You cannot directly configure multiple LDAP servers with this utility. However, you can use it to help you to create the required `ldapRegistry.xml` files. For more information, see [“Configuring multiple LDAP servers” on page 41](#).

Procedure

1. To stop the Log Analysis server, use the following command:

```
./ unity.sh -stop
```

2. To specify the LDAP server details, edit the `ldapRegistryHelper.properties` file that is in the `<HOME>/IBM/LogAnalysis/utilities/` directory. For more information about the `ldapRegistryHelper` properties, see the [“ldapRegistryHelper.properties” on page 37](#) topic in the *Configuration* guide.

3. Navigate to the `<HOME>/IBM/LogAnalysis/utilities` directory and run the following command:

```
./ldapRegistryHelper.sh config
```

4. Run the following command:

```
./ldapRegistryHelper.sh enable
```

5. If you cannot use the default groups, for example if you cannot create these groups in your LDAP repository, you can map the custom groups in the LDAP registry to security roles in Log Analysis. For more information, see [“Mapping LDAP groups to the security role” on page 41](#).
6. Create the required users and roles. For more information, see [“Mapping LDAP users in Log Analysis” on page 42](#)
7. To start the Log Analysis server, use the following command:

```
./ unity.sh -start
```

Results

Basic LDAP authentication between Log Analysis and the IBM Tivoli Directory Server or the Microsoft Active Directory server is enabled.

What to do next

After you configure LDAP, you must update the password in the configuration files. For more information, see [“Updating passwords in the configuration files” on page 44](#).

ldapRegistryHelper.properties

You can edit the `ldapRegistryHelper.properties` to specify LDAP server details.

The following properties are required and define the connection information for the target LDAP server.

Table 5. LDAP server connection information properties	
Property	Description
ldap_hostname_property=	The LDAP hostname.
ldap_port_property=	The LDAP port.
ldap_baseDN_property=	The LDAP baseDN. For example, "dc=com" for TDS users, and "CN=Users,DC=sflab,DC=local" for AD users.

The following properties are optional and define the connection information for the target LDAP server. Where applicable, default settings are assigned.

The **bindPassword** value for AD users is encrypted in the `ldapRegistryHelper_config.xml`.

Table 6. Optional LDAP server connection information properties	
Property	Description
ldap_bindDN_property=	The LDAP bindDN. For example, "CN=Administrator,CN=Users,DC=sflab,DC=local" for AD users.

Table 6. Optional LDAP server connection information properties (continued)	
Property	Description
ldap_bindPassword_property=	The LDAP bind password.
ldap_realm_property=	The LDAP realm. The default value is LdapRegistryRealm.
ldap_id_property=	The LDAP ID. The default value is LdapRegistryId.
ldap_ignoreCase_property=	The LDAP ignore case. The default value is true.
Fix Pack 1 recursiveSearch=	Fix Pack 1 Enable recursive search. The default value is true. This is only available in IBM Operations Analytics - Log Analysis 1.3.3 Fix Pack 1.

ldapRegistryHelper.sh command

You can use the `ldapRegistryHelper.sh` command to enable a basic connection for user authentication in IBM Operations Analytics - Log Analysis.

For more information about how to use the command to set up LDAP authentication with IBM Tivoli Directory Server or Microsoft Active Directory, see [“Configuring LDAP authentication with the ldapRegistryHelper.sh script” on page 36.](#)

Supported integrations

This command currently supports connections to the IBM Tivoli Directory Server and Microsoft Active Directory.

Prerequisites

Before you use this command, you must update the `ldapRegistryHelper.properties` file in the `<HOME>/IBM/LogAnalysis/utilities/` directory with the connection and configuration information for the target LDAP server.

Syntax

The `ldapRegistryHelper.sh` command is in the `<HOME>/IBM/LogAnalysis/utilities` directory and it has the following syntax:

```
ldapRegistryHelper.sh    config | enable
```



Warning:

To run the script, the `JAVA_HOME` variable must be set correctly for IBM Operations Analytics - Log Analysis. If the script fails, run the following command to set the `JAVA_HOME` variable:

```
JAVA_HOME=${<HOME>}/IBM-java
```

Parameters

The `ldapRegistryHelper.sh` command has the following parameters:

config

Use the `config` parameter to create an XML file that is called `ldapRegistry.xml` in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity` directory. This file uses the connection and configuration information that is defined in the `ldapRegistryHelper.properties` file.

enable

Use the enable parameter to enable LDAP authentication that uses the information that is specified in the ldapRegistry.xml file. This parameter also disables the reference to the database-managed custom user registry.

Manually configuring LDAP authentication

If you want to manually configure LDAP authentication, you can manually configure the settings in your own XML file or you can modify the ldapRegistry.xml that is output by the ldapRegistryHelper.sh command to meet your requirements.

About this task

The following procedure describes some of the steps that are automated by the ldapRegistryHelper.sh command. Read this procedure to help you understand the necessary steps for configuring LDAP authentication. For more information, see [Configuring an LDAP user registry with the Liberty profile](#).

Procedure

1. To stop the Log Analysis server, use the following command:

```
./ unity.sh -stop
```

2. Manually create an LDAP configuration file that is named ldapRegistry.xml and save it in the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity directory or modify the ldapRegistry.xml that is output by the ldapRegistryHelper.sh command.
3. Update the ldapRegistry.xml with the appropriate configuration information:

- For IBM Tivoli Directory Server, add the text:

```
<ldapRegistry id="IBMDirectoryServerLDAP" realm="SampleLdapIDSRealm"
  host="host.domain.com" port="389" ignoreCase="true"
  baseDN="o=domain,c=us"
  bindDN="cn=root"
  bindPassword="password"
  ldapType="IBM Tivoli Directory Server">
  <idsFilters
    userFilter="(&(uid=%v)(objectclass=ePerson))"
    groupFilter="(&(cn=%v)(|(objectclass=groupOfNames)
(objectclass=groupOfUniqueNames)(objectclass=groupOfURLs)))"
    userIdMap="*:uid"
    groupIdMap="*:cn"
    groupMemberIdMap="ibm-allGroups:member;ibm-allGroups:
uniqueMember;groupOfNames:member;groupOfUniqueNames:uniqueMember" />
  </ldapRegistry>
```

- For Microsoft Active Directory, add the text:

```
<ldapRegistry id="ActiveDirectoryLDAP" realm="SampleLdapADRealm"
  host="host.domain.com" port="389" ignoreCase="true"
  baseDN="cn=users,dc=domain,dc=com"
  bindDN="cn=myuser,cn=users,dc=domain,dc=com"
  bindPassword="password"
  ldapType="Microsoft Active Directory" />
```

4. Update these attributes to reflect your LDAP server configuration:

- ID
- realm
- host
- port
- baseDN
- bindDN

5. AD users must run the securityUtility command that is in the <HOME>/IBM/LogAnalysis/bin directory to encode the bindPassword password. This step is optional for TDS users as they do not require the bindPassword password.

After you run the command, copy the encrypted value that is output by the command to the bindPassword property.

For example, you can run the following command to encrypt the password:

```
./securityUtility encode myPassw0rd
```

The encrypted password is displayed as the {xor} value. You need to add this value to your configuration file.

```
{xor}MiYPPiwsKG8t0w==
```

For more information about this command, see [“securityUtility.sh utility”](#) on page 109.

6. (Optional) If your implementation uses a Microsoft Active Directory LDAP that uses different object classes to define users and groups, update the userFilter and groupFilter attributes as required.
7. (Optional) If your implementation uses Microsoft Active Directory, update the user, group and group member mapping attributes as required for your LDAP environment.
8. Open the server.xml file in the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity directory and add these lines:
 - a. Only one type of user registry can be configured for authentication, therefore you must disable the database-managed custom user registry to enable LDAP. Comment out the following lines in the server.xml file that reference the database-managed custom user registry:

```
<!-- Include the basic registry predefined with default users
and groups -->
<!-- <include optional="true" location="${server.config.dir}/
unityUserRegistry.xml"/>
-->
```

If you do not remove this reference, an error message is displayed.

- b. Add an include tag to replace the reference to the custom user registry with a reference to the ldapRegistry.xml file. For example:

```
<!-- Include the LDAP registry for user and groups -->
<include optional="true" location="${server.config.dir}/
ldapRegistry.xml"/>
```

9. If you cannot use the default groups, for example if you cannot create these groups in your LDAP repository, you can map the custom groups in the LDAP registry to security roles in Log Analysis. For more information, see [“Mapping LDAP groups to the security role”](#) on page 41.
10. Create the required users and roles. For more information, see [“Mapping LDAP users in Log Analysis”](#) on page 42
11. To start the Log Analysis server, use the following command:

```
./ unity.sh -start
```

What to do next

After you configure LDAP user registry, you must update the unityadmin password in the IBM Operations Analytics - Log Analysis configuration files. For more information, see [“Updating passwords in the configuration files”](#) on page 44.

Configuring multiple LDAP servers

If you use multiple servers in your LDAP landscape, you can configure the servers to integrate with Log Analysis.

Before you begin

The LDAP user names must be unique across all servers.

Procedure

1. To stop the Log Analysis server, use the following command:

```
./ unity.sh -stop
```

2. To specify the LDAP server details, complete the following steps.

You can use the `ldapRegistryHelper` utility to help you to create the `ldapRegistry.xml` file or you can create the `ldapRegistry.xml` file manually.

- a) To use the `ldapRegistryHelper` utility to help you to create the `ldapRegistry.xml` file:

- 1) Edit the `ldapRegistryHelper.properties` file that is in the `<HOME>/IBM/LogAnalysis/utilities/` directory. For more information about the `ldapRegistryHelper` properties, see [“ldapRegistryHelper.properties” on page 37](#).
- 2) To ensure that the servers are distinct, edit the **realm** and **id** properties in the `ldapRegistryHelper.properties` file that is in the `<HOME>/IBM/LogAnalysis/utilities/` directory. For more information about the `ldapRegistryHelper` properties, see [“ldapRegistryHelper.properties” on page 37](#).
- 3) Navigate to the `<HOME>/IBM/LogAnalysis/utilities` directory. Use the `ldapRegistryHelper.sh` command to generate the `ldapRegistry.xml` file and run the following command to generate the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/ldapRegistry.xml` file:

```
./ldapRegistryHelper.sh config
```

- b) Alternatively, you can create the `ldapRegistry.xml` file manually. To ensure that the servers are distinct, edit the **realm** and **id** properties in the file.
3. Back up the `ldapRegistry.xml` file.
 4. Repeat the steps 2 and 3 for each LDAP server.
 5. Merge the `ldapRegistry.xml` files into a single file. Ensure that you only use one `<server><server>` element in the merged file.
 6. If you cannot use the default groups, for example if you cannot create these groups in your LDAP repository, you can map the custom groups in the LDAP registry to security roles in Log Analysis. For more information, see [“Mapping LDAP groups to the security role” on page 41](#).
 7. Create the required users and roles. For more information, see [“Mapping LDAP users in Log Analysis” on page 42](#).
 8. To start the Log Analysis server, use the following command:

```
./ unity.sh -start
```

Mapping LDAP groups to the security role

To enable LDAP authentication for custom LDAP groups, you must map the custom groups to the required security roles in Log Analysis.

About this task

Access to Log Analysis is controlled by the two security roles in Log Analysis:

UnityUser

This role grants access to the Search UI.

UnityAdmin

This role grants access to the Search and Administrators UI.

These roles are mapped to the UnityUsers and UnityAdmins groups in Log Analysis by default.

If you cannot use the UnityUsers and UnityAdmins groups, for example if you cannot create these in your LDAP application, or if you use custom groups in your LDAP directory, you must add your LDAP groups to the security roles.

You also need to add these custom groups to the OAuth security role to enable access to Log Analysis.

Procedure

1. Open the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/UnityConfig.xml file.
2. If your landscape requires it, you can map other groups in the LDAP registry to security roles in Log Analysis. To map groups to the security roles, replace <Group_name> with the group names. For example:

```
<security-role name="UnityUser">
  <group name="UnityUsers"/>
  <group name="UnityAdmins" />
  <group name="<Group_name1>"/>
  <group name="<Group_name2>"/>
</security-role>
<security-role name="UnityAdmin">
  <group name="<Group_name1>" />
  <group name="TestLAAdmin"/>
</security-role>
```

where <Group_name> are the names of any custom groups that you use.

3. To enable the oauth security role that allows users to access Log Analysis, specify the custom group names.

For example:

```
<oauth-roles>
  <authenticated>
    <group name="UnityUsers"/>
    <group name="UnityAdmins"/>
    <group name="<Group_name1>"/>
    <group name="<Group_name2>"/>
  </authenticated>
</oauth-roles>
```

where <Group_name> are the names of any custom groups that you use. The UnityUsers role is added by default.

4. If you are not using the UnityUsers and UnityAdmins default groups in your implementation, you can remove these groups.
5. Save your changes.
6. Start the IBM Operations Analytics - Log Analysis server by using the following command:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -start
```

Mapping LDAP users in Log Analysis

To enable LDAP authentication, create the appropriate users, groups, and roles in Log Analysis and your LDAP directory.

About this task

For more information about how to create users and roles in Log Analysis, see [“Users and roles” on page 58](#).

To map your LDAP users in Log Analysis, complete the following steps:

Procedure

1. Create groups called UnityAdmin and UnityUsers in your LDAP repository. If you cannot create these groups, map the existing LDAP groups in Log Analysis. For more information, see [“Mapping LDAP groups to the security role”](#) on page 41.
2. Create a user called unityadmin in your LDAP directory.
The unityadmin user can access Log Analysis.
3. If you want to use any users other than unityadmin to access Log Analysis, complete the following steps:
 - a. Use the unityadmin user to log in to Log Analysis.
 - b. Create Log Analysis users with the same name as the users that are defined in your LDAP repository.
 - c. Create roles in Log Analysis and assign these roles to the users that you created in the previous step.

Results

After you complete this task, you can update the passwords in the configuration files. For more information, see [“Updating passwords in the configuration files”](#) on page 44.

Enabling case sensitivity for LDAP

The LDAP user registry does not use case-sensitive user names by default. You can enable case-sensitivity for LDAP.

Before you begin

Before you complete the procedure, you must apply APAR PI53797 for IBM WebSphere Application Server Liberty Profile.

For more information, see technote [2404194](#)

Ensure that the LDAP you use supports case-sensitive user names and is enabled.

Procedure

To make LDAP user names case sensitive, complete the following steps.

1. Stop Log Analysis.
2. Add the following property to the ldapRegistry.xml file:

```
ignoreCase="false"
```

For example,

```
<server>
  <ldapRegistry
    host="9.118.40.171"
    port="389"
    baseDN="dc=com"
    realm="LdapRegistryRealm"
    id="LdapRegistryId"
    ignoreCase="false"
    ldapType="IBM Tivoli Directory Server">
    <idsFilters
      userFilter="(&(uid=%v)(|(objectclass=ePerson)
(objectclass=inetOrgPerson)))"
      groupFilter="(&(cn=%v)(|(objectclass=groupOfNames)
(objectclass=groupOfUniqueNames)(objectclass=groupOfURLs)))"
      userIdMap="*:uid"
      groupIdMap="*:cn"
      groupMemberIdMap="ibm-allGroups:member;ibm-allGroups:uniqueMember;
groupOfNames:member;groupOfUniqueNames:uniqueMember"/>
    </ldapRegistry>
  </server>
```

3. Save the `ldapRegistry.xml` file.
4. Ensure that the **LDAP_IGNORE_PROPERTY** parameter in the `unitysetup.properties` files is set to **false**.
For example,

```
LDAP_IGNORECASE_PROPERTY=false
```

For more information about Configuring properties files, see *Configuration properties file* in the *Reference* guide.

5. Start Log Analysis.

Related reference

[“Configuration properties file” on page 25](#)

After you complete the installation, to modify the configuration properties edit the `unitysetup.properties` file.

Updating passwords in the configuration files

In most cases, after you create or change a user or password in your Lightweight Directory Access Protocol (LDAP) application, you do not need to update the passwords in IBM Operations Analytics - Log Analysis. However, if the new or changed password is specified in the IBM Operations Analytics - Log Analysis configuration files, you must update the files with the new or changed information.

Procedure

1. To stop the IBM Operations Analytics - Log Analysis server, use the `unity.sh` script that is in the `<HOME>/IBM/LogAnalysis/utilities`:

```
./unity.sh -stop
```

2. If you do add or update a password for an LDAP user, you can encrypt the password. You must add the encrypted password in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/ldapRegistry.xml` file. To encrypt the password, run the `securityUtility.sh` script that is in the `<HOME>/wlp/bin/` directory. For more information, see [“securityUtility.sh utility” on page 109](#).

For example, you run the following command to encrypt the password:

```
./securityUtility encode myPassword
```

The encrypted password is displayed as the `{xor}` value. You need to add this value to your configuration file.

```
{xor}MiYPPiwsKG8t0w==
```

3. If you change the password that is used by the `unityadmin`, you must update the encrypted password in the following files to match the updated password. To generate the encrypted password use the `unity_securityUtility.sh` utility in the `<HOME>/IBM/LogAnalysis/utilities` directory. For example:

```
unity_securityUtility.sh encode password
```

- `<HOME>/IBM/LogAnalysis/utilities/datacollector-client/javaDatacollector.properties`. For example:

```
#The password to use to access the unity rest service  
password={aes}EF712133E0677FEBB30624BA5EE62BC2
```

- `<HOME>/IBM/LogAnalysis/remote_install_tool/config/rest-api.properties`. For example:

```
ibm.scala.rest.password={aes}EF712133E0677FEBB30624BA5EE62BC2
```

- <HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/unity.conf. For example:

```
unity.data.collector.password={aes}EF712133E0677FE9B30624BA5EE62BC2
```

- <HOME>/IBM/LogAnalysis/solr_install_tool/scripts/register_solr_instance.sh. For example:

```
PASSWD={aes}EF712133E0677FE9B30624BA5EE62BC2
```

4. If you change the password that is used by the unityadmin, you must update the password parameter in the <HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh script.

```
password={aes}7A0B2401A8E29F37CD768CB78E205CAD
```

5. To start the IBM Operations Analytics - Log Analysis server, use the unity.sh script that is in the <HOME>/IBM/LogAnalysis/utilities directory:

```
./ unity.sh -start
```

Fix Pack 1 **Changing the default administrative user**

Fix Pack 1 unityadmin is the default administrative user for Log Analysis. If you cannot create a user called unityadmin in your LDAP repository, you can change the default user in Log Analysis to match the user specified in your LDAP directory.

Before you begin

You must use IBM Operations Analytics - Log Analysis 1.3.3 Fix Pack 1 or later.

About this task

You should only change this user if you cannot create the unityadmin in your LDAP repository.

Procedure

1. Go to the <HOME>/IBM/LogAnalysis/utilities/authorization/ directory.
2. Edit the admin.properties file. Add values for the following parameters, adding the name of the new user in the LDAP_USER_NAME= parameter:

```
LDAP_USER_NAME=  
DISPLAY_NAME=  
DESCRIPTION=
```

For example:

```
LDAP_USER_NAME=ldap_admin  
DISPLAY_NAME=LA Administrator  
DESCRIPTION=Admin for LA and LDAP
```

3. Save the file.
4. To update the name, run the script. Enter the following command:

```
./makeLAadmin.sh admin.properties
```

What to do next

The new user that you specify must be a member of the LDAP group. For more information, see [“Mapping LDAP users in Log Analysis”](#) on page 42.

The UnityAdmin security role needs to be assigned to the LDAP group. For more information, see [“Mapping LDAP groups to the security role”](#) on page 41.

You can now use the new user to log in to Log Analysis.

LDAP and SSL configuration example

Use this end to end example to help you to configure Lightweight Directory Access Protocol (LDAP) and secure socket layer (SSL) for Log Analysis.

The steps here are for exemplary purposes to help you to understand how to set up LDAP authentication with SSL. The exact steps required to set up your own implementation will differ.

Prerequisites

Ensure that the Java home variable is set to `JAVA_HOME= /opt/IBM/SCALA/LogAnalysis/ibm-java/jre`. For example, to set this variable, enter the following command:

```
export JAVA_HOME= /opt/IBM/SCALA/LogAnalysis/ibm-java/jre
```

Configure the LDAP registry helper properties

Configure the properties of the LDAP registry helper script:

1. Edit the `<HOME>/IBM/LogAnalysis/utilities/ldapRegistryHelper.properties` file.
2. Ensure that you do not change the default LDAP type property:

```
ldap_type_property=IBM Tivoli Directory Server
```

3. Specify the mandatory connection information:

```
ldap_hostname_property=123.example.com
ldap_port_property=636
ldap_baseDN_property=o=example.com
```

4. Specify the optional connection properties for the target LDAP server. The following properties are optional. The `ldap_bindPassword_property` parameter is later encrypted by the `ldapRegistryHelper_config.xml` script and the encrypted version is written to the `ldapRegistry.xml` file. The password is automatically removed from the `ldapRegistryHelper.properties` file after the `ldapRegistryHelper_config.xml` script runs. The following example includes some default values:

```
ldap_bindDN_property=
ldap_bindPassword_property=
ldap_realm_property=LdapRegistryRealm
ldap_id_property=example
ldap_ignoreCase_property=true
```

5. Specify the default LDAP filters for each vendor. The filter properties that are used by the `ldapRegistryHelper_config.xml` script. These properties are determined by the LDAP type that is specified in the `ldap_type_property` parameter in step 1.

```
# IBM Tivoli Directory Server
ldap_TDS_userFilter_property=(&(emailAddress=%v)(objectclass=person))
ldap_TDS_groupFilter_property=(&(cn=%v)(|(objectclass=groupOfNames)
(objectclass=groupOfUniqueNames)(objectclass=groupOfURLs)))
ldap_TDS_userIdMap_property=*:emailAddress
ldap_TDS_groupIdMap_property=*:cn
ldap_TDS_groupMemberIdMap_property=ibm-allGroups:member;
ibm-allGroups:uniqueMember;groupOfNames:member;groupOfUniqueNames:
uniqueMember
```

Run the LDAP registry helper script

To run the LDAP registry helper script, enter the following command:

```
<HOME>/IBM/LogAnalysis/utilities/
ldapRegistryHelper.sh config
```

The script generates the `ldapRegistry.xml` based on the properties that are specified in the `<HOME>/IBM/LogAnalysis/utilities/ldapRegistryHelper.properties` file.

Edit the ldapRegistry.xml file

Next, you need to edit the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/ldapRegistry.xml.

1. Edit the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/ldapRegistry.xml
2. Add the following properties:

```
sslEnabled="true"
sslRef="LDAPSSLSettings">
```

For example:

```
<server>
<ldapRegistry
host="123.example.com"
port="636"
baseDN="o=example.com"
realm="LdapRegistryRealm"
id="example"
ignoreCase="true"
ldapType="IBM Tivoli Directory Server"
sslEnabled="true"
sslRef="LDAPSSLSettings">
<idsFilters
userFilter="( & (emailAddress=%v) (objectclass=person)) "
groupFilter="( & (cn=%v) (| (objectclass=groupOfNames)
(objectclass=groupOfUniqueNames) (objectclass=groupOfURLs))) "
userIdMap="*:emailAddress"
groupIdMap="*:cn"
groupMemberIdMap="ibm-allGroups:member;ibm-allGroups:uniqueMember;
groupOfNames:member;groupOfUniqueNames:uniqueMember"/>
</ldapRegistry>
</server>
```

Edit the unityConfig.xml file

Next, you need to add the Log Analysis user information to your LDAP configuration information in Log Analysis:

1. Edit the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/unityConfig.xml.
2. Specify the following attributes:

```
<server>
<application type="war" id="Unity" name="Unity"
location="${server.config.dir}/apps/Unity.war">
<application-bnd>
<security-role name="UnityUser">
<group name="UnityUsers" />
<group name="UnityAdmins" />
<group name="IGA_SCALA_ADMIN" />
<group name="IGA_SCALA_USER" />
</security-role>
<security-role name="UnityAdmin">
<group name="UnityAdmins" />
<group name="IGA_SCALA_ADMIN" />
</security-role>
</application-bnd>
</application>

<oauth-roles>
<authenticated>
<group name="UnityUsers" />
</authenticated>
</oauth-roles>
</server>
```

Create the JKS keystore

To create the JKS keystore file:

1. Go to the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity directory.

2. Run the keytool. For example:

```
[scala@c25x0012 bin]$ /opt/IBM/SCALA/LogAnalysis/ibm-java/jre/bin/keytool
-genkeypair -alias scala -keyalg RSA -keystore LdapSSLKeyStore.jks
-keysize 2048 -validity 7300
Enter keystore password:
Keystore password is too short - must be at least 6 characters
Enter keystore password:
Keystore password is too short - must be at least 6 characters
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: <SCALA_SERVER_FQDN>
What is the name of your organizational unit?
[Unknown]: IGA
What is the name of your organization?
[Unknown]: IGA
What is the name of your City or Locality?
[Unknown]: US
What is the name of your State or Province?
[Unknown]: US
What is the two-letter country code for this unit?
[Unknown]: US
Is CN=IGA, OU=IGA, O=IGA, L=US, ST=US, C=US correct? (type "yes" or "no")
[no]: yes

Enter key password for <scala>:
(RETURN if same as keystore password):

[unity@nc9042037056 Unity]$ /home/unity/IBM/LogAnalysis/ibm-java/jre/bin/
keytool -genkeypair -alias scala -keyalg RSA -keystore LdapSSLKeyStore.jks
-keysize 2048 -validity 7300
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: nc9042037056.tivlab.raleigh.ibm.com
What is the name of your organizational unit?
[Unknown]: IGA
What is the name of your organization?
[Unknown]: IGA
What is the name of your City or Locality?
[Unknown]: US
What is the name of your State or Province?
[Unknown]: US
What is the two-letter country code for this unit?
[Unknown]: US
Is CN=nc9042037056.tivlab.raleigh.ibm.com, OU=IGA, O=IGA, L=US, ST=US,
C=US correct? (type "yes" or "no")
[no]:
```

Create the JKS truststore

To create the JKS truststore:

1. Go to the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity directory.
2. Run the keytool. For example:

```
[scala@c25x0012 ~]$ /opt/IBM/SCALA/LogAnalysis/ibm-java/jre/bin/
keytool -genkeypair -alias scala -keyalg RSA -keystore
LdapSSLTrustStore.jks -keysize 2048 -validity 7300
Enter keystore password:
Re-enter new password:
What is your first and last name?
[Unknown]: <SCALA_SERVER_FQDN>
What is the name of your organizational unit?
[Unknown]: IGA
What is the name of your organization?
[Unknown]: IGA
What is the name of your City or Locality?
[Unknown]: US
What is the name of your State or Province?
[Unknown]: US
What is the two-letter country code for this unit?
[Unknown]: US
Is CN=IGA, OU=IGA, O=IGA, L=US, ST=US, C=US correct?
(type "yes" or "no")
[no]: yes
```



```
Enter key password for <scala>:  
(RETURN if same as keystore password):
```

Add the related LDAP root certificate to the truststore

To add the root certificate to the truststore, run the keytool:

```
[scala@c25x0012 Unity]$ /opt/IBM/SCALA/LogAnalysis/ibm-java/jre/bin/keytool  
-import -trustcacerts -alias root -file bluepages.crt  
-keystore LdapSSLTrustStore.jks  
Enter keystore password:  
Certificate already exists in system-wide CA keystore  
under alias <equifaxsecureca>  
Do you still want to add it to your own keystore?  
[no]: yes  
Certificate was added to keystore
```

Encode the trust and keystore passwords

1. Go to the <HOME>/IBM/LogAnalysis/wlp/bin/.
2. Run the security utility tool. For example, the password that is used here, t1v011, is the same as the one used to create the trust and keystores in the previous step:

```
[scala@c25x0012 bin]$ ./securityUtility encode t1v011  
{xor}K24pbzNu
```

Edit the server.xml file

To add the trust and keystore details to the server.xml file:

1. Go to the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity directory.
2. Edit the server.xml file.
3. Add the security settings before the defaultKeystore tag. For example:

```
<sslDefault sslRef="LDAPSSLSettings" />  
  
<ssl id="LDAPSSLSettings" keyStoreRef="LDAPKeyStore"  
trustStoreRef="LDAPTrustStore" />  
  
<keyStore id="LDAPKeyStore" location=  
"${server.config.dir}/LdapSSLKeyStore.jks"  
type="JKS" password="{xor}K24pbzNu" />  
<keyStore id="LDAPTrustStore" location=  
"${server.config.dir}/LdapSSLTrustStore.jks"  
type="JKS" password="{xor}K24pbzNu" />  
  
<!-- default keystore for certificates. located in  
<install home>/wlp/usr/servers/Unity/resources/security -->  
<!-- file name is key.jks . If it does not exist at startup  
it will be automatically created. -->  
<keyStore id="defaultKeystore"  
password="{xor}MzA4PjE+MyYrNjws" />
```

Configure the example data collector application ID

1. Go to the <HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/ directory.
2. Edit the unity.conf file. Specify the data collector ID and password. For example:

```
unity.data.collector.userid=LVH8JF631@nomail.relay.example.com  
unity.data.collector.password=pk6b7dg
```

where LVH8JF631@nomail.relay.ibm.com is the application ID that your manager requested in the DRMS application.

3. Go to the <HOME>/IBM/LogAnalysis/utilities/datacollector-client directory.

4. Edit the `javaDatacollector.properties` file, specifying the data collector user ID and password:

```
#The user ID to use to access the unity rest service
userid=L VH8JF631@nomail.relay.example.com
#The password to use to access the unity rest service
password=pk h b67dg
```

Enable the LDAP configuration

To enable the LDAP configuration, go to the `<HOME>/IBM/LogAnalysis/utilities` directory and run the following command:

```
ldapRegistryHelper.sh enable
```

Import certificates from the LDAP keystore in Log Analysis

Finally, you need to import certificates from the LDAP keystore in Log Analysis

1. Export the certificate from `LDAPSSLKeyStore.jks`.

```
<HOME>/IBM/LogAnalysis/ibm-java/bin/keytool -exportcert -keystore <HOME>/IBM/
LogAnalysis/wlp/usr/servers/Unity/LdapSSLKeyStore.jks -alias scala -file <HOME>/IBM/
LogAnalysis/wlp/usr/servers/Unity/ldap-clientcert.crt
```

2. Import the `ldap-clientcert.crt` certificate file:

```
<HOME>/IBM/LogAnalysis/ibm-java/bin/keytool -import -file <HOME>/IBM/LogAnalysis/wlp/usr/
servers/Unity/ldap-clientcert.crt -keystore ../jre/lib/security/cacerts -alias scala
```

Configuring Secure Sockets Layer (SSL)

To ensure that you can communicate in a secure way, you can configure SSL.

Configuring CA certificates for SSL

Use this example to help you to deploy your certificate authority (CA) certificates as part of your implementation of SSL.

Assumptions and prerequisites

This example assumes that you have purchased a CA certificate from a third-party vendor. It is intended to help you with your configuration. The exact steps can differ depending on your installation and configuration.

If you deployed a keystore certificate, you need to delete the old keystore file, the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/resources/security/key.js` file.

Deploying the certificate

To deploy this certificate, you complete the following steps:

1. Generate the key

The first step is not required for all installations. Log Analysis generates a key when it is installed. However, some CAs require a self-signed certificate with a specific name, for example for `-dname`.

If your CA requires a self-signed certificate with a specific name, then run one of the following commands depending on which signature algorithm you are using:

- For users of the SHA1withRSA signature algorithm:

```
./keytool -genkey -keystore ~/IBM/LogAnalysis/wlp/usr/
servers/Unity/resources/security/key.jks
-storepass loganalytics -keypass loganalytics -validity 365
-dname "CN=abc12345678.in.example.com, OU=IT, O=EXAMPLE LTD,
L=Bangalore,S=Karnataka, C=IN" -alias default -keyalg RSA
-sigalg SHA1withRSA -ext san=dns:localhost.localdomain,dns:abc12345678,
dns:abc12345678.example.com,dns:localhost,ip:1.234.56.78 -keysize <encryption-key-size>
```

Where *<encryption-key-size>* is the size of the encryption key; for example, 2048.

- For users of the SHA256withRSA signature algorithm:

```
./keytool -genkey -keystore ~/IBM/LogAnalysis/wlp/usr/
servers/Unity/resources/security/key.jks
-storepass loganalytics -keypass loganalytics -validity 365
-dname "CN=abc12345678.in.example.com, OU=IT, O=EXAMPLE LTD,
L=Bangalore,S=Karnataka, C=IN" -alias default -keyalg RSA
-sigalg SHA256withRSA -ext san=dns:localhost.localdomain,dns:abc12345678,
dns:abc12345678.example.com,dns:localhost,ip:1.234.56.78 -keysize <encryption-key-size>
```

Where *<encryption-key-size>* is the size of the encryption key; for example, 2048.

The keystore file for Log Analysis is `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/resources/security/key.jks`.

The value for the `-dname` parameter is the domain that your server is identified with. The details that are specified here are used to identify the server. For example, this parameter is specified as follows in this example:

```
-dname "CN=abc12345678.in.example.com, OU=IT, O=EXAMPLE LTD,
L=Bangalore,S=Karnataka, C=IN"
```

where CN is the common name. OU is the organizational unit. O is the organization. L is the location. S is the state or province. C is the country.

Note:

If your *<encryption-key-size>* is 4096 or greater, then the policy files in the Java SDK might not be able to handle the larger certificate key size. This causes the GUI to have problems loading, and for the following error to be seen in the Liberty logs (ffdc):

```
Stack Dump = java.lang.RuntimeException: Could not generate dummy secret
    at com.ibm.jsse2.C.z(C.java:488)
    at com.ibm.jsse2.ap.b(ap.java:476)
    at com.ibm.jsse2.ap.a(ap.java:44)
Caused by: java.security.InvalidKeyException: Illegal key size or default parameters
    at javax.crypto.Cipher.a(Unknown Source)
    at javax.crypto.Cipher.a(Unknown Source)
..
```

Run the following command to use a policy file that supports a larger key size:

```
cd <LA_HOME>
cp -p ./ibm-java/demo/jce/policy-files/unrestricted/US_export_policy.jar ./ibm-java/jre/lib/
security/
cp -p ./ibm-java/demo/jce/policy-files/unrestricted/local_policy.jar ./ibm-java/jre/lib/
security/
```

2. Export the self-signed certificate to a file

After you generate the keystore in the first step, a default self-signed certificate is generated with an alias called `default`. You need to export this certificate to a file. After this step is done, you can import the file into the `<HOME>/IBM/LogAnalysis/ibm-java` folder that is part of the folders that are created by Log Analysis when it is installed. Completing this step ensures that all the components of Log Analysis use the same certificate.

This step is required because you are generating a new keystore and this change requires you to refresh the public certificates for the clients.

If you do not delete your old keystore certificate before you generate the new one, and the older certificate used the same alias, that is `default`, an error can result. To avoid this you can delete the older certificates or you can change the alias value.

To export the certificate into a file, run the following command:

```
./keytool -exportcert
-keystore ~/IBM/LogAnalysis/
```

```
wlp/usr/servers/Unity/resources/security/key.jks
-alias default -file client.crt

password - loganalytics
```

3. Import the self-signed certificate

To import this certificate into the Java runtime environment keystore, enter the following command:

```
./keytool
-import -keystore ~/IBM/LogAnalysis/ibm-java/
jre/lib/security/cacerts -alias default -file client.crt

keystore password - changeit
```

If you installed remote instances of Log Analysis components like the EIF Receiver, IBM Tivoli Monitoring Log File Agent, or Logstash, you must import the certificate in the Java runtime environment on the remote servers.

4. Generate the Certificate Signing request (CSR) and send for signing

To generate the CSR, run the following command:

```
./keytool -keystore ~/IBM/LogAnalysis/wlp/usr/servers/Unity/
resources/security/key.jks -certreq -alias default
-keyalg rsa -file csr-req.txt
```

Send the CSR that you generated to your CA for signing. The CA sends you three files, a root file, an intermediate certificate, and a primary certificate.

5. Import the root file

To import the root file, enter the following command:

```
./keytool -import -trustcacerts -keystore ~/IBM/LogAnalysis/wlp/usr/
/servers/Unity/resources/
security/key.jks -alias theCARoot -file root.cer.txt

Enter keystore password:
Certificate already exists in system-wide CA keystore under alias
verisignclass3g5ca

Do you still want to add it to your own keystore? [no]: yes
Certificate was added to keystore
```

6. Import the intermediate certificate

To import the intermediate certificate, run the following command:

```
[yogesh@scm91135985 bin]$ ./keytool -import -trustcacerts -keystore
~/IBM/LogAnalysis/wlp/usr/servers/Unity/resources/security/key.jks
-alias theIntermediate
-file intermediate.cer.txt

Enter keystore password:
Certificate was added to keystore
```

7. Import the primary certificate

To import the primary certificate, run the following command:

```
[yogesh@scm91135985 bin]$ ./keytool -import -trustcacerts -keystore
~/IBM/LogAnalysis/wlp/usr/servers/Unity/resources/security/key.jks
-alias primaryCert
-file scm91135985.in.ibm.com.crt.txt

Enter keystore password:
Certificate reply was installed in keystore
```

8. Import the root and the intermediate certificates from your certificate authority to the truststore of IBM Java, with commands similar to the following:

Note: This step is only needed if the root and intermediate certificates are not already available in your Log Analysis Java cacerts file.

```
./keytool -import -keystore ~/IBM/LogAnalysis/ibm-java/jre/lib/security/cacerts -alias theRoot  
-file root.cer.txt  
keytool password - changeit  
./keytool -import -keystore ~/IBM/LogAnalysis/ibm-java/jre/lib/security/cacerts -alias theIntermediate  
-file intermediate.cer.txt  
keytool password - changeit
```

Configure Transport Layer Security (TLS)

To ensure that you can communicate in a secure way, you can configure TLS.

Configuring Transport Layer Security (TLS)

Configure the Secure Sockets Layer (SSL) configuration to use the TLS 1.2 protocol.

About this task

The Liberty profile supports a flexible SSL configuration that allows for multiple SSL settings, including SSL configuration to use the TLS 1.2 protocol.

Procedure

1. To stop Log Analysis, enter the following command:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
```

2. Open the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/server.xml file.
3. To configure your SSL configuration to use the TLS 1.2 protocol, edit the `ssl id` and `sslProtocol` line as follows.

```
<ssl id="defaultSSLConfig" sslProtocol="TLSv1.2" /
```

4. To restart Log Analysis, enter the following command:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
```

For more information about configuring SSL and TLS, and for scenario examples, see [Configuring SSL for Liberty](#).

Setting up Secure Shell to use key-based authentication

Secure Shell (SSH) is a cryptographic network protocol for secure data communication between different computers. You set up key-based authentication between the IBM Operations Analytics - Log Analysis servers and the remote computers to which it connects.

About this task

Benefits of using key-based authentication:

- Data is transferred across a secure channel.
- The administrator is no longer concerned about the password changes for the remote servers.
- The passphrase is independent of the individual server password policy.
- One passphrase is used for multiple servers. Only the public key file must be copied to the client server.

For more information you can view the man pages for **ssh-keygen** by running this command:

```
man ssh-keygen
```

Procedure

1. To generate public and private keys, enter the following command:

```
ssh-keygen -t rsa
```

or either of the following commands:

```
ssh-keygen
(This command generates the same results as ssh-keygen -t rsa.)
```

```
ssh-keygen -t dsa
```

(If you specify dsa, the generated keys include _dsa in their file names.)

The following example shows what a valid output might look like:

```
bash-3.2$  
bash-3.2$ ssh-keygen -t rsa  
Generating public/private rsa key pair.  
Enter file in which you want to save the key (/home/unity/.ssh/id_rsa):  
Enter passphrase (empty for no passphrase):  
Enter same passphrase again:  
Your identification has been saved in /home/unity/.ssh/id_rsa.  
Your public key has been saved in /home/unity/.ssh/id_rsa.pub.  
The key fingerprint is:  
4a:ef:d5:7a:d8:55:b3:98:a1:1f:62:be:dd:c4:60:6e unity@<variable>.example.com  
The key's randomart image is:  
+---[ RSA 2048]-----+  
|  
|  
|. S .o+.o |  
|. o =o+++. |  
|. . +o+E.o |  
|. ..o=.o |  
|. .o.. |  
+-----+  
bash-3.2$
```

Enter the passphrase. (The **Enter passphrase** field can remain blank to specify an empty passphrase.)

2. To view the contents of the public key file, run the following commands:

```
cd ~/.ssh
ls -l id_rsa*
cat id_rsa.pub
```

The command output is:

```
bash-3.2$  
bash-3.2$ cat .ssh/id_rsa.pub  
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAQDQg0/GGoxGzyC7AwjbnwP0hCaeztIRt6yhAg  
GKdwM7nb71iv0RgwT4/48E26K1Uz9HrI1W/j0K0JHQW  
vaAFibqelmqLdk9ctCE901ywTOPFcYeBYPUF9vp/MgaypGxVwDbw/e0SNPb7YAtZpjRoqeUq  
oYoKzFXXspQkxhdhcQfpx0RYMbQdGGg03hDCM2wr2Kp  
VtVniF1vDvUk14cfcRkUpR8aQNMiueCJgV3VHh1au/0Uo0YpH53NXKhN/sx8xdyTVsKQ1rhW8  
g07HIVc2Tf9ZF2gYXn/HbjE509xK/APu2nztt0h+Air  
JyT5jYMi/IvSi0zbPyc0p9WijPeG8r/v unity@<variable>.in.ibm.com  
bash-3.2$
```

3. Create a directory called `.ssh` on the remote server. Use this to store the public key.
4. Copy the public key file (`id_rsa.pub`) to the `.ssh` directory on the remote client:

```
scp /home/unity/.ssh/id_rsa.pub  
<username>@<remotehostname>:/  
<HOME>/.ssh/id_rsa.pub
```

where *<hostname>* is the system host name and *<username>* is the system user name.

5. Add the content of the public key to the authorized keys file on the remote host.

```

bash-3.2$ ssh <username>@<remotehostname>
bash-3.2$ cd ~/.ssh
bash-3.2$ cat id_rsa.pub >> authorized_keys
bash-3.2$ rm id_rsa.pub
bash-3.2$ exit

```

6. Ensure that there are no duplicate keys for the same client in the `authorized_keys` file.

7. Log in to the remote computer to ensure that key-based SSH is working:

```
ssh <username>@<hostname>
```

Enter the passphrase, if prompted.

```

bash-3.2$ bash-3.2$ ssh <username>@<remotehostname>
Enter passphrase for key '/home/unity/.ssh/id_rsa':
Last unsuccessful login: Mon Jul 15 14:22:37 2013 on ssh from <variable>.example.com
Last login: Mon Jul 15 14:26:54 2013 on ssh from <variable>.example.com
$

```

Configuration of key-based authentication is complete.

Results

The steps may not work because different versions of SSH are supported by the operating systems that are used by the remote servers. For more information about how to solve this issue, see the *Secure Shell (SSH) configuration does not work* topic in the *Troubleshooting IBM Operations Analytics - Log Analysis* guide.

Configuring single sign-on (SSO) with the Tivoli Integrated Portal

You can configure SSO authentication between the Tivoli Integrated Portal and IBM Operations Analytics - Log Analysis.

Before you begin

- The Tivoli Integrated Portal server and the IBM Operations Analytics - Log Analysis server must use the same LDAP server for authentication.
- The Tivoli Integrated Portal server must use a Lightweight Directory Access Protocol (LDAP) server for authentication.
- You must configure SSO for the Tivoli Integrated Portal. To configure SSO:
 1. Log in to the Tivoli Integrated Portal server
 2. In the **Security** area, click **Global security**.
 3. In the **Authentication** area, click **Single-sign on (SSO)**.
 4. Ensure that the **Enabled** check box is selected.
 5. The domain value that you must have to complete step 4 is displayed in the **Domain name** field. If this field is blank, enter the domain name and click **Apply**.

Procedure

1. To export the Lightweight Third-Party Authentication (LTPA) keys file from the Tivoli Integrated Portal, complete the following steps:
 - a. Log on to the Tivoli Integrated Portal as an administrator.
 - b. In the **Security** area, click **Global security**.
 - c. In the **Authentication** area, click **LTPA**.
 - d. In the **Cross-cell single sign on** area, enter a password for the keys file in the **Password** field. Confirm the password.
 - e. Create a blank plain text file to use as the keys file. Note the directory that you store the file in.

- f. Enter the location where the keys file that you created in the previous step is stored in the **Fully qualified key file name** field. The value must point to the properties file that contains the keys that you want to export. For example, for a Windows operating system, enter `C:\keys.properties`. For a Unix-based operating system, enter `<tip_home_dir>/profiles/TIPProfile`.
 - g. Click **Export keys**.
2. Add the Tivoli Integrated Portal LDAP realm to the IBM Operations Analytics - Log Analysis LDAP configuration. Ensure that the LDAP realm that is specified here is the same as the one used by Tivoli Integrated Portal.
To specify the realm, edit the `ldap_realm_property` property in the `ldapRegistryHelper.properties` file:


```
ldap_realm_property=<LdapRegistryRealm>
```

where `<LdapRegistryRealm>` is the realm that is used by the Tivoli Integrated Portal. To find this value:

 - a. Log on to the Tivoli Integrated Portal.
 - b. In the **Security** area, click **Global security**.
 - c. In the **User account repository** area, click **Configure**.
 - d. The LDAP realm value is displayed in the **Realm name** field. You specify this same value in the `ldapRegistryHelper.properties` file.
3. To add the updated realm to the LDAP configuration for IBM Operations Analytics - Log Analysis and to enable LDAP authentication, run the `ldapRegistryHelper.sh` script. For more information, see [“ldapRegistryHelper.sh command”](#) on page 38.
4. Configure LTPA on the Liberty Profile for the WebSphere® Application Server:
 - a. Copy the LTPA keys file that you exported from the Tivoli Integrated Portal server in step 1 to the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/resources/security` directory on the IBM Operations Analytics - Log Analysis server. The folder contains a default keys file. Do not change this file. Use a different name for your own key file.
 - b. Go to the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity` directory.
 - c. To add the SSO tag to the IBM Operations Analytics - Log Analysis server, add the following line to the `server.xml` file before the final `server` tag:


```
<webAppSecurity ssoDomainNames="<SSO_domain>" />
```

where `<SSO_domain>` is the SSO domain, for example `example.com`. This value must match the SSO domain that is used by the Tivoli Integrated Portal server. Specify the same value as the one that is entered in the **Domain name** field on the Tivoli Integrated Portal UI.
 - d. To add the LTPA tag to the IBM Operations Analytics - Log Analysis server, add the following line to the `server.xml` file before the final `server` tag:


```
<ltpa keysFileName="${server.output.dir}/resources/security/<ltpa_key_file>"
keysPassword="<keysPassword>" expiration="120" />
```

 - where `<ltpa_key_file>` is the LTPA key file, for example `example_ltpa.keys`.
 - `<keysPassword>` is the LTPA password that you entered in step 1 when you created the LTPA key file on the Tivoli Integrated Portal server.

(Optional) You can use the `unity_securityUtility` command that is in the `<HOME>/IBM/LogAnalysis/wlp/bin/` directory to generate an encrypted password. After you generate the encrypted password, enter it as the value for the `keysPassword` parameter.
5. Restart the IBM Operations Analytics - Log Analysis server and verify that the SSO connection between the two servers is working.

Results

To verify that the SSO connection is correctly set up, log in to the Tivoli Integrated Portal server. Open a new tab page in the browser and log in to IBM Operations Analytics - Log Analysis. If you are not prompted for the user name and password, the SSO connection is set up correctly. If you are prompted for the login details, the SSO connection is not configured correctly.

Configuring single sign-on (SSO) with Jazz for Service Management

If you want to integrate data from IBM Operations Analytics - Log Analysis with the Dashboard Application Services Hub component of Jazz for Service Management, you need to configure SSO between IBM Operations Analytics - Log Analysis and Jazz for Service Management.

Before you begin

- Jazz for Service Management server must use a Lightweight Directory Access Protocol (LDAP) server for authentication.
- Jazz for Service Management server and the IBM Operations Analytics - Log Analysis server must use the same LDAP server for authentication.
- You must configure SSO for the Jazz for Service Management server. For more information, see the [Configuring SSO on the application server](http://www-01.ibm.com/support/knowledgecenter/SSEKCU/welcome?lang=en) topic in the Jazz for Service Management Knowledge Center at <http://www-01.ibm.com/support/knowledgecenter/SSEKCU/welcome?lang=en>.

Procedure

1. Export the Lightweight Third-Party Authentication (LTPA) keys file from the Jazz for Service Management server.
For more information, see the [Exporting LTPA keys](http://www-01.ibm.com/support/knowledgecenter/SSEKCU/welcome?lang=en) topic in the Jazz for Service Management Knowledge Center at <http://www-01.ibm.com/support/knowledgecenter/SSEKCU/welcome?lang=en>.
2. Add the Jazz for Service Management LDAP realm to the IBM Operations Analytics - Log Analysis LDAP configuration. Ensure that the LDAP realm that is specified here is the same as the one used by Jazz for Service Management.

To specify the realm, edit the `ldap_realm_property` property in the `LdapRegistryHelper.properties` file:

```
ldap_realm_property=<LdapRegistryRealm>
```

where `<LdapRegistryRealm>` is the realm that is used by Jazz for Service Management.

3. To add the updated realm to the LDAP configuration for IBM Operations Analytics - Log Analysis and to enable LDAP authentication, run the `LdapRegistryHelper.sh` script. For more information, see [“LdapRegistryHelper.sh command”](#) on page 38.
4. Configure LTPA on the Liberty Profile for the WebSphere Application Server:
 - a. Copy the LTPA keys file that you exported from the Jazz for Service Management server in step 1 to the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/resources/security` directory on the IBM Operations Analytics - Log Analysis server. The folder contains a default keys file. Do not change this file. Use a different name for your own key file.
 - b. Go to the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity` directory.
 - c. To add the SSO tag to the IBM Operations Analytics - Log Analysis server, add the following line to the `server.xml` file before the final `server` tag:

```
<webAppSecurity ssoDomainNames="<SSO_domain>" />
```

where `<SSO_domain>` is the SSO domain, for example `example.com`. This value must match the SSO domain that is used by the Jazz for Service Management server. Specify the same value as the one that is entered in the **Domain name** field on the Jazz for Service Management UI.

- d. To add the LTPA tag to the IBM Operations Analytics - Log Analysis server, add the following line to the `server.xml` file before the final `server` tag:

```
<ltpa keysFileName="${server.output.dir}/resources/security/<ltpa_key_file>"
keysPassword="<keysPassword>" expiration="120" />
```

- where `<ltpa_key_file>` is the LTPA key file, for example `example_ltpa.keys`.
- `<keysPassword>` is the LTPA password that you entered in step 1 when you created the LTPA key file on the Tivoli Integrated Portal server.

(Optional) You can use the `unity_securityUtility` command that is in the `<HOME>/IBM/LogAnalysis/wlp/bin/` directory to generate an encrypted password. After you generate the encrypted password, enter it as the value for the `keysPassword` parameter.

5. Restart the IBM Operations Analytics - Log Analysis server and verify that the SSO connection between the two servers is working.

Results

To verify that the SSO connection is correctly set up, log in to the Jazz for Service Management server. Open a new tab page in the browser and log in to IBM Operations Analytics - Log Analysis. If you are not prompted for the user name and password, the SSO connection is set up correctly. If you are prompted for the login details, the SSO connection is not configured correctly.

Users and roles

You can create and modify users and roles in the IBM Operations Analytics - Log Analysis UI to assign role-based access control to individual users.

You can use the IBM Operations Analytics - Log Analysis UI to create and modify users and user roles to provide individual accounts and roles to users. This role-based access control enables the administrator to assign individual access and does not support user or object groups.

IBM Operations Analytics - Log Analysis includes the following default users and roles.

unityuser

This default user is assigned the default `unityusers` role. All users are assigned to the `unityusers` role by default.

unityadmin

This default user is assigned the default `unityadmins` role. This user has access to all data in IBM Operations Analytics - Log Analysis. No explicit permission needs to be set for this user.

The Log Analysis user registry is case sensitive. The LDAP user directory is not.

You can use the default users and roles as outlined in Table 1, or create new users and roles. For more information about creating users and roles, see *Create and modify users* and *Create and modify roles* in the *Postinstallation configuration* section of the *Configuring IBM Operations Analytics - Log Analysis* guide.

Table 7. Default users and roles	
Default user	Default role
unityuser	unityusers
unityadmin	unityadmins, unityusers

Users with the `unityuser` role can access the search workspace. Users with the `unityadmin` role can access the search and administration workspaces.

By default, all users are assigned the `unityuser` role.

The `unityadmin` user has access to all data. Only one `unityadmin` user with the `unityadmins` role is supported in IBM Operations Analytics - Log Analysis.

Note: To create or modify users or roles, you must have access to the administration workspace.

Creating a user



You can use the IBM Operations Analytics - Log Analysis UI to create a user.

Before you begin

To create new user accounts, you must have administrative access.

Procedure

To create a new user account, complete the following steps.

1. Open the IBM Operations Analytics - Log Analysis UI and click **Administrative settings**.
2. Select the **Users** tab.
3. Click the add icon  to open a new **Add user** pane.
4. Complete the new user details in the **Add user** pane.
5. To add a role to the user, click the add icon  in the **Add user** pane. Select the required role and click **OK**.
6. To save the new user, click **OK**.

Adding or deleting user roles




You can add or delete user roles in the IBM Operations Analytics - Log Analysis UI to create a user.

About this task

Adding and deleting user roles in the IBM Operations Analytics - Log Analysis UI enables management of the individual profiles of distinct users.

Procedure

To add or delete user roles, complete the following steps.

1. Open the IBM Operations Analytics - Log Analysis UI and click **Administrative settings**.
2. Select the **Users** tab.
3. Select the user that you want to edit.
4. Click the edit icon . This opens a new **Edit <username>** pane.
5. To add a user role to the selected user, click the add icon  in the **Edit <username>** pane.
6. To delete a user role from the selected user, click the delete icon  in the **Edit <username>** pane.
7. Select the required role and click **OK**.
8. To save your changes, click **OK**.

Editing a user


You can use the IBM Operations Analytics - Log Analysis UI to edit a user.

Before you begin

To edit user accounts, you must have administrative access.

Procedure

To edit a user account, complete the following steps.

1. Open the IBM Operations Analytics - Log Analysis UI and click **Administrative settings**.
2. Select the **Users** tab.
3. Click the edit icon . This opens a new **Edit <username>** pane.

4. Edit the user details in the **Edit <username>** pane.
5. To save your changes, click **OK**.

Changing a user password

Users passwords are changed by using the IBM Operations Analytics - Log Analysis UI.

Users can choose to change their own passwords or the unityadmin user can change the password of an existing user.

Changing your password

You can use the IBM Operations Analytics - Log Analysis UI to change your password.

Procedure

To edit your password, complete the following steps.

1. Open the IBM Operations Analytics - Log Analysis UI.
2. On the **Getting Started** page, expand the **username**.
3. Select **Change Password**.
4. Complete the fields Edit Password pane.
5. Click **OK**.


Changing a user password

You can use the IBM Operations Analytics - Log Analysis UI to change the password of an existing user.

Before you begin

To change the password of an existing user, you must have administrative access.

Procedure

1. To edit a user password, complete the following steps.
 - a) Open the IBM Operations Analytics - Log Analysis UI and click **Administrative settings**.
 - b) Select the **Users** tab.
 - c) Click the edit icon . This opens a new **Edit <username>** pane.
 - d) Change the users password details in the **Edit <username>** pane.
 - a) To save your changes, click **OK**.
2. If you want to change the unityadmin password, you must update the encrypted password in the following files to match the updated password.
 - a) To generate the encrypted password, use the `unity_securityUtility.sh` utility in the `<HOME>/IBM/LogAnalysis/utilities` directory.
For example:

```
unity_securityUtility.sh encode password
```

- `<HOME>/IBM/LogAnalysis/utilities/datacollector-client/javaDatacollector.properties`. For example:

```
#The password to use to access the unity rest service  
password={aes}EF712133E0677FEBB30624BA5EE62BC2
```

- `<HOME>/IBM/LogAnalysis/remote_install_tool/config/rest-api.properties`. For example:

```
ibm.scala.rest.password={aes}EF712133E0677FEBB30624BA5EE62BC2
```

- <HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/unity.conf. For example:

```
unity.data.collector.password={aes}EF712133E0677FE0B30624BA5EE62BC2
```

- <HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh. For example:

```
PASSWD={aes}EF712133E0677FE0B30624BA5EE62BC2
```

- b) If you change the password that is used by the unityuser, you must update the password parameter in the <HOME>/IBM/LogAnalysis/solr_install_tool/scripts/register_solr_instance.sh script.

```
password={aes}7A0B2401A8E29F37CD768CB78E205CAD
```

Deleting a user


To delete a user, complete this task.

Before you begin

To delete user accounts, you must have administrative access.

Note: If a user is deleted while they are logged in to IBM Operations Analytics - Log Analysis, their session continues until they log out. The user is not logged out automatically when they are deleted.

Procedure

1. Open the IBM Operations Analytics - Log Analysis UI and click **Administrative settings**.
2. Select the **Users** tab.
3. Select the user that you want to delete.
4. Click the delete icon .
5. A message opens asking you to confirm that you want to delete the selected users, click **OK**.

What to do next

After you delete a user, you can delete any alerts created by this user if you want. You can also set the alerts to inactive. To edit or delete the alert, open the **Manage Alerts** UI, select the alerts and click **Edit** or **Delete**. For more information, see [“Manage Alerts UI” on page 76](#).

Creating a role


You can use the IBM Operations Analytics - Log Analysis UI to create an application role.

Before you begin

To create new application roles, you must have administrative access.

Procedure

To create a new role, complete the following steps.

1. Open the IBM Operations Analytics - Log Analysis UI and click **Administrative Settings**.
2. Select the **Roles** tab.
3. Click the add icon .
4. Complete the new role details in the **Add Role** pane.
5. To save the new role, click **OK**.

Editing a role


You can use the IBM Operations Analytics - Log Analysis UI to edit role profiles.

Before you begin

To edit role profiles, you must have administrative access.

Procedure

To edit role profiles, complete the following steps.

1. Open the IBM Operations Analytics - Log Analysis UI and click **Administrative Settings**.
2. Select the **Roles** tab.
3. Select the role that you want to edit, and click the edit icon .
4. Modify the fields in the **Edit <username>** pane.
5. To save the changes to the role, click **OK**.

Adding users to roles




You can use the IBM Operations Analytics - Log Analysis UI to add users to new or existing role.

Before you begin

To add a user to a new or existing role, you must have administrative access.

Procedure

To add a user to a new or existing role, complete the following steps.

1. Open the IBM Operations Analytics - Log Analysis UI and click **Administrative Settings**.
2. Select the **Roles** tab.
3. Select the role that you want to add users to:
 - To add a user to a new role, click the add icon .
 - To add a user to an existing role, select the role and click the edit icon .
4. Select the **Assign Users to Role** tab in the **Add Role** or **Edit <username>** pane.
5. Click the add icon .
6. Select the users that you want to add to the role from the list, and click **OK**.

To clear a selected user, hold down the **Ctrl** or **Command** key, and click the user row.
7. To save the changes to the user, click **OK**.

Deleting a user from a role


You can use the IBM Operations Analytics - Log Analysis UI to delete a user from a role.


Before you begin

To delete a user from a role, you must have administrative access.

Procedure

To delete a user from a role, complete the following steps.

1. Open the IBM Operations Analytics - Log Analysis UI and click **Administrative Settings**.
2. Select the **Roles** tab.
3. Select the role that you want to edit, and click the edit icon .
4. Select the **Assign Users to Role** tab in the **Edit <username>** pane.

5. Select the users that you want to delete, and click the delete icon .
6. To save the changes to the role, click **OK**.

Adding permissions to roles

You can use the IBM Operations Analytics - Log Analysis UI to add permissions to new or existing roles.

Before you begin

To add permissions to a new or existing role, you must have administrative access.



The default permission is none. Users with the default permissions do not have access to the data.

Note: If you run a saved search, or create a dashboard or Custom Search Dashboard that includes a list of data sources, the search or dashboard is rendered with data from data sources for which you have permission.


Procedure

To add permissions to a new or existing role, complete the following steps.

1. Open the IBM Operations Analytics - Log Analysis UI and click **Administrative Settings**.
2. Select the **Roles** tab.
3. Select the role that you want to add permissions to:

- To add permissions to a new role, click the add icon .
- To add permissions to an existing role, select the role and click the edit icon .

4. Select the **Assign Permissions to Role** tab in the **Add Role** or **Edit <username>** pane.

5. Click the add icon .

6. Select the permissions that you want to add to the role from the list, and click **OK**.

Note: IBM Operations Analytics - Log Analysis only supports read permissions for datasource objects. Users with read permissions can run ingestion and deletion operations.

To clear a selected permission, hold down the **Ctrl** or **Command** key, and click the permission row.

7. To save the changes to the role, click **OK**.

Deleting permissions from roles



You can use the IBM Operations Analytics - Log Analysis UI to delete permissions from roles.

Before you begin

To delete permissions from roles, you must have administrative access.

Note: If you run a saved search, or create a dashboard or Custom Search Dashboard that includes a list of data sources, the search or dashboard is rendered with data from data sources for which you have permission.

Procedure

1. Open the IBM Operations Analytics - Log Analysis UI and click **Administrative Settings**.
2. Select the **Roles** tab.
3. Select the role from which you want to delete permissions, and click the edit icon .
4. Select the **Assign Permissions to Role** tab in the **Edit <username>** pane.
5. Select the permissions that you want to delete, and click the delete icon .
6. To save the changes to the role, click **OK**.

System

Before you can use Log Analysis, you need to configure the system.

Configuring the Indexing Engines

Log Analysis uses an Apache Solr-based indexing engine to index log records.

You can install the Indexing Engines on the same server as Log Analysis. Typically, you would do this in a smaller deployment.

To improve performance, you can install the Indexing Engines on dedicated remote servers.

Prerequisites

Before you can use the Indexing Engines, you must download a license for Apache Solr from Passport Advantage at <http://www-01.ibm.com/software/lotus/passportadvantage/> and complete the steps that are outlined in the readme file.

Installing Apache Solr on remote machines

After you install IBM Operations Analytics - Log Analysis, you can use the Apache Solr remote installer to install Apache Solr on a remote machine.

About this task

If no local instances of Apache Solr exist, then you need to install the instances on the remote machine as soon as you install IBM Operations Analytics - Log Analysis. If there is a local instance of Apache Solr, you can install the remote instances whenever you want.

You must use a non-root user to run the script.

You cannot use the installer to install Apache Solr on a local machine.

You cannot use the installer to install multiple Apache Solr nodes on a single remote machine.

Ensure that the Apache Solr ports are open and free from any firewall restrictions on the remote machine.

To install Apache Solr on multiple remote machines, run the script separately for each remote machine. You cannot use the installer to install instances of Apache Solr simultaneously or in parallel.

Procedure

1. Change the directory to <HOME>/IBM/LogAnalysis/solr_install_tool. For example:

```
cd <HOME>/IBM/LogAnalysis/solr_install_tool
```

2. To run the remote_deploy.sh script, enter the following command:

```
./remote_deploy_solr.sh -install
```

3. The script prompts you for the following information:

Remote Hostname in FQDN format

Enter the Fully Qualified Domain Name (FQDN) of the remote host.

Username

Enter the user name.

Password

Enter the password if password-less SSH authentication is disabled. If password-less SSH is enabled between the IBM Operations Analytics - Log Analysis server and the remote host, the script reads the values that are specified in the `ssh_config.properties` file in the <HOME>/IBM/LogAnalysis/utilities/config directory. For more information, see the *Setting up Secure Shell to use key-based authentication* topic in the IBM Operations Analytics - Log Analysis information center.

SSH Port

Enter the remote machine port that is used for SSH. To use the default value of 22, press enter.

Top-level Installation Directory

To use the default value, which is <HOME>, press enter. Alternatively, you can enter the path to the directory where you want to install the DE.

Apache Solr Search Port

To use the default value, 9989, press enter. To use another port, enter the port number for another valid port. Do not enter a port that is being used for something else.

Apache Solr Query Service Port

To use the default value, 7205, press enter. To use another port, enter the port number for another valid port. Do not enter a port that is being used for something else.

4. To start the installation, press enter. In most cases, the installation takes about 5 minutes to complete.

Results

The results of the installation are output in the log file in the <HOME>/IBM/LogAnalysis/solr_install_tool/logs/ManageSolrnodes.log file.

To view the status for the instances of Apache Solr that are installed remote machines, run the `unity.sh -status` command.

Example

Here is an example script output:

```
Remote Hostname in FQDN format:12345.example.com
username:unity
password:*****
SSH port: [22]
Top-level Installation Directory: [/home/unity]
Solr Search Port: [9989]
Solr Query Service Port: [7205]

Script is ready for remote installation of Solr:
Review the following inputs ....
-----
Remote Host Name: 12345.example.com
Remote User Name: unity
Remote SSH Port: 22
Top-level remote installation directory: /home/unity
Solr v9.0 - remote installation directory:
/home/unity/IBM/LogAnalysis
Solr - remote ports: 9989, 7205
-----
['q' - Abort]['Enter' - Install]

Sat Nov 16 03:08:38 CST 2013 Starting remote installation of Solr
, this will take couple of minutes to complete ....
Sat Nov 16 03:08:38 CST 2013 Waiting for remote installation to complete ....
Sat Nov 16 03:11:47 CST 2013 Successfully installed Solr
Solr on remote host:12345.example.com ....
```

Removing Apache Solr instances

Before you remove an installation of IBM Operations Analytics - Log Analysis, you must remove Apache Solr.

About this task



Warning: Do not remove Apache Solr if IBM Operations Analytics - Log Analysis is still being used. IBM Operations Analytics - Log Analysis does not function properly when any instances of Apache Solr are removed. For this reason, only remove Apache Solr when you are about to uninstall IBM Operations Analytics - Log Analysis.

If you installed Apache Solr locally and remotely, remove the local instance first, then remove the remotely installed instances.

This process uses Installation Manager to remove Apache Solr instances. You can also do so silently. To run the silent removal, run following `imcl -c` command, enter 3 to modify the installation, and remove the instance.

Procedure

1. Change the directory to `<HOME>/IBM/LogAnalysis/solr_install_tool`. For example:

```
cd <HOME>/IBM/LogAnalysis/solr_install_tool
```

2. To run the `remote_deploy.sh` uninstall script, enter the following command:

```
./remote_deploy.sh -uninstall
```

3. The script prompts you for the following information:

Remote Hostname in FQDN format

Enter the Fully Qualified Domain Name (FQDN) of the remote host.

Username

Enter the user name.

Password

Enter the password if password less SSH authentication is disabled. If password less SSH is enabled between the IBM Operations Analytics - Log Analysis server and the remote host, the script reads the values that are specified in the `ssh_config.properties` file in the `<HOME>/IBM/LogAnalysis/utilities/config` directory. For more information, see the *Setting up Secure Shell to use key-based authentication* topic in the IBM Operations Analytics - Log Analysis information center.

SSH Port

Enter the remote machine port that is used for SSH. To use the default value of 22, press enter.

Top-level Installation Directory

To use the default value, which is `<HOME>/IBM/LogAnalysis`, press enter. Alternatively, you can enter the path to the directory where Apache Solr is installed.

4. To start the removal, press enter. You can view the logs in the `<HOME>/IBM/LogAnalysis/solr_install_tool/logs` directory.

Results

When all the remote nodes are removed, you can safely uninstall IBM Operations Analytics - Log Analysis.

Extending storage space available to Apache Solr

You can add more Apache Solr storage directories outside the initial IBM Operations Analytics - Log Analysis Apache Solr installation location if the disk on which Apache Solr was installed reached maximum capacity.

Before you begin

Ensure that the Apache Solr storage directories are present on all Apache Solr servers and are writable.

About this task

Switching to a new Apache Solr directory is not instantaneous. Therefore, it is to monitor the disk usage of your Apache Solr directory to ensure that extra directories are added before the current storage directory reaches maximum capacity.

Procedure

To enable the storage extension capability, complete the following steps.

1. Stop IBM Operations Analytics - Log Analysis with the following command.

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
```

2. Open the `unitysetup.properties` file in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF` directory.

3. Add the following property to the directory

```
ENABLE_STORAGE_RELOCATION=true
```

4. Create the following properties file

```
<HOME>/solrConfigs/storageConfig.properties
```

For example,

```
/home/unity/IBM/LogAnalysis/solrConfigs/storageConfig.properties
```

5. Open the `storageConfig.properties` file and add the following property to the file.

```
SOLR_STORAGE_DIR=storage-path-on-solr-nodes
```

For example,

```
SOLR_STORAGE_DIR=/opt/scala/ext_storage
```

6. Restart IBM Operations Analytics - Log Analysis with the following command.

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
```

Results

The new IBM Operations Analytics - Log Analysis configuration file enables the specification of custom data storage locations. The new locations are written to when IBM Operations Analytics - Log Analysis crosses the default boundary of 1 day.

Changing the default boundary for creating Apache Solr collections

You can change the default boundary that is associated with extending Apache Solr storage space depending on your business needs.

Procedure

1. Open the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties` file.
2. Locate and modify the value of the `COLLECTION_ASYNC_WINDOW` property from the default value of 1d (1 day).

Note: The minimum property size is 6h.

The boundary size can be specified in minutes (m), hours (h), or days (d).

3. Restart IBM Operations Analytics - Log Analysis with the following command.

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
```

Globalization

Some of the text in IBM Operations Analytics - Log Analysis is only available in English. Some of these texts can be globalized but others cannot. Read the following information to get an overview of what you can and cannot manually globalize.

Insight Packs

Insight Packs are not globalized by default. To localize Insight Pack content, you must configure the content manually. For more information about localizing Insight Packs content, see *Globalizing Custom Search Dashboard* and *Globalizing dashboards, chart labels, and index configuration fields* in the *Globalization* section of the *Configuring* guide.

To facilitate the ingestion of non-English-language log files, you must change the IBM Tivoli Monitoring Log File Agent locale to match the value that is used by the source application to create the log files. For more information, see *Ingestion of non-English-language log files* in the *Loading and streaming data* guide.

You can change the default time zone that is used by IBM Operations Analytics - Log Analysis to create index fields when data is loaded. For more information about configuring time zone, see *Configuring the time zone for indexing* in the *Configuration reference* guide.

IBM Operations Analytics - Log Analysis supports multiple languages. For a full list of supported languages, see *Supported languages* in the *Configuration reference* guide.

Limitations

The following data cannot be globalized:

Log file data

Log file data is always displayed in the language that it is used in the original log files. This text is displayed in the **Search** UI.

Installation directory and files

The directory where IBM Operations Analytics - Log Analysis is installed and the files that are created during the installation are only available in English.

IBM Operations Analytics - Log Analysis log files

The log files that are generated by IBM Operations Analytics - Log Analysis and the associated applications such as IBM Tivoli Monitoring Log File Agent, Indexing Engines, and the WebSphere Application Server are only available in English.

Artifacts that are created in the Admin UI

Artifacts such as Source Types, Collections, Data Sources, File Sets, and Rule Sets are only available in English. If a user creates one of these objects in another language, the artifacts are only available in that language.

Sample log files

The log files that are used by the sample scenarios are only available in English.

IBM Operations Analytics - Log Analysis scripts

Scripts such as `unity.sh`, `EIFUTIL.sh`, and `LDAPRegistryHelper.sh` are only available in English and cannot be globalized. This limitation encompasses Shell, Python, and Ruby scripts.

Time and date formats

The time and date format can only be changed for the entire IBM Operations Analytics - Log Analysis installation.

Related concepts

[“Ingestion of non-English-language log files” on page 238](#)

To facilitate the ingestion of non-English-language log files, you must change the IBM Tivoli Monitoring Log File Agent locale to match the value that is used by the source application to create the log files.

[“Configuring the time zone for indexing” on page 72](#)

You can change the default time zone that is used by Log Analysis for to create index fields when data is loaded.

Related tasks

[“Globalizing Custom Search Dashboard” on page 69](#)

Complete this procedure to globalize Custom Search Dashboard to ensure that messages, errors and exceptions, and the Custom Search Dashboard generated output is globalized.

[“Globalizing Insight Pack content, dashboards, chart labels, and index configuration fields” on page 70](#)

Complete this procedure to globalize index configuration fields, dashboards, and chart labels.

Related reference

[“Supported languages” on page 134](#)

IBM Operations Analytics - Log Analysis supports multiple languages.

Globalizing Custom Search Dashboard

Complete this procedure to globalize Custom Search Dashboard to ensure that messages, errors and exceptions, and the Custom Search Dashboard generated output is globalized.

About this task

To enable globalization of the Custom Search Dashboard generated output, the IBM Operations Analytics - Log Analysis framework passes the locale that is used by your application.

For information about the limitations of the globalization process, see [“Globalization” on page 67](#).

To globalize your Custom Search Dashboard, complete the following steps:

Procedure

1. Open the script that the Custom Search Dashboard is based on.
In most cases, this script is the <custom_search_dashboard_name>.app file that is stored in the relevant Insight Pack folder, for example <HOME>/IBM/LogAnalysis/unity_content/WindowsOSEventsInsightPack_<version>/unity_apps/apps. The Custom Search Dashboard can be based on other languages, such as Python and Java.
2. To extract the locale information that is passed from IBM Operations Analytics - Log Analysis to your Custom Search Dashboard, you must add a JSON compatible code to your Custom Search Dashboard script. For example:

```
{
  "parameters": [
    {}
  ],
  "_fwParameters": [
    {
      "name": "locale",
      "value": "<locale>",
      "type": "String"
    }
  ]
}
```

The following example is in the Python language and it shows how you can extract the locale information that is passed to the Insight Pack or Custom Search Dashboard from IBM Operations Analytics - Log Analysis:

```
if len(sys.argv) > 1:
    filename = str(sys.argv[1])
    fk = open(filename, "r")
    data = json.load(fk)
    locale = data["_fwParameters"][0]["value"]
```

In this example, the locale is sent to the script from IBM Operations Analytics - Log Analysis in the following format:

```
{
  "parameters": [
    {}
  ],
  "_fwParameters": [
    {
      "name": "locale",
      "value": "<locale>",
      "type": "String"
    }
  ]
}
```

where <locale> is the locale that you want to use. For example, "value": "en_US".

3. Save and start the application to see the globalized Custom Search Dashboard.

Results

The extracted locale globalizes messages, errors and exceptions, and the Custom Search Dashboard generated output.

Globalizing Insight Pack content, dashboards, chart labels, and index configuration fields

Complete this procedure to globalize index configuration fields, dashboards, and chart labels.

About this task

Dashboard, charts, and index configuration fields are globalized by using a resource bundle. The resource bundle is based on the Java resource bundle mechanism. The Insight Pack developer creates the resource bundle in supported languages. There is one resource bundle for the Insight Pack that contains keys for all the artifacts.

Insight Packs are available in English. To globalize these objects and the associated content into another language, you must configure the locale and create a resource bundle.

To globalize your Insight Pack content, dashboards, charts, and index configuration fields complete the following steps:

Procedure

1. Create a folder that is named `i18n` in the directory where your Insight Pack is stored.
2. Create a resource bundle file in the `i18n` folder. The Insight Pack name that you use in the resource bundle file must match the Insight Pack exactly.

```
<Insight_Pack_Name>_locale.properties
```

where `<Insight_Pack_Name>` is the exact name of the Insight Pack that you want to globalize.

3. Specify the keys for each artifact that you want to globalize. Keys are values of the fields to be translated for globalization. For example, the value of the "name" field for your Custom Search Dashboard or dashboard in the `.app` file. If you do not specify a value in the resource bundle file, the existing name is used by default. Keys that are not specified in the resource bundle are displayed in English. The resource bundle supports three different types of specification:

Global keys

`key=value`

Global key applies to all artifacts in the Insight Pack.

Global artifact keys

`artifact_type.key=value`

Global artifact key applies to all artifacts under a specific artifact type.

Specific artifact keys

`artifact_type.artifact_name.key=value`

Specific artifact keys override the general key.

The following artifact_types are supported:

- `sourcetype`
- `customapp`

The artifact name is the name that you created, for example `WASSystemOut`.

You can specify globalized text for each of the following artifacts:

Index configuration

To globalize the field names in the index configuration, specify the field names and the localized text in the resource bundle:

```
sourcetype.<Source_Type_name>.<Index_Configuration_Field> = Localized_text
```

For example:

```
sourcetype.WASSystemOut.severity = severity_translated  
sourcetype.WASSystemErr.message = message_translated  
sourcetype.WASSTrace.msgclassifier = msgclassifier_translated
```

Custom Search Dashboard name

To specify a globalized version of a Custom Search Dashboard name, you must specify the name that is used in the Custom Search Dashboard specification file in the resource bundle. For example, the name in the Custom Search Dashboard specification file is:

```
"name": "Error Analysis"
```

You specify the following information in the resource bundle:

```
customapp.<custom_app_name>.Error\ Analysis = Log Severity Trend
```

Log Severity Trend is displayed as the application name on the UI.

Tags in the Search Dashboard

When you create a Custom Search Dashboard, it is displayed in the **Search Dashboards** pane in the **Search** UI. The Custom Search Dashboard is grouped under a directory. The name of this directory is globalized by using tags. To globalize this tag name, you need to specify the tag in the resource bundle as:

```
customapp.tag.<Insight_Pack_Name> = SEVERITY ANALYSIS
```

where *<Insight_Pack_Name>* is the name of the Insight Pack.

For example:

```
customapp.tag.NewAppPack_v1.1.0.0 = SEVERITY ANALYSIS
```

The tag name SEVERITY ANALYSIS is displayed on the UI.

Chart titles

The chart titles are specified as follows in the chart specification:

```
spec: "title": "Event Diagnostics"
```

To globalize the title, add it to the resource bundle:

```
customapp.<custom_app_name>.Event\ Diagnostics = Error while ingesting data
```

Labels for chart axis

The labels that are used to identify chart axis are specified by the `label` parameter in the chart specifications. For example:

```
"parameters": [{ "type": "Axis", "name": "xaxis", "label": "xlabel" },  
{ "type": "Axis", "name": "yaxis", "datatype": "number",  
  "label": "ylabel" }, ],
```

The new axis labels from chart specification should be used in your Custom Search Dashboard specification.

```
"charts": [ { "title": "Event Diagnostics", "labels":  
  { "xlabel": "timestamp", "ylabel": "Counter" }, }, ]
```

To globalize the label names, specify the globalized text in the resource bundle in the following format:

```
customapp.<custom_app_name>.<Label_name> = <Localized_name>
```

where *<Label_name>* is the name that is specified in the chart specifications. *<Localized_name>* is the name that you want to display on the UI. For example:

```
customapp.<custom_app_name>.Counter = Counter_t
```

Counter_t is the axis name that is displayed on the UI.

4. Package the Insight Pack as the next version and use `pkg_mngt.sh` utility to update the Insight Packs.

Enabling facet cache for wildcard searches

If you use the wildcard search term (*) to search data that is older than 1 day, you can configure Log Analysis to count facets before they are indexed for search. This setting can help optimize this type of search.

Procedure

1. Open the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties` file.
2. Change the `ENABLE_SOLR_FACET_CACHE=false` parameter to true.
3. Save the file.
4. To restart Log Analysis, enter the following command:

```
./<HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
```

Configuring the time zone for indexing

You can change the default time zone that is used by Log Analysis for to create index fields when data is loaded.

If you want to change the time zone that is used to create index fields when data is loaded, you need to edit the `unitysetup.properties` file before you load any data. You cannot change the default time zone after you load data. If you do, this action causes errors in the log file records. For more information, see *Changing the index time zone*.

You can also change the time zone that is used for specific search results at any time. If you want to view a specific set of search results in a particular time zone, you can change the time zone in the Search workspace. You can also set a default search time zone.

For more information, see *Changing the search time zone* topic in the *Using* guide.

Related tasks

[“Changing the index time zone” on page 72](#)

Log Analysis uses Coordinated Universal Time (UTC) as the default time zone for indexing during data loading. To change the default time zone, complete this procedure.

Changing the index time zone

Log Analysis uses Coordinated Universal Time (UTC) as the default time zone for indexing during data loading. To change the default time zone, complete this procedure.

About this task

You must change this setting after you install IBM Operations Analytics - Log Analysis but before you load any data, including the sample data that is provided on the **Getting Started** page.

Note: You cannot change this time zone value after you load data. IBM Operations Analytics - Log Analysis cannot resolve the different time stamps and this conflict causes errors in the search that cannot be resolved. After you change the timezone and load data, do not change the timezone again.

IBM Operations Analytics - Log Analysis supports the Java timezone classification. For a list of supported timezone names, see *Supported time zone names*.

Procedure

1. Install Log Analysis. Do not load or install any data.
2. To stop Log Analysis, enter the following command:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
```

3. To change the default timezone value, edit the UNITY_TIME_ZONE parameter in the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties file.
For example, to change to Central European Time (CET), edit the parameter as follows:

```
UNITY_TIME_ZONE=Europe/Paris
```

4. Save your changes.
5. To restart Log Analysis, enter the following command:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
```

What to do next

After you change the default time zone and load data, do not change it again. If you want to change the time zone for searches, you can change it on the Search UI.

For more information, see *Changing the time zone for searches* topic in the *Using* guide.

Related reference

[“Supported time zone names” on page 119](#)

IBM Operations Analytics - Log Analysis uses Coordinated Universal Time as the default timezone.

Standard Managing alerts

You can use the alert management features of Log Analysis to monitor real-time data ingestion and trigger events based on specified conditions.

Use the alert management feature to help you to complete the following tasks:

- Generate alerts in real time as data is streamed into Log Analysis.
- Specify conditions and trigger actions such as sending an email notification, running a custom script, or logging an alert. You can also specify custom actions.
- Detect user specified conditions when the data is streamed into Log Analysis and trigger custom actions when the conditions are met.
- Index the generated alerts for reporting.
- If you also use IBM Tivoli Netcool/OMNIBus, you can generate alerts for EIF events or SNMP traps. For more information, see [“Configuring alerts for SNMP and EIF events” on page 79](#).

To create an alert, you need to specify a condition and an action. Use the **Manage Alerts** user interface (UI) to create, edit, and delete alerts, alert actions, and conditions. Only the `unityadmin` user can edit all alerts.

For more information about configuring the emails that are sent by the email alert action, see [“Configuring email alert actions” on page 84](#).

Standard Creating alerts

To trigger actions based on events, create an alert.

About this task

Only Standard Edition users can use this feature.

Procedure




1. To open the **Manage Alerts** UI, click the **Manage Alerts** icon ().
2. To create an alert, click **Create**.
3. Enter a name.
4. To ensure that the alert is active, select the **Enabled** check box.
5. Select a severity level. When this level is reached, the alert action is triggered.
6. You can also enter a note.
7. To create a condition, click **Create condition**
8. Select a template and enter the required keywords or search queries. The table summarizes the conditions.

Table 8. Conditions	
Condition	Description
Keyword match	Use this condition to trigger an alert when a keyword is found in all or only in the specified data sources. You can enter a keyword or you can enter a search query.
Keyword match based on threshold	Use this condition to trigger an alert when a specified number of keywords or search results occur in all or specified data sources during a specified time period. Enter the keyword or search query, specify the number of occurrences that are required to trigger the alert in the time period that you specify in seconds, minutes, or hours.
Keyword match with de-duplicates	Use condition to trigger an alert when a keyword or search query occurs in all or the specified data sources during the specified time period. If a condition is met, a single alert is sent. Log Analysis does not send multiple, duplicate actions.
Co-Occurrence Match	Use this condition to trigger an alert when 2 or more keywords or query search results occur during the same specified time period.
Anomaly Detection	Use this condition to trigger an alert when a change is detected in a specified field in a data source.
Anomaly Detection based on threshold	Use this condition to trigger an alert when a specified value occurs a specified number of times in the time period.

9. Save the condition.
10. Select the action that occurs when the alert is triggered.
The options are explained in the table.

Table 9. Alert actions	
Action	Description
Index	<p>Use this action to index any alerts in the _alerts data source. This does not require any configuration. The _alerts data source contains 3 indexed fields:</p> <p>conditionName The name of the condition.</p> <p>conditionsDatasource The name of the data source which met the condition.</p> <p>timestamp The time that the condition was met.</p>
Send Email	<p>Use this action to send an email. You can select a template or enter the sender, receiver, subject prefix and body text manually. If you want to send the mail to multiple recipients, you need to separate each address with a comma.</p> <p>Before you can use the email action, you need to configure the <HOME>/wlp/usr/servers/Unity/apps/Unity.war/configs/AlertActionEmailConfig.json file.</p>
Write to Log	<p>Use this action to record triggered alerts in a specific log file. You can select a recently viewed log file or you can enter the log path file manually. The triggered alerts are updated every 10 seconds.</p>
Script	<p>Use this action to run a script when a condition is met. You can select a recently viewed script or you can enter the details manually. You enter the directory where the script is stored, the directory where the script is run, and any command line parameters that need to be passed to the script when it is run.</p>

Table 9. Alert actions (continued)	
Action	Description
SNMP Trap	<p>Use this action to send SNMP trap events. You need to specify the following values:</p> <p>SNMP Server Enter the host name of the SNMP server.</p> <p>SNMP Port Enter the port that is used by the SNMP server.</p> <p>SNMP Version Displays the supported version of SNMP. This field is read only. Verion 2c is the only supported version.</p> <p>SNMP community string Enter the community string for the SNMP receiver. The default value is "public".</p> <p>SNMP Transport Protocol Specify a transport protocol. UDP and TCP are the supported transport protocols.</p>
EIF Event	Use this action to trigger an EIF event. You need to specify the EIF server and port.

11. Click **Create**.

Standard Manage Alerts UI

Use the **Manage Alerts** user interface (UI) to create, edit, and delete alerts, conditions, and alert actions. Only Standard Edition users can use this feature.

Buttons, fields, and check boxes

Table 10. Buttons, fields and check boxes on the Manage Alerts UI	
Button, field or check box	Description
Create New	Create an alert.
Edit	Edit an existing alert.
Delete	Delete an existing alert.
Search box	Search for an existing alert.
Refresh icon	Refresh the results table to include any new alerts.

Columns

Table 11. Columns on the Manager Alerts UI	
Column	Description
Selection check box	Use this check box to select an alert for editing.
Status	Indicates whether the alert is active or inactive.
Alert Name	The name of the alert.
Severity	The level of severity.

<i>Table 11. Columns on the Manager Alerts UI (continued)</i>	
Column	Description
Author	The person who created the alert action template.
Condition Template	The condition template that is used by the alert.
Actions	The actions that are triggered by the alert.

Alerts editor

<i>Table 12. Fields and check box on the Alerts editor</i>	
Fields and check box	Description
Alert Name	Enter a name for the alert.
Last Modified	The time and date when the alert was last modified. This field is read only.
Enabled check box	To deactivate the alert, clear this check box.
Severity	Select the severity level of the alert.
Author	The user who created the alert. This field is read only.
Notes	Enter any notes that you want to add.

Conditions editor

<i>Table 13. Conditions</i>	
Condition	Description
Keyword match	Use this condition to trigger an alert when a keyword is found in all or only in the specified data sources. You can enter a keyword or you can enter a search query.
Keyword match based on threshold	Use this condition to trigger an alert when a specified number of keywords or search results occur in all or specified data sources during a specified time period. Enter the keyword or search query, specify the number of occurrences that are required to trigger the alert in the time period that you specify in seconds, minutes, or hours.
Keyword match with de-duplicates	Use condition to trigger an alert when a keyword or search query occurs in all or the specified data sources during the specified time period. If a condition is met, a single alert is sent. Log Analysis does not send multiple, duplicate actions.
Co-Occurrence Match	Use this condition to trigger an alert when 2 or more keywords or query search results occur during the same specified time period.
Anomaly Detection	Use this condition to trigger an alert when a change is detected in a specified field in a data source.

Table 13. Conditions (continued)	
Condition	Description
Anomaly Detection based on threshold	Use this condition to trigger an alert when a specified value occurs a specified number of times in the time period.

Alert actions editor

Table 14. Alert actions	
Action	Description
Index	<p>Use this action to index any alerts in the _alerts data source. This does not require any configuration. The _alerts data source contains 3 indexed fields:</p> <p>conditionName The name of the condition.</p> <p>conditionsDatasource The name of the data source which met the condition.</p> <p>timestamp The time that the condition was met.</p>
Send Email	<p>Use this action to send an email. You can select a template or enter the sender, receiver, subject prefix and body text manually. If you want to send the mail to multiple recipients, you need to separate each address with a comma.</p> <p>Before you can use the email action, you need to configure the <HOME>/wlp/usr/servers/Unity/apps/Unity.war/configs/AlertActionEmailConfig.json file.</p>
Write to Log	<p>Use this action to record triggered alerts in a specific log file. You can select a recently viewed log file or you can enter the log path file manually. The triggered alerts are updated every 10 seconds.</p>
Script	<p>Use this action to run a script when a condition is met. You can select a recently viewed script or you can enter the details manually. You enter the directory where the script is stored, the directory where the script is run, and any command line parameters that need to be passed to the script when it is run.</p>

Table 14. Alert actions (continued)

Action	Description
SNMP Trap	<p>Use this action to send SNMP trap events. You need to specify the following values:</p> <p>SNMP Server Enter the host name of the SNMP server.</p> <p>SNMP Port Enter the port that is used by the SNMP server.</p> <p>SNMP Version Displays the supported version of SNMP. This field is read only. Verion 2c is the only supported version.</p> <p>SNMP community string Enter the community string for the SNMP receiver. The default value is "public".</p> <p>SNMP Transport Protocol Specify a transport protocol. UDP and TCP are the supported transport protocols.</p>
EIF Event	<p>Use this action to trigger an EIF event. You need to specify the EIF server and port.</p>

Configuring alerts for SNMP and EIF events

You can configure Log Analysis to send alerts as EIF events or SNMP traps to IBM Tivoli Netcool/OMNIBus.

Supported versions

You need to install IBM Tivoli Netcool/OMNIBus 8.1. Log Analysis does not include IBM Tivoli Netcool/OMNIBus or any of its components. You need to install IBM Tivoli Netcool/OMNIBus independently.

For SNMP traps, version 2 c is supported.

Prerequisites

Before you can configure this feature, you need to install IBM Tivoli Netcool/OMNIBus 8.1 on a server that the Log Analysis can connect to.

Before you start the configuration, you need to copy these three files to IBM Tivoli Netcool/OMNIBus:

- tivoli_eif_la.rules
- mttrapd_la.rules
- ioa-la.mib

Configuring IBM Tivoli Netcool/OMNIBus

You must configure IBM Tivoli Netcool/OMNIBus so that it can receive the EIF events or SNMP traps that Log Analysis sends it.

Procedure

1. Log in to IBM Tivoli Netcool/OMNIBus and select either EIF Events or SNMP Traps for the protocol.
2. If you want to use EIF Events, complete the following steps:
 - a. Install the IBM Tivoli Netcool/OMNIBus EIF Probe. For more information, see [IBM Tivoli Netcool/OMNIBus Probe for Tivoli EIF](#).
 - b. Copy the tivoli_eif_la.rules file to the IBM Tivoli Netcool/OMNIBus server.

- c. Configure the EIF Probe to send and receive EIF events from Log Analysis. The location for default probe rules and properties files in an OMNIBus installation is <OMNIBUS_HOME>/probes, in a platform specific sub directory. For example, for an x86 Linux installation, the location is <OMNIBUS_HOME>/probes/linux2x86:

- 1) Copy the default EIF props file, `tivoli_eif.props`, from the platform specific directory to a Log Analysis properties file, for example `tivoli_eif_la.props`.
- 2) Remove the comments and specify values for the following properties:

RulesFile

Specify the directory where you copied the rules file in step 2.b.

PortNumber

Specify the port that the EIF probe will use to listen to Log Analysis. This information is required to facilitate the alert actions in Log Analysis.

MessageLevel

This is optional. Set this property to debug' to log additional debugging information.

MessageLog

This is optional. You can specify a log file for the EIF probe. The default is <OMNIBUS_HOME>/log/tivoli_eif.log.

- 3) To start the EIF Probe, enter the following command:

```
<OMNIBUS_HOME>/probes/nco_p_tivoli_eif -propsfile <path>/tivoli_eif_la.props
```

3. If you want to use SNMP Traps, complete the following steps:

- a. Install the IBM Tivoli Netcool/OMNIBus SNMP Probe. For more information, see [IBM Tivoli Netcool/OMNIBus SNMP Probe](#).
- b. Copy the `mttrapd_la.rules` file to the IBM Tivoli Netcool/OMNIBus server.
- c. Configure the SNMP Probe to send and receive SNMP Traps from Log Analysis. The location for default probe rules and properties files in an OMNIBus installation is <OMNIBUS_HOME>/probes, in a platform specific sub directory. For example, for an x86 Linux installation, the location is <OMNIBUS_HOME>/probes/linux2x86:

- 1) Copy the default EIF props file, `mttrapd.props`, from the platform specific directory to a Log Analysis properties file, for example `mttrapd_la.props`.
- 2) Remove the comments and specify values for the following properties:

RulesFile

Specify the directory where you copied the rules file in step 3.b.

PortNumber

Specify the port that the SNMP Probe will use to listen to Log Analysis. This information is required to facilitate the alert actions in Log Analysis.

Protocol

Specify the protocol that the SNMP Probe uses. You can specify UDP or TCP. The default value is UDP.

MessageLevel

This is optional. Set this property to debug' to log additional debugging information.

MessageLog

This is optional. You can specify a log file for the SNMP Probe. The default is <OMNIBUS_HOME>/log/mttrapd.log.

- 3) To start the SNMP Probe, enter the following command:

```
<OMNIBUS_HOME>/probes/nco_p_mttrapd -propsfile <path>/mttrapd_la.props
```

What to do next

After you complete the configuration in IBM Tivoli Netcool/OMNIBus, you need to configure the alerts.

Configuring alerts for EIF Events and SNMP Traps

Before you can trigger alerts for EIF Events and SNMP Traps, you need to create the alert actions in Log Analysis,

About this task

You must install your chosen Probe and complete the required configuration in IBM Tivoli Netcool/OMNIBus before you create the alerts.

Procedure

1. Open the **Manage Alerts** UI and create an alert for EIF Events or SNMP Traps.
2. If you are using EIF Events, you need to specify the host name of the IBM Tivoli Netcool/OMNIBus server and the port number used by the EIF Probe.
3. If you are using SNMP Traps, you need to specify the following values:
 - Host name of the IBM Tivoli Netcool/OMNIBus server,
 - SNMP Probe port number
 - SNMP community string. The default is "public"
 - Transport protocol. The default is "udp"

Related tasks

[“Creating alerts” on page 73](#)

To trigger actions based on events, create an alert.

EIF Event structure

The EIF Event contains structured information that is sent to IBM Tivoli Netcool/OMNIBus.

The EIF Event is structured as follows:

```
"source": "IBM_Operational_Analytics_Log_Analysis"
  "origin": // hostname/ip-address of IBM Operations Analytics - Log Analysis server
  "event": "IOA_LA_<alert-type>_Alert"
  "alertName": // name of the alert
  "alertType": "KEYWORD_MATCH"/"KEYWORD_MATCH_THRESHOLD"/"KEYWORD_MATCH_DEDUP"
               "ANOMALY_MATCH"/"ANOMALY_MATCH_THRESHOLD"/"CO_OCCURRENCE"
  "description": // description of the alert
  "datasources": // list of associated datasources as a JSON array
  "timestamp": // epoch (long) timestamp for the alert
  "severity": 1/2/3/4 (critical/error/warning/info)
```

The EIF Event also contains specific fields for different alert types in the alertType parameter:

```
KEYWORD_MATCH
  "searchQuery": // query that triggered the alert
  "alertLogRecord": // log record that triggered the alert

KEYWORD_MATCH_THRESHOLD
  "windowStartTimestamp": // epoch (long) timestamp of the first log record that matched
                          // the search query
  "threshold": // the number of log records that matched the search query
  "windowDuration": // duration of the alert window in seconds, minutes or hours e.g.
3s/7m/2h

KEYWORD_MATCH_DEDUP
  "windowDuration": //duration of the de-duplication window in seconds, minutes or hours
e.g. 3s/7m/2h

ANOMALY_MATCH
  "fieldName": // name of the field for which a new value was observed
```

```

        "fieldValue": // new field value that was observed
        "alertLogRecord": // log record containing the new field value

    ANOMALY_MATCH_THRESHOLD
        "windowStartTimestamp": // epoch (long) timestamp of the first log record that
contained
                                // a new field value
        "threshold": // the number of new field value that were observed
        "windowDuration": // duration of the alert window in seconds, minutes or hours e.g.
3s/7m/2h

    CO_OCCURRENCE
        "windowStartTimestamp": // epoch (long) timestamp of the first detected event
        "windowDuration": // duration of the alert window in seconds, minutes or hours e.g.
3s/7m/2h

```

For example, this EIF Event is sent for KEYWORD_MATCH alert type and a severity 2 warning:

```

event: IOA_LA_KEYWORD_MATCH_Alert
source: IBM_Operational_Analytics_Log_Analysis
alertName: 'was-severity-error/was1/0'
alertType: 'KEYWORD_MATCH'
description: 'WAS error condition'
severity: 2
datasources: ['was1']
alertLogRecord='[09/1/15 13:27:23:964 GMT+05:30] 00000010 TraceResponse E DSRA1120E:
Application
did not explicitly close all handles to this Connection. Connection cannot be pooled.';
timestamp=1441094243964
origin='<ioala-server-fqdn>/<ioala-server-ip-address>'
searchQuery: 'severity:E'

```

SNMP Trap structure

The SNMP Trap contains structured information that is sent to IBM Tivoli Netcool/OMNIBus.

Log Analysis traps are assigned the 1.3.6.1.4.1.2.6.258 OID. All the SNMP traps that are sent from Log Analysis specifies an OID that this OID as a prefix. Every variable that is sent in a trap contains this OID as the prefix for the variable name.

The structure of the SNMP Trap is:

```

"1.3.6.1.6.3.1.1.4.1.0": // Alert OID
                        // "1.3.6.1.4.1.2.6.258.0.1" (KEYWORD_MATCH)
                        // "1.3.6.1.4.1.2.6.258.0.2" (KEYWORD_MATCH_THRESHOLD)
                        // "1.3.6.1.4.1.2.6.258.0.3" (KEYWORD_MATCH_DEDUP)
                        // "1.3.6.1.4.1.2.6.258.0.4" (ANOMALY_MATCH)
                        // "1.3.6.1.4.1.2.6.258.0.5" (ANOMALY_MATCH_THRESHOLD)
                        // "1.3.6.1.4.1.2.6.258.0.6" (CO_OCCURRENCE)
"1.3.6.1.4.1.2.6.258.1.1": // name of the alert
"1.3.6.1.4.1.2.6.258.1.2": // description of the alert
"1.3.6.1.4.1.2.6.258.1.3": 1/2/3/4 // alert severity (critical/error/warning/info)
"1.3.6.1.4.1.2.6.258.1.4": // list of associated datasources as a JSON array
"1.3.6.1.4.1.2.6.258.1.5": // epoch (long) timestamp for the alert

```

The SNMP Trap also contains specific fields for each alert type:

```

    KEYWORD_MATCH
        "1.3.6.1.4.1.2.6.258.1.6": // log record that triggered the alert
        "1.3.6.1.4.1.2.6.258.1.8": // query that triggered the alert

    KEYWORD_MATCH_THRESHOLD
        "1.3.6.1.4.1.2.6.258.1.11": // duration of the alert window in seconds, minutes or
hours e.g. 3s/7m/2h
        "1.3.6.1.4.1.2.6.258.1.12": // the number of log records that matched the search query
        "1.3.6.1.4.1.2.6.258.1.13": // epoch (long) timestamp of the first log record that
matched
                                // the search query

    KEYWORD_MATCH_DEDUP
        "1.3.6.1.4.1.2.6.258.1.11": //duration of the de-duplication window in seconds,
minutes or hours e.g. 3s/7m/2h

    ANOMALY_MATCH
        "1.3.6.1.4.1.2.6.258.1.6": // log record containing the new field value
        "1.3.6.1.4.1.2.6.258.1.9": // Name of the field for which a new value was observed
        "1.3.6.1.4.1.2.6.258.1.10": // New field value that was observed

```

```

    ANOMALY_MATCH_THRESHOLD
    "1.3.6.1.4.1.2.6.258.1.11": // duration of the alert window in seconds, minutes or
hours e.g. 3s/7m/2h
    "1.3.6.1.4.1.2.6.258.1.12": // the number of new field value that were observed
    "1.3.6.1.4.1.2.6.258.1.13": // epoch (long) timestamp of the first log record that
contained
                                // a new field value

    CO_OCCURRENCE
    "1.3.6.1.4.1.2.6.258.1.11": // duration of the alert window in seconds, minutes or
hours e.g. 3s/7m/2h
    "1.3.6.1.4.1.2.6.258.1.13": // epoch (long) timestamp of the first detected event

```

For example, the following SNMP event is sent for a **KEYWORD_MATCH** alert with a severity 2 error:

```

1.3.6.1.6.3.1.1.4.1.0: 1.3.6.1.4.1.2.6.258.0.1
1.3.6.1.4.1.2.6.258.1.1: was-severity-error/was1/0
1.3.6.1.4.1.2.6.258.1.2: WAS error condition
1.3.6.1.4.1.2.6.258.1.3 = 2
1.3.6.1.4.1.2.6.258.1.4 = [was1]
1.3.6.1.4.1.2.6.258.1.5 = 1441094243964
1.3.6.1.4.1.2.6.258.1.6 = [09/1/15 13:27:23:964 GMT+05:30] 00000010 TraceResponse E
DSRA1120E: Application
did not explicitly close all handles to this Connection. Connection cannot be pooled.
1.3.6.1.4.1.2.6.258.1.8 = severity:E

```

SNMP MIB

The complete MIB definition for alerts that are generated by Log Analysis are stored in the `ioa-la.mib` file.

The Log Analysis MIB module, `ioaLaMIB`, uses the 1.3.6.1.4.1.2.6.258 OID in the MIB hierarchy.

All the alert notifications and corresponding information from Log Analysis uses the 1.3.6.1.4.1.2.6.258 OID as a prefix.

The Log Analysis MIB contains six notification types, one for each alert type that is defined in Log Analysis. The notification types are:

ioaLaKeywordMatchAlert (OID 1.3.6.1.4.1.2.6.258.0.1)

This notification is generated when a keyword match occurs on a log record.

ioaLaKeywordMatchThresholdAlert (OID 1.3.6.1.4.1.2.6.258.0.2)

This notification is generated when a keyword match occurs on log records multiple times within a specified duration.

ioaLaKeywordMatchDedupAlert (OID 1.3.6.1.4.1.2.6.258.0.3)

This notification is generated the first time a keyword match occurs on a log record within a specific time window. Subsequent matches within the window are ignored.

ioaLaAnomalyMatchAlert (OID 1.3.6.1.4.1.2.6.258.0.4)

This notification is generated when a new field value is seen for a datasource within a specific time period.

ioaLaAnomalyMatchThresholdAlert (OID 1.3.6.1.4.1.2.6.258.0.5)

This notification is generated when multiple new field values are seen for a datasource within a specific time period.

ioaLaCoOccurrenceAlert (OID 1.3.6.1.4.1.2.6.258.0.6)

This notification is generated when multiple conditions are seen across one or more datasources within a specific time window.

Associated with each of these notification types, is a set of objects that provide detailed information about the notification. Some occur in all notifications, while others are specific to a notification type.

The objects in a Log Analysis notification are:

- `ioaLaAlertName` (OID 1.3.6.1.4.1.2.6.258.1.1)
- `ioaLaAlertDescription` (OID 1.3.6.1.4.1.2.6.258.1.2)
- `ioaLaAlertSeverity` (OID 1.3.6.1.4.1.2.6.258.1.3)
- `ioaLaAlertDatasources` (OID 1.3.6.1.4.1.2.6.258.1.4)

- ioaLaAlertTimestamp (OID 1.3.6.1.4.1.2.6.258.1.5)
- ioaLaAlertLogRecord (OID 1.3.6.1.4.1.2.6.258.1.6)
- ioaLaAlertLogRecordAnnotations (OID 1.3.6.1.4.1.2.6.258.1.7)
- ioaLaAlertSearchQuery (OID 1.3.6.1.4.1.2.6.258.1.8)
- ioaLaAlertFieldName (OID 1.3.6.1.4.1.2.6.258.1.9)
- ioaLaAlertFieldValue (OID 1.3.6.1.4.1.2.6.258.1.10)
- ioaLaAlertWindowDuration (OID 1.3.6.1.4.1.2.6.258.1.11)
- ioaLaAlertThreshold (OID 1.3.6.1.4.1.2.6.258.1.12)
- ioaLaAlertWindowStartTimestamp (OID 1.3.6.1.4.1.2.6.258.1.13)

Configuring email alert actions

Before you can use the email alert action, you need to specify the Simple Mail Transfer Protocol (SMTP) server host name in Log Analysis.

About this task

The email alert action feature is compatible with normal and secure SMTP servers.

Procedure

1. Open the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/configs/AlertActionEmailConfig.json file.
2. Specify the email properties. These properties are outlined in the following table.

Table 15. Email configuration properties		
Property name	Description	Value
secure	Decide whether you want to use secure SMTP or not.	Boolean
smtpMailServer	Specify the SMTP server's host name.	String
mailServerUser	User name that is used for authentication with the mail server.	String
mailServerPassword	Password for the user that is used for authentication with the mail server.	String
footer	Text that is added to the end of each email,	String
cc	Email addresses that are copied on each email. Recipients can view these addresses in the final mail.	String
bcc	Email addresses that are copied on each email. Recipients cannot view these addresses in the final mail.	String
attachLogRecordAnnotations	Decide whether to attach the log file record that triggered the alert.	Boolean

If you want to use encrypted passwords, you can use the `unity_securityUtility.sh` script to generate an encrypted password.

3. Save the file.

Example

Here is an example configuration:

```
{
  "smtpMailServer": "mail-server-host-name",
  "secure": true,
  "mailServerUser" : "example@example.com",
  "mailServerPassword" : "encrypted_pwd"
}
```

Related concepts

[“unity_securityUtility.sh command” on page 109](#)

You can use the `unity_securityUtility.sh` command to change the password that the Data Collector and EIF Receiver use to connect to IBM Operations Analytics - Log Analysis.

Example email alert

The email here is an example of one sent by the email alert action feature. The sender, receiver, subject prefix and body text that are specified in the alert action configuration.

```
Dear User,
Alert condition d1-severity was triggered at time 2015-01-09T16:45:00.000Z for the
datasource(s) d1.

The following log record caused the alert condition to trigger:[01/9/15 11:45:00:000 -0500]
00000010 TraceResponse E DSRA1120E: Application2 did not explicitly close all handles to
this connection. Connection cannot be pooled.

*** This is a system generated e-mail, please do not reply to this e-mail ***
```

Configuring auditability

You can use the auditing features of IBM Operations Analytics - Log Analysis to track activities to provide fact analysis and actively monitor user activities.

The IBM Operations Analytics - Log Analysis auditing feature is installed and enabled by default.

The audit data is stored in the `audit.log` file in the `<HOME>/IBM/LogAnalysis/logs` directory, and in the Indexing Engine index.

IBM Operations Analytics - Log Analysis supports auditing of the following processes:

- Access (login/logout)
- Authorization (user/role/permission management)
- Ingestion
- Search
- Alerts

Audit parameters

The auditing feature is enabled by default. The administrator can modify the default parameters after installation if required.

The `unitysetup.properties` file is in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF` folder. The administrator can edit values for the parameters in table 1, if required.

Table 16. Audit <code>unitysetup.properties</code>	
Parameters	Value
AUDIT_ACTIONS=	Specifies where the audit data is stored. The default value is LOG , INDEX. These values are the only supported values and are enabled by default.

Table 16. Audit unitysetup.properties (continued)	
Parameters	Value
AUDIT_INTERVAL=	Defines how frequently the audit data is written. The default value is 120000 milliseconds.

The audit data is written in JSON format to the <HOME>/IBM/LogAnalysis/logs/audit.log file. The audit file is a rolling file, supporting up to 20, 50-MB files by default.

The default file properties are in the log4j.properties file in <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/classes/ directory.

Note: If you change any of the audit parameters, you must save your changes and restart IBM Operations Analytics - Log Analysis.

Viewing the audit file

The audit file is a rolling file that can be accessed and searched at any time in the audit.log file or searched on the Indexing Engine console.

About this task

The audit feature of IBM Operations Analytics - Log Analysis is enabled by default, and records audit data at specified intervals. For information about the modifying the default parameters, see the *Audit parameters* topic in the *Configuring Auditing* section of the *Configuration* guide.

Procedure

To view the audit data, open the audit.log file in the <HOME>/IBM/LogAnalysis/logs directory.

The audit data is written in JSON format to the <HOME>/IBM/LogAnalysis/logs/audit.log file.

Configuring Expert Advice Custom Search Dashboard

You can use the Expert Advice Custom Search Dashboard to search a knowledge source for terms that occur in log entries.

You can use the default version of the Expert Advice Custom Search Dashboard. This version searches the IBM Support portal for the terms that are contained in selected columns. You can customize this version of the Expert Advice Custom Search Dashboard to suit your needs. See [“Customizing the default Expert Advice custom Custom Search Dashboard”](#) on page 86.

You can also implement your own version of the Expert Advice Custom Search Dashboard. You can also customize this version so that it searches a knowledge base other than the IBM support portal.

For more information, see [“Configuring a custom Expert Advice app”](#) on page 88.

Customizing the default Expert Advice custom Custom Search Dashboard

You can use the default implementation of the Expert Advice Custom Search Dashboard to search the IBM Support portal for the terms that are contained in selected columns. You can also configure the Expert Advice Custom Search Dashboard settings to suits your needs.

The Expert Advice Custom Search Dashboard that is used to provide expert advice is in the <HOME>/AppFramework/Apps/ directory where <HOME> is the directory in which IBM Operations Analytics - Log Analysis is installed. The Expert Advice Custom Search Dashboard is named IBMSupportPortal-ExpertAdvice.app.

Displaying more results

By default, the 10 most relevant results are displayed. To change this value, open the Expert Advice Custom Search Dashboard file and edit the number of displayed items. This setting is determined by the

value parameter. To edit this parameter, open the Expert Advice Custom Search Dashboard file and edit the value parameter:

```
{
  "name": "__MAX_RESULTS",
  "value": "10",
  "type": "String"
},
```

Note: Increasing this value might affect the performance of the Expert Advice Custom Search Dashboard as the additional data must be collected before it can be displayed.

Increasing the number of search terms

You can configure the number of unique terms that can be accepted for a search. By default, the number of unique terms is set to 7. To edit this parameter, open the Expert Advice Custom Search Dashboard file and edit the value parameter:

```
{
  "name": "__TERM_LIMIT",
  "value": "7",
  "type": "String"
},
```

Enhancing search strings

To ensure that a search return results, the Expert Advice Custom Search Dashboard removes content from the search term that is unlikely to return results. For example, a log message that contains the search string `unable to access jarfile /myMachine/foo/bar/foobar.jar` is not likely to return a specific match as the server path is likely to be specific to a user. To ensure better search results, the Expert Advice Custom Search Dashboard abbreviates this to `unable to access jarfile`.

The Expert Advice Custom Search Dashboard uses a set of criteria to amend the search string. The following values that are removed:

- URL
- File name
- File path
- IP address
- Number
- Punctuation

These values are specified as regular expression patterns in JavaScript Object Notation (JSON) format in the `GeneralizerPatterns_1.0.json` file. This file is the dictionary of pattern definitions. To add more patterns or to edit the existing patterns, edit this file.

The set of patterns that are applied to the input data are specified in the Expert Advice Custom Search Dashboard file. To apply the new pattern definitions that you created in the `GeneralizerPatterns_1.0.json` file, or to remove some of the patterns that are being applied, edit:

```
{
  "name": "__DEFAULT_GENERALIZATION",
  "value": ["URL", "FILENAME", "FILEPATH", "IP", "NUMBER", "PUNCTUATION"],
  "type": "StringArray"
},
```

The order in which the patterns are specified in the `__DEFAULT_GENERALIZATION` parameter is the order in which they are applied to the data. Ensure that the order is correct. For example, IP must be matched before the NUMBER value. Otherwise, the IP pattern is not matched.

Debugging

The Expert Advice Custom Search Dashboard generates a number of debug messages that provide information about the execution flow and Expert Advice Custom Search Dashboard status. These

messages are not displayed by default. To enable displaying these messages, set the following parameter to true:

```
{
  "name": "__DEBUG",
  "value": "false",
  "type": "String"
},
```

The Expert Advice Custom Search Dashboard supports two levels of debugging. Level 1 displays only the most important messages and Level 2 displays all messages. To set this level, edit the value parameter in this section of the file:

```
{
  "name": "__LOGLEVEL",
  "value": "2",
  "type": "String"
}
```

Configuring a custom Expert Advice app

To implement a custom version of the Expert Advice app that uses a knowledge source other than the IBM support portal, you must create a new searcher.

Procedure

1. Create a search wrapper code for your customized knowledge source. The wrapper code must contain implementations of the `Searcher`, `SearchPage`, and `Hit` interfaces. For details about each interface, see [“Searcher interface” on page 89](#), [“SearchPage interface” on page 89](#), and [“Hit interface” on page 90](#).
2. Save the search wrapper code as a Java Archive file.
3. Specify the path to this Java Archive file in the `__SEARCH_CONNECTOR_JAR` parameter in the Expert Advice app file. For example:

```
{
  "name": "__SEARCH_CONNECTOR_JAR",
  "value": "ExampleConnector.jar",
  "type": "String"
},
```

4. The search wrapper code must implement a searcher interface. Specify the class that implements the interface in `__SEARCH_CONNECTOR_CLASS` parameter in the Expert Advice app file. For example:

```
{
  "name": "__SEARCH_CONNECTOR_CLASS",
  "value": "com.foo.bar.connector.example.ExampleSearcher",
  "type": "String"
},
```

For more information about the `Searcher` interface, see [“Searcher interface” on page 89](#).

5. If you want to pass arguments to the search wrapper code, specify the arguments in the `__SEARCH_CONNECTOR_ARGS` parameter in the Expert Advice app file. For example:

```
{
  "name": "__SEARCH_CONNECTOR_ARGS",
  "value": ["foo", "bar"],
  "type": "String"
},
```

6. If your search wrapper requires more Java Archive files, you must add the paths for these files to the Java class path in the `ExpertAdvice.sh` file.

Searcher interface

The Searcher interface is the main interface that the search wrapper must implement.

The basic function of the wrapper as implemented by the Searcher interface can be summarized as follows:

1. Accepts a query.
2. Run the query against the specified knowledge source.
3. Return the query results.

The interface is defined as follows:

```
package com.ibm.tivoli.unity.loganalytics.knowledgeSource;
public interface Searcher {
    public void init(String[] args) throws Exception;
    public SearchPage issue(String query) throws Exception;
    public SearchPage issue(QueryLevel<String> query) throws Exception;
}
```

Methods

The interface contains the following methods:

init(String[] args)

Expert advice dynamically loads the Java Archive file that is specified in the app file. It then creates an instance of the class that implements the Searcher interface that is also specified in the app file. You must ensure that an empty constructor exists for this class. After the Expert Advice app creates an instance of the class, it calls the `init()` method of that instance along with the arguments that are specified in the app file. You can use the `init()` method to specify any initialization that requires external arguments.

issue(String query)

Use this method to run a simple query against the knowledge source. Billing error is an example of a simple input query. The output object must be a class that implements the SearchPage interface. For more information about the SearchPage interface, see [“SearchPage interface” on page 89](#)

issue(QueryLevel<String> query)

Use this method to run a structured query against the knowledge source. The QueryLevel structured query object encapsulates the query levels or order. The output object must be a class that implements the SearchPage interface.

SearchPage interface

The SearchPage interface represents the set of ordered results that are returned by the query from the knowledge source.

This interface is defined as:

```
package com.ibm.tivoli.unity.loganalytics.knowledgeSource;
public interface SearchPage {
    public String getUserQuery();
    public String getProcessingTime();
    public long getTotalResults();
    public long getStartIndex();
    public long getNumResults();
    public List<Hit> getHits();
}
```

Methods

This interface has the following methods:

getProcessingTime()

The processing time as reported by the search system. This method is not parsed into any numeric format and is for display purposes only.

getTotalResults()

The total number of available results as reported by the search system. It is only for display purposes.

getStartIndex()

The start index value for the first result in the search page. For example, if the query returns 100 results and the wrapper reads 10 pages at a time, the method returns a value of 0 and the page contains results 0 to 9. The next search page, the method returns a value of 10 and the page contains results 10 to 19.

getNumResults()

The number of results that are available for the search page.

getHits()

The ranked list of results for the search page. Each result entry must be a class that implements the `Hit` interface. For more information about the `Hit` interface, see [“Hit interface” on page 90](#).

Hit interface

The `Hit` interface represents a single result object.

This interface is defined as:

```
package com.foo.bar.example.loganalytics.knowledgeSource;

public interface Hit {
    public int getOrdinalNum();
    public String getTitle();
    public String getUrl();
    public String getSummary();
}
```

Methods

This interface contains the following methods:

getOrdinalNumber()

The ordinal number of this result that is used on the parent search page.

getTitle()

The title of the result page. The title is used as the anchor text when results are displayed on the Expert Advice search results page.

getUrl()

The URL of the result page. The url is used to generate the links that are displayed on the Expert Advice search results page.

Additional classes

When it interacts with the Expert Advice code, the wrapper code can refer to two classes, `QueryLevel` and `ItemSet`.

QueryLevel

The app generates a series of queries that are based on the set of columns that the user specifies. The precision of the queries varies and the queries are ordered to reflect the precision of the results. The order is such that a query at a higher-level yields more accurate results than a lower-level query. The `QueryLevel` class captures the ordering of the queries. Some queries can yield results with the same level of precision without yielding the same results. The `QueryLevel` class can contain more than one query for this reason.

This class is defined as:

```
package foo.bar.example.loganalytics.artifacts;  
public class QueryLevel<T> extends ArrayList<ItemSet<T>>
```

Each `ItemSet` object that is contained in a `QueryLevel` object represents a group of terms that must all occur on the same page. You must use an AND query for these terms.

If there is more than one `ItemSet` object in a `QueryLevel` object, you must use an OR query to separate the individual AND queries. For example:

```
(ItemSet_1_Term_1 AND ItemSet_1_Term_2) OR (ItemSet_2_Term_1 AND ItemSet_2_Term_2)
```

The syntax that specifies the AND and OR logic is specific to the knowledge source and is handled by the wrapper code.

ItemSet

The `ItemSet` class represents a single group of terms that must all occur in the same results page. You must use an AND query for this class.

This class is defined as:

```
package foo.bar.example.loganalytics.partialorder;  
public class ItemSet<T> extends HashSet<T>
```

Configuring launch in context

You can launch IBM Operations Analytics - Log Analysis in context from within an approved IBM product with the Search **UI** or Custom Search Dashboard.

Search UI launch-in-context

You can use the Search **UI** to launch IBM Operations Analytics - Log Analysis in context from within other products.

To start IBM Operations Analytics - Log Analysis in context using the Search **UI**, you must specify a URL in the following format:

```
https://<hostname>:<port>/Unity/SearchUI?queryString=<q>&timefilter=<t>  
&dataSources=<ds>
```

where:

hostname

The host name that corresponds to the data source.

port

The port that is used for communication with the IBM Operations Analytics - Log Analysis web application.

q

The value of the search string with a valid velocity query syntax.

t

A JSON format file filter to specify absolute or relative time.

For example, absolute time filters include "startTime": "24/206/2013 05:30:00" and "endTime": "25/06/2013 05:30:00". Relative time filters include "granularity": "Day" and "lastnum": "7".

ds

A JSON file format to specify single or group data sources to be queried.

In this example, the user uses the Search **UI** to launch IBM Operations Analytics - Log Analysis.

```
https://0.000.00.00:1111/Unity/SearchUI?queryString=severity==  
"Critical"&timefilter={"type":"relative","lastnum":"7","granularity": "Day"}  
&dataSources=[{"type": "datasource", "name": <omnibusEvents>}]
```

Custom Search Dashboard launch-in-context

You can launch IBM Operations Analytics - Log Analysis Custom Search Dashboards in context from within other products.

To start IBM Operations Analytics - Log Analysis in context, use the following URL:

```
https://<ip_address>:<port>:/Unity/CustomAppsUI?name=<name>&appParameters=<params>
```

where:

url

The URL format that you specify must be in the format:

```
https://<ip_address>:<port>:/Unity/CustomAppsUI?name=<name>&appParameters=<params>
```

ip_address

The IP address of the server on which IBM Operations Analytics - Log Analysis is installed.

port

The port that is used for communication with the IBM Operations Analytics - Log Analysis web application.

name

Specify the name of the application file. This is the name of the .app file that displays in the Custom Search Dashboards pane in the Search workspace.

params

(Optional) Specify a Custom Search Dashboard parameter JSON.

In this example, the user launches the *Day Trader App* Custom Search Dashboard in context without a custom app parameter.

```
https://0.000.00.00:1111/Unity/CustomAppsUI?name=Day%20Trader&20App
```

Configuring the DSV toolkit

The DSV toolkit is used to create Insight Packs that allow you to load Delimiter Separated Value (DSV) data into IBM Operations Analytics - Log Analysis. The DSV toolkit contains python scripts that take input from a properties file that describes a DSV log type and produces as output an installable Insight Pack.

IBM Operations Analytics - Log Analysis provides a semi-structured data analytics solution. Use IBM Operations Analytics - Log Analysis to identify problems and propose solutions by searching and indexing large volumes of unstructured data from a variety of sources. IBM Operations Analytics - Log Analysis allows you to reduce problem diagnosis and resolution time and more effectively manage your infrastructure and applications.

Before you can perform a search on log or other data, you must first load the data into IBM Operations Analytics - Log Analysis. An Insight Pack is a set of artifacts packaged together to allow IBM Operations Analytics - Log Analysis to ingest the data that is loaded. An Insight Pack contains a complete set of artifacts required to process a data source. You can install, uninstall, or upgrade an Insight Pack as a stand-alone package.

Each Insight Pack defines:

- The type of log data that is to be consumed.
- How data is annotated. The data is annotated to highlight relevant information.

- How the annotated data is indexed. The indexing process allows you to manipulate search results for better problem determination and diagnostics.
- Optionally, how to render data in an app chart or visualization.

What's new

The DSV toolkit was updated to improve ingestion performance and update property files.

- This DSVToolkit provides improved ingestion performance, with the ability to ingest large DSV records. The adjustment to the Java stack size for files with long log records is no longer required.
- Users can continue to use Insight Packs that were generated by older versions of the DSVToolkit. The improved performance of the updated DSVToolkit does not apply to these Insight Packs.
- The `aqlModuleName` property was renamed as `moduleName`. Generated property files that contain the `aqlModuleName` property continue to function correctly but a warning message, indicating that the `aqlModuleName` property is deprecated, is displayed.
- Use the newly added `quoteChar` property to specify the quotation character that you want to use to enclose fields that contain delimiters and line-breaks.

Note: If the `quoteChar` property is not specified, the double quotation mark (") is added by default during processing.

Property files that were created before the `quoteChar` property was added work as before as the double quotation mark was implicit in previous DSVToolkit versions.

- The `totalColumns` property is no longer required.

Create an Insight Pack using the DSV toolkit

This topic outlines how to install the DSV toolkit and use it to create an Insight Pack. If you are using IBM Operations Analytics - Log Analysis 1.2 or later, the DSV Toolkit is installed by default, and therefore does not need to be installed separately.

Before you begin

Copy the `DSVToolkit_v1.1.0.4.zip` from `<HOME>/SmartCloudAnalyticsLogAnalysisContent/tooling` to the `<HOME>/unity_content` directory and extract the files. The files are extracted to the `<HOME>/unity_content/DSVToolkit_v1.1.0.4` directory. Ensure that you, and any additional users that require the DSV toolkit, have write permissions to the `DSVToolkit_v1.1.0.4` directory.

Procedure

1. Run the `primeProps.py` script to create a new properties file, or to update an existing properties file.
2. Edit the properties file to meet the requirements of your DSV log file format. For information about the requirements of each section of the properties file, see the *Specifying properties for a log file type* section of this document.
3. Run the `devGen.py` script to generate, and where required deploy, your Insight Pack.

Specifying properties for a log file type

This topic outlines the properties that describe the contents of your DSV file. Each type of DSV file that you wish to ingest requires a properties file.

The properties file must conform to a specific format. For example, section headers must be enclosed in brackets and each section is made up of items that must be in the format `key: value` or `key=value`.

Note: When specifying items in a `key: value` format, a space is required after the colon (:).

For more information, see <http://docs.python.org/2/library/configparser.html>

SCALA_server

Specify the details for your IBM Operations Analytics - Log Analysis server in the `SCALA_server` section.

The following parameter must be defined:

scalaHome

Specify the path to the home directory of the IBM Operations Analytics - Log Analysis server.

DSV_file

The `DSV_file` section specifies parameters that apply to the entire log file and the entire Insight Pack.

DSV_file parameters**delimiter**

Specify the column separator that is used in the DSV log.

version

Specify the version number of the Insight Pack. The version must be a four digit number with a period separating each number. For example, 1.2.3.4.

moduleName

Specify the name of the Insight Pack and the underlying IOL - LA artifacts. The module name must:

- Start with a letter or an underscore (`_`). The letters can be upper or lowercase.
- Subsequent characters can be upper or lowercase letters, underscores, or digits (0-9).

quoteChar

Specify the quotation mark character that is used to enclose fields that contain delimiters and line-breaks.

Updates to the DSV_file parameters

- The `aqlModuleName` property was renamed as `moduleName`. Generated property files that contain the `aqlModuleName` property continue to function correctly but a warning message, indicating that the `aqlModuleName` property is deprecated, is displayed.
- Use the newly added `quoteChar` property to specify the quotation character that you want to use to enclose fields that contain delimiters and line-breaks.
- The `totalColumns` property is no longer required.

field*_indexConfig

Define values for index configuration in the `field*_indexConfig` section.

You can create a section in your properties file for each column of your DSV log file type. For each column that you want to add to the properties file, replace the `*` in `field*_indexConfig` with the column number in your DSV log file. For example, `field3_indexConfig` corresponds to column 3 of the DSV log file. Add fields as required. For more information, see the *Index Configuration* topics in the *Administering IBM Operations Analytics - Log Analysis* section of the Information Center.

name

Specify the name of the field in the index configuration. The field name is displayed in the Search workspace.

dataType

Specify the type of data contained in the log column. The valid values are TEXT, DATE, LONG, or DOUBLE.

dateFormat

This field is required if you have specified DATE as the `dataType` value. Specify the format of the timestamp used by the DSV log file. This format must conform with the Java 7 `SimpleDateFormat` class specification.

retrievable

(Optional) Specify the value for the `retrievable` field in the index configuration. The default value is TRUE.

retrieveByDefault

(Optional) Specify the value for the `retrieveByDefault` field in the index configuration. The default value is `TRUE`.

sortable

(Optional) Specify the value for the `sortable` field in the index configuration. The default value is `FALSE`. For the timestamp field, this value must be set to `TRUE`.

filterable

(Optional) Specify the value for the `filterable` field in the index configuration. The default value is `FALSE`. For the timestamp field, this value must be set to `TRUE`.

searchable

(Optional) Specify the value for the `searchable` field in the index configuration. The default value is `TRUE`.

path_*

(Optional) Specify the additional paths that are added to the list of paths in the index configuration. Replace the asterisk (*) with an integer. Start at 1 and increment by 1 for each additional path. One path is generated dynamically and placed at the end of the list.

A path is a JSON path that points to the data that is displayed in the Search workspace. The path must have the form:

```
path_1: annotations.aqlModuleName_viewName.fieldName
```

where the dynamically generated paths are created using the following substitutions:

aqlModuleName

The `aqlModuleName` in the properties file.

viewName

The name item in an `indexConfig` section with the word `Final` appended to it.

fieldName

The name in the `indexConfig` section. For an example of `fieldName`, see the *Excluding and combining columns* topic.

combine

This field is required if a path is specified. Specify the method used to merge the contents of multiple paths. The valid values are `FIRST` and `ALL`. The default value is `FIRST`. When the `combine` value is set to `FIRST`, paths that you have defined are checked before the dynamically generated path.

Defining a timestamp section

The properties file must contain a section that corresponds with the timestamp of the log record. The name item must be `timestamp`, the `dataType` must be `DATE`, a `dateFormat` must be specified, and `sortable`, `filterable`, and `searchable` must all be set to `true`.

For an example of a correctly completed timestamp section, see the *Example properties file* topic. For more information about supported date formats, see the *Supported formats* section.

field0_indexConfig

Define the index configuration for the whole log record.

The name, `dataType`, `path`, and `combine` values are shown in the *Example properties file* topic. These values must appear unchanged in every properties file. The field number in the section name must be 0.

Example properties file with edited index configuration fields

This is a sample properties file. All of the required values have been specified.

```
[SCALA_server]
scalaHome: $HOME/IBM/LogAnalysis

[DSV_file]
delimiter: ,
aqlModuleName: csv3Column
version: 1.0.0.0
totalColumns: 3
```

```

[field0_indexConfig]
name: logRecord
dataType: TEXT
retrievable: true
retrieveByDefault: true
sortable: false
filterable: false
searchable: true
path_1: content.text
combine: FIRST

[field1_indexConfig]
name: timestamp
retrievable: true
retrieveByDefault: true
sortable: true
filterable: true
searchable: true
dataType: DATE
dateFormat: yyyy-MM-dd'T'HH:mm:ss.SSSX

[field2_indexConfig]
name: severity
retrievable: true
retrieveByDefault: true
sortable: false
filterable: true
searchable: true
dataType: TEXT

[field3_indexConfig]
name: message
retrievable: true
retrieveByDefault: true
sortable: false
filterable: false
searchable: true
dataType: TEXT

```

This properties file creates an Index configuration that processes log records similar to this example:

```

2013-04-25T12:30:49.456-02:00, Warning,
Heap utilization patterns indicate that you may have a memory leak

```

Excluding and combining columns

This topic outlines how you can configure your properties file to exclude and combine columns from your DSV log file when it is displayed in IBM Operations Analytics - Log Analysis.

Excluding columns

If you do not want to display a column that is included in a DSV log file, do not specify that column when you add `indexConfig` sections to the properties file. For example, if you do not want to display columns 2, 3, and 4 of a 5 column log file, only specify the `field1_indexConfig` and `field5_indexConfig` property sections.

Combining columns

You can combine multiple columns from the DSV log file into one column in the Search workspace by specifying multiple paths in one `indexConfig` section. The section with multiple paths must be the one with the highest column number to ensure that the correct annotation is applied to the DSV log file. Fields that are part of the combined column, but are otherwise unimportant, can have all `true/false` index configuration fields set to `false` to ensure that data that is not required is not indexed.

The sample properties file combines the 2nd and 4th columns of the DSV log file into one column when it is displayed in the IBM Operations Analytics - Log Analysis Search workspace.

```

[field2_indexConfig]
name: shortMessage
dataType: TEXT
retrievable: false
retrieveByDefault: false

```



```

sortable: false
filterable: false
searchable: false

[field4_indexConfig]
name: longMessage
dataType: TEXT
retrievable: true
retrieveByDefault: true
sortable: false
filterable: false
searchable: true
path_1: shortMessage
combine: ALL

```

Generate a properties file

Use the `primeProps.py` to generate a template properties file. Default values are added where appropriate. Update the template properties file before running the `dsvGen` script.

Syntax

```
python primeProps.py pathToProps numNewSections [options]
```

Parameters

These parameters are defined for this script:

pathToProps

The path to the properties file that you want to create.

numNewSections

Specify the number of `indexConfig` sections that you want to add to the properties file. Each `indexConfig` section corresponds to a column in your DSV file.

Options

These additional options are defined for this script:

-o

Overwrites the existing properties file. The default value for this property is `false`.

-h

Displays the help screen for this script.

-f

Add this option and specify the path to a DSV file containing a header. The header is parsed and the name item for each generated section uses the header name instead of a default name.

After running this command, open and review the properties file to ensure that the name of each section complies with the requirements of AQL and the DSV specifications. For example, the timestamp must be lower case and the name cannot contain spaces.

Example `primeProps.py` output

This example displays the output generated by the `primeProps.py` script. It shows a template with default values for the `DSV_file`, `SCALA_server`, and `field0_indexConfig` sections. The command `python primeProps.py dsvProperties.properties 1` results in this output:

```

[SCALA_server]
scalaHome: $HOME/IBM/LogAnalysis

[DSV_file]
delimiter: ,
aqlModuleName: dsv1Column
version: 1.0.0.0

[field0_indexConfig]

```

```

name: logRecord
dataType: TEXT
retrievable: true
retrieveByDefault: true
sortable: false
filterable: false
searchable: true
path_1: content.text
combine: FIRST

[field1_indexConfig]
name: field1
dataType: TEXT
retrievable: true
retrieveByDefault: true
sortable: false
filterable: false
searchable: true

```

Next steps

To ensure that the properties file contains sufficient detail to generate an Insight Pack, complete these steps:

- Verify the default username and password.
- Verify the default delimiter.
- Verify that the `indexConfig` section of the properties file contains a field named `timestamp`. You must edit the relevant field name.
- Edit a section so that the name is `timestamp`, the `dataType` is set to `DATE`, the `dateFormat` value is appropriate, and that the `sortable` and `filterable` values are set to `TRUE`.
- Verify the default `scalaHome` location.

Generate an Insight Pack

Use the `dsvGen.py` script to generate and, where required deploy, an Insight Pack. You can also use the `pkg_mgmt.sh` command to install the Insight Pack. After the Insight Pack has been generated you can use the Log Analysis Insight Pack Tooling to make any additional changes that you require.

Syntax

The syntax required to run the `dsvGen.py` script is:

```
python dsvGen.py pathToProps [options]
```

Parameters

These parameters are defined for this script:

pathToProps

The path to the properties file that you want to use to generate your Insight Pack.

Options

These additional options are defined for this script:

-o

Overwrites the existing Insight Pack archive file. The default value for this property is `false`.

-h

Displays the help screen for this script.

-d

Deploy the Insight Pack using the `pkg_mgmt.sh` command. The `install` and `deploylfa` options are used. The default value for this property is `false`.

If you specify both the `-d` and `-o` options, any Insight Pack of the same name is removed, using the `pkg_mgmt.sh` `uninstall` option, before the new Insight Pack is installed.

-f

Applies the `-f` option to the `pkg_mgmt.sh` command. This eliminates all user prompts. The `-d` option must be used when using `-f`.

-l

Creates a Log Source for the generated Insight Pack. The Log Source hostname corresponds to the short hostname of the current server, and the Log Source log path points to a default file in the `<HOME>/logsources/` directory. The `-d`, `-u`, and `-p` options must be used when you use the `-l` option, even if the default `unityadmin` credentials exist.

-u

Specify a username to pass to `pkg_mgmt.sh`. The default `unityadmin` credentials will be used if nothing is supplied.

-p

Specify a password to pass to `pkg_mgmt.sh`. The default `unityadmin` credentials will be used if nothing is supplied.

Example

Executing this command results in the output described:

```
python dsvGen.py dsv5.properties -o -d
```

In this example:

- The `aqlModuleName` item is set to `dsv5Column`.
- In the `DSVToolkit_v1.1.0.4/build` directory, the `dsv5ColumnInsightPack_v1.0.0.0` directory is deleted.
- In the `DSVToolkit_v1.1.0.4/dist` directory, the archive `dsv5ColumnInsightPack_v1.0.0.0.zip` is deleted.
- The Insight Pack archive is created in the `DSVToolkit_v1.1.0.4/dist` directory and is named `dsv5ColumnInsightPack_v1.0.0.0.zip`.
- The Insight Pack archive is copied to the `<HOME>/unity_content/DSV` directory.
- The `pkg_mgmt.sh` command removes the old Insight Pack, if it exists, and re-installs it using the new archive.

Troubleshooting

There are several commonly encountered problems when using the DSV Toolkit. This is a description of the symptoms encountered and suggested solutions for resolving the problems.

The number of successfully ingested records is 0

Problem:

The number of successfully ingested records is 0, and there are no errors.

Resolution:

The number of columns in the DSV file does not match the number specified in the properties file. Verify that the properties file is correct and that none of the records in the DSV file have an abnormal number of fields.

The dsvGen.py script is displaying an error

Problem:

The `dsvGen.py` script is displaying an error like "The timestamp field must be sortable and filterable" or "A field of type DATE was specified, but no dateFormat item was provided".

Resolution:

The timestamp field you define has a specific set of requirements. See the [“Defining a timestamp section”](#) on page 95 for the detailed requirement.

Supported formats

This section outlines the formats that are supported for DSV log files.

Timestamp formats

The timestamp format from the DSV log file is based on the timestamp formats supported by the Java 7 `SimpleDateFormat` class. Any date format not supported by the `SimpleDateFormat` class cannot be processed from a DSV log file.

If a time-only timestamp is contained in the log, `SimpleDateFormat` assigns a default date of Jan 1, 1970 to the timestamp. Any double quotes that surround a timestamp are removed. Other characters must be included as literals in the `dateFormat` item. For example, a timestamp surrounded by brackets must be specified in the format:

```
dateFormat '['yyyy-MM-dd HH:mm:ss']'
```

For more information on valid timestamp formats for `SimpleDateFormat`, see <http://docs.oracle.com/javase/7/docs/api/java/text/SimpleDateFormat.html>.

Delimiter formats

The follow delimiters are supported:

- comma (,)
- colon (:)
- semicolon (;)
- pipe (|)
- dash (-)
- slash (/)
- backslash (\)
- tab (\t)
- tilde (~)

Quotation characters

The follow quotation characters are supported:

- Double quotation mark (")
- Single quotation mark (')

Note: If the `quoteChar` property is not specified, the double quotation mark (") is added by default during processing.

Property files that were created before the `quoteChar` property was added work as before as the double quotation mark was implicit in previous DSVToolkit versions.

DSV formats

The Insight Packs generated by the DSV toolkit support DSV log file types that meet these requirements. In each case, a comma is used as a delimiter:

- Each log record is on a separate line. A line break is used to delimit the log records. CRLF denotes a line break. For example:

```
aaa,bbb,ccc CRLF  
zzz,yyy,xxx CRLF
```

- The last record in the file might, or might not, end with a line break. For example:

```
aaa,bbb,ccc CRLF  
zzz,yyy,xxx
```

- A header line might be present as first line of the file. This header line has same format as all standard record lines and contains names that correspond to the fields in the file. The header line also contains the same number of fields as the records in the rest of the file. For example:

```
field_name,field_name,field_name CRLF
aaa,bbb,ccc CRLF
zzz,yyy,xxx CRLF
```

- Within the header and each record, there might be one or more fields that are separated by delimiters. Each line contains the same number of fields. Spaces are considered as part of a field and are ignored. The last field in the record must not be followed by a delimiter. For example:

```
aaa,bbb,ccc
```

- Each field might or might not be enclosed in quotation characters. If fields are not enclosed in quotation characters, quotation characters might not appear inside the fields. The following examples show fields that are enclosed by single and double quotation characters:

```
"aaa","bbb","ccc" CRLF
'zzz','yyy','xxx'
```

- Fields containing quotation characters, delimiters, and line breaks must be enclosed in quotation characters. The following examples show fields that are enclosed by single and double quotation characters:

```
"aaa","b,bb","ccc" CRLF
'zzz','y,yy','xxx'
```

- If quotation characters are used to enclose fields, a quotation character that appears inside a field must be escaped by preceding it with another quotation character. The following examples show single and double quotation characters:

```
"aaa","b""bb","ccc"
'zzz','y''yy','xxx'
```

Configuring aliases

You can use the IBM Operations Analytics - Log Analysis alias feature to specify an alias.

The IBM Operations Analytics - Log Analysis alias feature is installed and enabled by default.

The alias data is stored in the `aliasSource.json` file in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF` directory.

Configuring an alias

You can use the IBM Operations Analytics - Log Analysis alias feature to specify an alias, consisting of an alternative name.

About this task

Aliases are displayed in the IBM Operations Analytics - Log Analysis UI.

Procedure

1. Open the `aliasSource.json` file in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF` directory.
2. To apply an alias to a fieldname, edit the fieldname parameters in the `aliasSource.json` file. The following example shows the parameters to apply an alias to a hostname:

```
{
  "aliasKeyValueArray": [
    {
      "sourceType": "WASSystemOut",
```

```

"fields": [
  {
    "fieldName": "hostname",
    "alias_field": "severity_alias",
    "translations": {
      "E": "Error",
      "W": "Warning",
      "O": "Overflow",
      "I": "Info"
    }
  }
]
}

```

Where:

- **sourceType** specifies the source type for which the alias is created.
- **fieldName** specifies the field name of the source type.
- **alias_field** specifies the alias name of the field.
- **translations** contains the field values and aliases.

What to do next

Verify that the aliases are correctly configured in the IBM Operations Analytics - Log Analysis UI.

Errors that are recorded during configuration are stored in the `UnityApplication.log` file in the `<HOME>/IBM/LogAnalysis/logs` directory.

Configuration reference

Read reference information about the scripts and properties, which you can configure after you install Log Analysis.

ldapRegistryHelper.properties

You can edit the `ldapRegistryHelper.properties` to specify LDAP server details.

The following properties are required and define the connection information for the target LDAP server.

Table 17. LDAP server connection information properties	
Property	Description
ldap_hostname_property=	The LDAP hostname.
ldap_port_property=	The LDAP port.
ldap_baseDN_property=	The LDAP baseDN. For example, "dc=com" for TDS users, and "CN=Users,DC=sflab,DC=local" for AD users.

The following properties are optional and define the connection information for the target LDAP server. Where applicable, default settings are assigned.

The **bindPassword** value for AD users is encrypted in the `ldapRegistryHelper_config.xml`.

Table 18. Optional LDAP server connection information properties	
Property	Description
ldap_bindDN_property=	The LDAP bindDN. For example, "CN=Administrator,CN=Users,DC=sflab,DC=local" for AD users.

Table 18. Optional LDAP server connection information properties (continued)	
Property	Description
<code>ldap_bindPassword_property=</code>	The LDAP bind password.
<code>ldap_realm_property=</code>	The LDAP realm. The default value is <code>LdapRegistryRealm</code> .
<code>ldap_id_property=</code>	The LDAP ID. The default value is <code>LdapRegistryId</code> .
<code>ldap_ignoreCase_property=</code>	The LDAP ignore case. The default value is <code>true</code> .
Fix Pack 1 <code>recursiveSearch=</code>	Fix Pack 1 Enable recursive search. The default value is <code>true</code> . This is only available in IBM Operations Analytics - Log Analysis 1.3.3 Fix Pack 1.

ldapRegistryHelper.sh command

You can use the `ldapRegistryHelper.sh` command to enable a basic connection for user authentication in IBM Operations Analytics - Log Analysis.

For more information about how to use the command to set up LDAP authentication with IBM Tivoli Directory Server or Microsoft Active Directory, see [“Configuring LDAP authentication with the ldapRegistryhelper.sh script” on page 36](#).

Supported integrations

This command currently supports connections to the IBM Tivoli Directory Server and Microsoft Active Directory.

Prerequisites

Before you use this command, you must update the `ldapRegistryHelper.properties` file in the `<HOME>/IBM/LogAnalysis/utilities/` directory with the connection and configuration information for the target LDAP server.

Syntax

The `ldapRegistryHelper.sh` command is in the `<HOME>/IBM/LogAnalysis/utilities` directory and it has the following syntax:

```
ldapRegistryHelper.sh    config | enable
```



Warning:

To run the script, the `JAVA_HOME` variable must be set correctly for IBM Operations Analytics - Log Analysis. If the script fails, run the following command to set the `JAVA_HOME` variable:

```
JAVA_HOME=${<HOME>}/IBM-java
```

Parameters

The `ldapRegistryHelper.sh` command has the following parameters:

config

Use the `config` parameter to create an XML file that is called `ldapRegistry.xml` in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity` directory. This file uses the connection and configuration information that is defined in the `ldapRegistryHelper.properties` file.

enable

Use the enable parameter to enable LDAP authentication that uses the information that is specified in the `ldapRegistry.xml` file. This parameter also disables the reference to the database-managed custom user registry.

unity command

Use the unity command to start, stop, and restart IBM Operations Analytics - Log Analysis. Also use this command to display the status of processes and to display version information for IBM Operations Analytics - Log Analysis.

Syntax

The unity command is located in the `<HOME>/IBM/LogAnalysis/utilities` directory and has the following syntax:

```
unity.sh -start | -stop | -version | -restart | -status
```

Parameters

The parameters for this command are:

- start

Use this parameter to start IBM Operations Analytics - Log Analysis and associated services.

-stop

Use this parameter to stop IBM Operations Analytics - Log Analysis and associated services.

-version

Use this parameter to determine the currently installed version of IBM Operations Analytics - Log Analysis.

-restart

Use this parameter to restart IBM Operations Analytics - Log Analysis and other associated services.

-status

Use this parameter to display a list of IBM Operations Analytics - Log Analysis processes, including the Indexing Engine nodes, and their status.

LFA configuration file parameters

The IBM Tivoli Monitoring Log File Agent uses the information that is specified in the configuration file to process log file information.

Table 1 explains that parameters that you can modify in this file.

Table 19. Parameters for LFA configuration file		
Required for	Parameter	Description
All	LogSources	Specify the data source that you want to monitor. If you are specifying multiple data sources, they must be comma-separated and without spaces. When you configure a remote directory in the LFA configuration file, the directory you specify must not contain any subdirectories.

Table 19. Parameters for LFA configuration file (continued)

Required for	Parameter	Description
Internal LFAs installed on remote servers	ServerLocation,	Specify the server location for the EIF receiver server. For example, for a server that is at 111.222.333.444, specify the following value: ServerLocation=111.222.333.444
Internal LFAs installed on remote servers	ServerPort	Specify the port that the EIF receiver uses. For example: ServerPort=5529
Internal LFAs installed on remote servers	BufEvtMaxSize	Specify the maximum buffer size for the LFA. This parameter is the maximum size that the cache is allowed to be. If the cache is full, events are dropped and performance can decline. The value that you enter here is in kilobytes. For example: BufEvtMaxSize=102400
External LFAs installed on remote servers	SshAuthType	You must set this value to either PASSWORD or PUBLICKEY. If you set this value to PASSWORD, IBM Operations Analytics - Log Analysis uses the value that is entered for SshPassword as the password for Secure Shell (SSH) authentication with all remote systems. If you set this value to PUBLICKEY, IBM Operations Analytics - Log Analysis uses the value that is entered for SshPassword as pass phrase that controls access to the private key file.

Table 19. Parameters for LFA configuration file (continued)

Required for	Parameter	Description
External LFAs installed on remote servers	SshHostList	<p>You use the SshHostList value to specify the hosts where the remotely monitored log files are generated. IBM Operations Analytics - Log Analysis monitors all the log files that are specified in the LogSources or RegexLogSources statements in each remote system.</p> <p>If you specify the local machine as a value for this parameter, the LFA monitors the files directly on the local system. If you specify that the localhost SSH is not used to access the files on the system, IBM Operations Analytics - Log Analysis reads the files directly.</p>
External LFAs installed on remote servers	SshPassword	<p>If the value of the SshAuthType parameter is PASSWORD, enter the account password for the user that is specified in the SshUserid parameter as the value for the SshPassword parameter.</p> <p>If the value of the SshAuthType parameter is PUBLICKEY, enter the pass phrase that decrypts the private key that is specified in the SshPrivKeyfile parameter.</p>
External LFAs installed on remote servers	SshPort	<p>You specify the TCP port that is used for SSH connections. If you do not enter anything, this value is defaulted to 22.</p>
External LFAs installed on remote servers	SshPrivKeyfile	<p>If the value of the SshAuthType parameter is set to PUBLICKEY, enter the directory path to the file that contains the private key of the user that is specified in the SshUserid parameter as the value for this parameter.</p> <p>If the value of the SshAuthType parameter is not set to PUBLICKEY, this value is not required.</p>

Table 19. Parameters for LFA configuration file (continued)

Required for	Parameter	Description
External LFAs installed on remote servers	SshPubKeyfile	If the value of the SshAuthType parameter is set to PUBLICKEY, enter the directory path to the file that contains the public key of the user that is specified in the SshUserid parameter as the value for this parameter. If the value of the SshAuthType parameter is not set to PUBLICKEY, this value is not required.
External LFAs installed on remote servers	SshUserid	Enter the user name from the remote system that the agent uses for SSH authentication.

eifutil.sh command

To administer EIF Receiver instances, use the `eifutil.sh` command.

Syntax

The `eifutil.sh` command has the following syntax and is in the `<USER_HOME_REMOTE>/DataForwarders/EIFReceivers/utilities` where `<USER_HOME_REMOTE>` is the directory on the remote host where the EIF Receiver instances are deployed:

```
eifutil.sh -status|-start <Inst_ID>|-stop <Inst_ID>|-startAll|-stopAll|-restart
<Inst_ID>|-restartAll
```

where `<Inst_ID>` is the ID for the specific EIF instance.

Parameters

-status

Displays the status for the installed instances. For example:

```
=====
COMPONENT      Instance      PID          PORT          STATUS
=====
EIF Receiver    eif_inst_1    13983        6601          UP
EIF Receiver    eif_inst_2    14475        6602          UP
EIF Receiver    eif_inst_3    14982        6603          UP
EIF Receiver    eif_inst_4    15474        6604          UP
EIF Receiver    eif_inst_5    15966        6605          UP
=====
```

-start <Inst_id>

Starts the specified instance.

-stop <Inst_id>

Stops the specified instance.

-startAll

Starts all instances.

-stopAll

Stops all instances.

-restart<Inst_id>

Restarts the specified instance.

-restartAll

Restarts all the instances.

lfautil.sh command

To administer IBM Tivoli Monitoring Log File Agent (LFA) instances, use the `lfautil.sh` command.

Syntax

The `lfautil.sh` command has the following syntax and is in the `<USER_HOME_REMOTE>/utilities/` directory on the remote host where `<USER_HOME_REMOTE>` is the directory on the remote host where the LFA instances are deployed:

```
lfautil.sh -start|-stop|-status|-restart
```

Parameters

-start

Starts all the LFA instances on the remote host.

-stop

Stops all the LFA instances on the remote host.

-status

Displays the status for the LFA instances on the remote host. For example:

```
=====
COMPONENT          PID          STATUS
=====
Log File Agent     23995         UP
=====
```

-restart

Restarts the LFA instances on the remote host.

Data Collector properties

Before you can use the data collector to stream data or load a batch of historic data, edit the `javaDatacollector.props` file.

The `javaDatacollector.props` file is in the `<HOME>/IBM/LogAnalysis/utilitiesdatacollector-client` folder.

The `logFile`, `hostname`, `logpath`, and `keystore` parameters are required.

The `userid`, `password`, and `keystore` parameters are automatically populated with the default values that are created during the installation. If you want, you can change these but you do not need to.

Table 20. Data Collector properties	
Parameter	Value
<code>logFile</code>	The full path of the file you want to load.
<code>servletURL</code>	The URL of the Data Collector service.
<code>userid</code>	The user ID for the Data Collector service.
<code>password</code>	The password for the Data Collector service.
<code>datasource</code>	The datasource that you want to use to load data.
<code>timestamp</code>	The time stamp to use if a time stamp is not found in the log file.
<code>batchsize</code>	The number of BYTES of logs sent in one batch. The default value is 500,000.

Table 20. Data Collector properties (continued)	
Parameter	Value
keystore	The full path to the keystore file.
inputType	The valid input type is LOGS.
flush flag	If the default <code>true</code> is set, the client sends a flush signal to the Generic Receiver for the last batch of the file. If set to <code>false</code> no flush signal is sent when the end-of-file is reached.

unity_securityUtility.sh command

You can use the `unity_securityUtility.sh` command to change the password that the Data Collector and EIF Receiver use to connect to IBM Operations Analytics - Log Analysis.

Syntax

The `unity_securityUtility.sh` command is in the `<HOME>/IBM/LogAnalysis/utilities` directory and it has the following syntax:

```
unity_securityUtility.sh encode [textToEncode] [unity.ks]
```

Parameters

The `unity_securityUtility.sh` command has the following parameters:

encode

The `encode` action returns an AES encrypted version of the text that you enter as the text to encrypt.

[*textToEncode*]

Use the [*textToEncode*] parameter to enter the password that you want to encrypt. If you do not specify a password for this parameter, IBM Operations Analytics - Log Analysis prompts you for one.

[unity.ks]

The `unity.ks` file is the default keystore that is generated automatically during installation. It controls how the password is encrypted and decrypted.

The `unity.ks` file is used to encrypt and decrypt passwords for the following features:

- Java data collector client in the `<HOME>/IBM/LogAnalysis/utilities/datacollector-client/javaDatacollector.properties` file.
- EIF Receiver in the `<HOME>/IBM/LogAnalysis/utilities/UnityEIFReceiver/config/unity.conf` file.

For an example of how to use this command, see [“Changing the default EIF Receiver or Data Collector password”](#) on page 237.

securityUtility.sh utility

You can use the `securityUtility.sh` utility to encrypt the passwords that you use in your Lightweight Directory Access Protocol (LDAP) configuration.

Syntax

The command is in the `<HOME>/wlp/bin/` directory and it has the following syntax:

```
securityUtility {encode}
```

Parameters

This command has the following parameter:

encode

Use the encode action to return an XOR encrypted version of the text that you enter.

Example

To encrypt a password, enter the following command:

```
./securityUtility encode myPassword
```

The encrypted password is displayed as the {xor} value:

```
{xor}MiYPPiwsKG8t0w==
```

eif.conf file

The file `eif.conf` is a configuration file for the TEC Adapter used by the `scala_custom_eif` plugin to send log records as events to the IBM Operations Analytics - Log Analysis EIF Receiver.

The file `eif.conf` is found in the `logstash/outputs` directory relative to where the logstash Integration Toolkit was installed on the logstash server. The logstash Integration Toolkit installation configures `eif.conf` during installation with the server location, server port, cache file location, and the log file. You can modify any other properties to customize your installation. You must restart the logstash agent in order effect changes to this configuration file.

Note: Please refer to the comments in the `eif.conf` file for the latest details on the available parameters and their descriptions.

eif.conf file parameters

BufEvtMaxSize=<kilobytes>

Specifies the maximum size, in kilobytes, of the adapter cache file. The default value is 64. The cache file stores events on disk when the `BufferEvents` keyword is set to YES. The minimum size for the file is 8 KB. File sizes specified below this level are ignored, and 8 KB is used. There is no upper limit for the file size.

Note: If the cache file already exists, you must delete the file for parameter changes to take effect.

The `BufEvtMaxSize` parameter is optional.

BufEvtPath=<pathname>

Specifies the full path name of the adapter cache file. This is a required parameter when the `BufferEvents` value is set to YES.

BufferEvents=YES | MEMORY_ONLY | NO

Specifies how event buffering is enabled.

- **YES** - Stores events in the file specified by the `BufEvtPath` keyword.
- **MEMORY_ONLY** - Buffers events in memory.
- **NO** - Does not store or buffer events.

The value is not case-sensitive. The default value is YES. This parameter is optional.

ConnectionMode=connection_oriented | connection_less

Specifies the connection mode to use to connect to the IBM Operations Analytics - Log Analysis EIF Receiver. The default value is `connection_less`.

- **connection_oriented** - A connection is established at adapter initialization and is maintained for all events sent. A new connection is established only if the initial connection is lost. The connection is discarded when the adapter is stopped. This option can be abbreviated to `co` or `CO`.
- **connection_less** - A new connection is established and discarded for each event or group of events that is sent.

This parameter is optional.

LogFileName=<pathname>

Specifies the full path name of the log file for the adapter.

LogLevel=<level>

Specifies whether the Java API generates log messages or not. By default, no messages are generated. Specify ALL to generate messages. If you specify any other value or no value, the API does not generate messages. This parameter is optional.

ServerLocation=<host>

Specifies the name of the host where the IBM Operations Analytics - Log Analysis EIF Receiver resides.

ServerPort=number Specifies the port number on which the IBM Operations Analytics - Log Analysis EIF Receiver listens for events.

FQDomain=YES | NO | <fully.qualified.domain.suffix>

Specifies the fqhostname slot.

- **YES** - The adapter will attempt to determine the fully qualified hostname and if successful will fill in the fqhostname slot in the event.
- **NO** - The fqhostname slot will be set to a null string.
- **<fully.qualified.domain.suffix>** - The adapter will append this value to the hostname in order to set the fqhostname slot.

unity.conf file

The `unity.conf` is a configuration file for the IBM Operations Analytics - Log Analysis EIF Receiver.

The `unity.conf` file is found in the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/` directory.

unity.conf file parameters

Table 21. unity.conf parameters	
Parameter	Description
<code>unity.data.collector.ip=host name</code>	The host name of the server on which IBM Operations Analytics - Log Analysis is installed
<code>unity.data.collector.port=9987</code>	The port that is specified during installation. The default value is 9987.
<code>unity.data.collector.protocol=https</code>	The communication protocol. The default value is https.
<code>unity.data.collector.uri=/Unity/DataCollector</code>	The uri used in the REST invocation.
<code>unity.data.collector.userid=unityadmin</code>	The user ID of the user assigned the UnityUser role. The default user ID is unityAdmin.
<code>unity.data.collector.password={aes}<Unique_string_of_alphanumeric_characters></code>	The password that is associated with the UnityAdmin role. The default value is {aes}<Unique_string_of_alphanumeric_characters>.
<code>unity.data.collector.keystore=/home/unity/IBM/LogAnalysis/wlp/usr/server/Unity/keystore/unity.ks</code>	The full path to the keystore file.
<code>unity.data.collector.eif.consumer.num.events=1000000</code>	The common queue for all of the EIF events. The default value is 1000000.

Table 21. <i>unity.conf</i> parameters (continued)	
Parameter	Description
<code>unity.data.collector.event.service.num.events=80000</code>	Each datasource has 1 service queue. Events are buffered in this queue and placed in batches. The batches are passed to the poster queue. The default value is 80000.
<code>unity.data.collector.event.poster.num.events=500</code>	Each datasource has 1 poster queue. Batches are selected and posted to the IBM Operations Analytics - Log Analysis server from this queue. The default value is 500.
<code>unity.data.collector.gc.interval=2</code>	Determine the EIF Receiver memory cleanup interval in minutes. The default value is 2.
<code>logsource.buffer.wait.timeout=10</code>	Determine the buffer timeout in seconds. The default value is 10.
<code>logsource.max.buffer.size=450000</code>	Determine the buffer size in Bytes. The default value is 450000.

install.sh command

Use the `install.sh` command to install IBM Operations Analytics - Log Analysis or configure data collection for scalability on multiple remote nodes. Tivoli Event Integration Facility Receiver or the IBM Tivoli Monitoring Log File Agent on a remote server.

The `install.sh` command is in the `<HOME>/IBM/LogAnalysis/remote_install_tool/` directory on the local installation of IBM Operations Analytics - Log Analysis.

install.sh command parameters

To install IBM Operations Analytics - Log Analysis with IBM Installation Manager, run the command:

```
./install.sh
```

This command installs IBM Operations Analytics - Log Analysis and installs or upgrades, IBM Installation Manager if no other version is installed. For more information, see [“Installing with the IBM Installation Manager UI”](#) on page 15.

To install IBM Operations Analytics - Log Analysis with the console, run the command:

```
./install.sh -c
```

This command installs IBM Operations Analytics - Log Analysis and installs or upgrades IBM Installation Manager, if no other version of IBM Installation Manager is installed. For more information, see [“Installing with the IBM Installation Manager command-line interface”](#) on page 17.

To silently install IBM Operations Analytics - Log Analysis, run the command:

```
./install.sh -s <HOME_DIR>/smcl_silent_install.xml
```

where `<HOME_DIR>` is your home directory. This command silently installs IBM Operations Analytics - Log Analysis and installs or upgrades IBM Installation Manager Version 1.8.2. For more information, see [“Silently installing Log Analysis”](#) on page 18.

To install the Tivoli Event Integration Facility Receiver or the IBM Tivoli Monitoring Log File Agent on a remote server, run the command:

```
./install.sh
```

For more information, see [“Deploying the LFA or EIF on remote servers”](#) on page 215.

ssh-config.properties

Before you can use the remote installer utility to install instances of Apache Solr, you must configure the Secure Shell (SSH) for the remote hosts.

The `ssh-config.properties` file is in the `<HOME>/IBM/LogAnalysis/remote_install_tool/config` directory.

Table 22. <i>ssh-config parameters</i>	
Parameter	Value
REMOTE_HOST=	<code><remote_host></code>
PORT=	<code><port></code>
TIME_OUT=	<code><timeout_value></code> . The default value is 60000.
USER=	<code><remote_user></code>
PASSWORD=	<code><password></code>
USE_PASSWORD_LESS_SSH=	<code><true_or_false></code> . The default value is <code>true</code> . This setting enables password SSH authentication.
PATH_OF_PASSWORD_LESS_SSH_KEY=	<code><path_to_ssh_key_file></code> . Enter the path to the key file. For example <code>/home/pass/.ssh/id_rsa</code> .
PASSPHRASE_OF_PASSWORD_LESS_SSH_KEY=	<code><passphrase></code> . The passphrase used for SSH authentication.

Audit parameters

The auditing feature is enabled by default. The administrator can modify the default parameters after installation if required.

The `unitysetup.properties` file is in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF` folder. The administrator can edit values for the parameters in table 1, if required.

Table 23. <i>Audit unitysetup.properties</i>	
Parameters	Value
AUDIT_ACTIONS=	Specifies where the audit data is stored. The default value is <code>LOG , INDEX</code> . These values are the only supported values and are enabled by default.
AUDIT_INTERVAL=	Defines how frequently the audit data is written. The default value is 120000 milliseconds.

The audit data is written in JSON format to the `<HOME>/IBM/LogAnalysis/logs/audit.log` file. The audit file is a rolling file, supporting up to 20, 50-MB files by default.

The default file properties are in the `log4j.properties` file in `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/classes/` directory.

Note: If you change any of the audit parameters, you must save your changes and restart IBM Operations Analytics - Log Analysis.

Version utility

Use the `unity_VersionInfoUtility.sh` script to check the version information for your installation of Log Analysis.

Syntax

The `unity_VersionInfoUtility.sh` script is in the `<HOME>/IBM/LogAnalysis/utilities` folder. To run it, enter the following command:

```
<HOME>/IBM/LogAnalysis/utilities/unity_VersionInfoUtility.sh
```

Version information

The command returns version information. For example:

```
=====
UNITY Version Utility
=====
Gathered:    Tue Jul 28 07:00:54 EDT 2015

Java:        1.8.0 [IBM Corporation]
              /home/unityadmin/IBM/LogAnalysis/ibm-java/jre

Java VM:      IBM J9 VM [j9jit28]

Platform:     Linux
              2.6.32-279.el6.x86_64
              amd64

UNITY_HOME:   /home/unityadmin/IBM/LogAnalysis

Machine:      nc9118042236.in.ibm.com [9.118.42.236]

=====
Environment
=====
_                 = /home/unityadmin/IBM/LogAnalysis/ibm-java/jre/bin/java
CVS_RSH           = ssh
G_BROKEN_FILENAMES = 1
HISTCONTROL       = ignoredups
HISTSIZE          = 1000
HOME              = /home/unityadmin
HOSTNAME          = nc9118042236
IBM_JAVA_COMMAND_LINE = /home/unityadmin/IBM/LogAnalysis/ibm-java/jre/
bin/java -Dcom.ibm.tivoli.unity.LFA_DIR=/home/unityadmin/IBM/LogAnalysis/
IBM-LFA-6.30 -Dcom.ibm.tivoli.unity.UNITY_HOME=/home/unityadmin/IBM/
LogAnalysis -Dcom.ibm.tivoli.unity.DERBY_INSTALL_DIR=/home/unityadmin/
IBM/LogAnalysis/database -classpath /home/unityadmin/IBM/LogAnalysis/
utilities/lib/verutil.jar:/home/unityadmin/IBM/LogAnalysis/wlp/usr/
servers/Unity/apps/Unity.war/WEB-INF/lib/log4j-1.2.16.jar:/home/
unityadmin/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/
WEB-INF/classes com.ibm.tivoli.unity.tools.VersionUtil
JAVA_HOME         = /home/unityadmin/IBM/LogAnalysis/ibm-java
LANG              = en_US.UTF-8
LESSOPEN          = |/usr/bin/lesspipe.sh %s
LOADEDMODULES     =
LOGNAME           = unityadmin
LS_COLORS         = rs=0:di=01;34:ln=01;36:mh=00:pi=40;33:so=
01;35:do=01;35:bd=40;33;01:cd=40;33;01:or=40;31;01:mi=01;05;37;41:su=37;41:
sg=30;43:ca=30;41:tw=30;42:ow=34;42:st=37;44:ex=01;32:*.tar=01;31:*.tgz=01;
31:*.arj=01;31:*.taz=01;31:*.lzh=01;31:*.lzm=01;31:*.tlz=01;31:*.txz=01;31
:*.zip=01;31:*.z=01;31:*.Z=01;31:*.dz=01;31:*.gz=01;31:*.lz=01;31:*.xz=01;31
:*.bz2=01;31:*.tbz=01;31:*.tbz2=01;31:*.bz=01;31:*.tz=01;31:*.deb=01;
31:*.rpm=01;31:*.jar=01;31:*.ace=01;31:*.zoo=01;31:*.cpio=01;31
:*.7z=01;31:*.rz=01;31:*.jpg=01;35:*.jpeg=01;35:*.gif=01;35:*.bmp=01;35:*.pbm=
01;35:*.pgm=01;35:*.ppm=01;35:*.tga=01;35:*.xbm=01;35:*.xpm=01;35:*.tif=01;35:
*.tiff=01;35:*.png=01;35:*.svg=01;35:*.svgz=01;35:*.mng=01;35:*.pcx=01;35:
*.mov=01;35:*.mpg=01;35:*.mpeg=01;35:*.m2v=01;35:*.mkv=01;35:*.ogm=01;35:
*.mp4=01;35:*.m4v=01;35:*.mp4v=01;35:*.vob=01;35:*.qt=01;35:*.nuv=01;35:*.
wmv=01;35:*.asf=01;35:*.rm=01;35:*.rmvb=01;35:*.flc=01;35:*.avi=01;35:*.fli
=01;35:*.flv=01;35:*.gl=01;35:*.dl=01;35:*.xcf=01;35:*.xwd=01;35:*.yuv=01;35:
*.cgm=01;35:*.emf=01;35:*.axv=01;35:*.anx=01;35:*.ogv=01;35:*.ogx=01;
35:*.aac=01;36:*.au=01;36:*.flac=01;36:*.mid=01;36:*.midi=01;36:*.mka=
01;36:*.mp3=01;36:*.mpc=01;36:*.ogg=01;36:*.ra=01;36:*.wav=01;36:*.axa=01;
36:*.oga=01;36:*.spx=01;36:*.xspf=01;36:
MAIL              = /var/spool/mail/unityadmin
```

```

module                                = () { eval `/usr/bin/modulecmd bash $*`
}
MODULEPATH                            = /usr/share/Modules/modulefiles:/etc/modulefiles
MODULESHOME                           = /usr/share/Modules
PATH                                  = /usr/lib64/qt-3.3/bin:/usr/local/bin:/bin:/usr/bin:/usr/
local/sbin:/usr/sbin:/sbin:/home/unityadmin/bin
PWD                                   = /home/unityadmin/IBM/LogAnalysis/utilities
QTDIR                                 = /usr/lib64/qt-3.3
QTINC                                 = /usr/lib64/qt-3.3/include
QTLIB                                 = /usr/lib64/qt-3.3/lib
SHELL                                 = /bin/bash
SHLVL                                 = 2
SSH_ASKPASS                           = /usr/libexec/openssh/gnome-ssh-askpass
SSH_CLIENT                            = 9.162.129.94 49216 22
SSH_CONNECTION                         = 9.162.129.94 49216 9.118.42.236 22
SSH_TTY                               = /dev/pts/0
TERM                                  = xterm
USER                                  = unityadmin
=====
UNITY Installed Component Version
=====
IBM Operations Analytics - Log Analysis_1_3_2_0_201507240538 ENTRY EDITION

Product License Information
Trial Download. No License.

=====
WebSphere Version Information
=====
Product name: WebSphere Application Server
Product version: 8.5.5.6
Product edition: LIBERTY_CORE

=====
Derby Version Information
=====
----- Java Information -----
Java Version:      1.8.0
Java Vendor:       IBM Corporation
Java home:         /home/unityadmin/IBM/LogAnalysis/ibm-java/jre
Java classpath:    /home/unityadmin/IBM/LogAnalysis/database/db_derby_bin/lib/derby.jar:/home/
unityadmin/IBM/LogAnalysis/database/db_derby_bin/lib/derbynet.jar:/home/unityadmin/IBM/
LogAnalysis/database/db_derby_bin/lib/derbytools.jar:/home/unityadmin/IBM/LogAnalysis/
database/db_derby_bin/lib/derbyclient.jar
OS name:           Linux
OS architecture:  amd64
OS version:        2.6.32-279.el6.x86_64
Java user name:    unityadmin
Java user home:    /home/unityadmin
Java user dir:     /home/unityadmin/IBM/LogAnalysis/utilities
java.specification.name: Java Platform API Specification
java.specification.version: 1.8
java.runtime.version: pxa6480sr1-20150417_01 (SR1)
java.fullversion:  JRE 1.8.0 IBM J9 2.8 Linux amd64-64 Compressed References
20150410_243669 (JIT enabled, AOT enabled)
J9VM - R28_Java8_SR1_20150410_1531_B243669
JIT  - tr.r14.java_20150402_88976.03
GC   - R28_Java8_SR1_20150410_1531_B243669_CMPRSS
J9CL - 20150410_243669
----- Derby Information -----
[/home/unityadmin/IBM/LogAnalysis/database/db_derby_bin/
lib/derby.jar] 10.10.2.1 - (1643489)
[/home/unityadmin/IBM/LogAnalysis/database/db_derby_bin/
lib/derbytools.jar] 10.10.2.1 - (1643489)
[/home/unityadmin/IBM/LogAnalysis/database/db_derby_bin/
lib/derbynet.jar] 10.10.2.1 - (1643489)
[/home/unityadmin/IBM/LogAnalysis/database/db_derby_bin/
lib/derbyclient.jar] 10.10.2.1 - (1643489)
-----
----- Locale Information -----
Current Locale : [English/United States [en_US]]
Found support for locale: [cs]
version: 10.10.2.1 - (1643489)
Found support for locale: [de_DE]
version: 10.10.2.1 - (1643489)
Found support for locale: [es]
version: 10.10.2.1 - (1643489)
Found support for locale: [fr]
version: 10.10.2.1 - (1643489)
Found support for locale: [hu]

```

```

        version: 10.10.2.1 - (1643489)
Found support for locale: [it]
        version: 10.10.2.1 - (1643489)
Found support for locale: [ja_JP]
        version: 10.10.2.1 - (1643489)
Found support for locale: [ko_KR]
        version: 10.10.2.1 - (1643489)
Found support for locale: [pl]
        version: 10.10.2.1 - (1643489)
Found support for locale: [pt_BR]
        version: 10.10.2.1 - (1643489)
Found support for locale: [ru]
        version: 10.10.2.1 - (1643489)
Found support for locale: [zh_CN]
        version: 10.10.2.1 - (1643489)
Found support for locale: [zh_TW]
        version: 10.10.2.1 - (1643489)
-----

=====
ITM Version Information
=====
--> Version information command is not available:
    /home/unityadmin/IBM/LogAnalysis/IBM-LFA-6.30/bin/cinfo.

=====
Insight Pack Information
=====
Buildfile: /home/unityadmin/IBM/LogAnalysis/utilities/pkg_mgmt.xml

initializeCustomTasks:

main:
[packagemanager] 07/28/15 07:01:07:776 EDT [main] INFO
- PrerequisitesManager : CTGLC0044I : Running
prerequisite checks...
[packagemanager] 07/28/15 07:01:07:790 EDT [main] INFO
- PrerequisitesManager : CTGLC0045I : Prerequisite
checks passed
[packagemanager] 07/28/15 07:01:07:792 EDT [main] INFO
- ContentPackManager : CTGLC0030I : Listing installed
insight packs started...
[packagemanager] 07/28/15 07:01:07:860 EDT [main] INFO
- ContentPackManager :
[packagemanager] GAInsightPack_v1.1.1.3
- /home/unityadmin/IBM/LogAnalysis/unity_content
[packagemanager] DB2AppInsightPack_v1.1.0.3
- /home/unityadmin/IBM/LogAnalysis/unity_content
[packagemanager] DB2InsightPack_v1.1.0.2
- /home/unityadmin/IBM/LogAnalysis/unity_content
[packagemanager] WindowsOSEventsInsightPack_v1.1.0.3
- /home/unityadmin/IBM/LogAnalysis/unity_content
[packagemanager] Sample_weblogInsightpack_v1.0.0.0
- /home/unityadmin/IBM/LogAnalysis/unity_content
[packagemanager] WASInsightPack_v1.1.0.3
- /home/unityadmin/IBM/LogAnalysis/unity_content
[packagemanager] JavacoreInsightPack_v1.1.0.3
- /home/unityadmin/IBM/LogAnalysis/unity_content
[packagemanager] SyslogInsightPack_v1.1.0.3
- /home/unityadmin/IBM/LogAnalysis/unity_content
[packagemanager] WASAppInsightPack_v1.1.0.3
- /home/unityadmin/IBM/LogAnalysis/unity_content
[packagemanager] WebAccessLogInsightPack_v1.1.0.2
- /home/unityadmin/IBM/LogAnalysis/unity_content
[packagemanager] Sample_EventInsightpack_v1.0.0.0
- /home/unityadmin/IBM/LogAnalysis/unity_content
[packagemanager] Sample_AppTransInsightpack_v1.0.0.0
- /home/unityadmin/IBM/LogAnalysis/unity_content
[packagemanager] 07/28/15 07:01:07:872 EDT [main] INFO
- ContentPackManager : CTGLC0031I : Listing completed
successfully

BUILD SUCCESSFUL
Total time: 4 seconds

=====
UNITY Jar Version Information
=====
UnityAqlUDF.jar
Build ID: 201507230424
Version: 1.1.0.0

```

```

Path:      /home/unityadmin/IBM/LogAnalysis/unity_content/
DB2InsightPack_v1.1.0.2/extractors/ruleset/common/lib

UnityAqlUDF.jar
Build ID: 201507230424
Version: 1.1.0.0
Path:      /home/unityadmin/IBM/LogAnalysis/unity_content/
GAInsightPack_v1.1.1.3/extractors/ruleset/common/lib

UnityAqlUDF.jar
Build ID: 201306070311
Version: 1.1.0.0
Path:      /home/unityadmin/IBM/LogAnalysis/unity_content/
Sample_weblogInsightpack_v1.0.0.0/extractors/ruleset/common/lib

UnityAqlUDF.jar
Build ID: 201507230424
Version: 1.1.0.0
Path:      /home/unityadmin/IBM/LogAnalysis/unity_content/
SyslogInsightPack_v1.1.0.3/extractors/ruleset/common/lib

UnityAqlUDF.jar
Build ID: 201507230424
Version: 1.1.0.0
Path:      /home/unityadmin/IBM/LogAnalysis/unity_content/
WASInsightPack_v1.1.0.3/extractors/ruleset/common/lib

UnityAqlUDFDB2.jar
Build ID: 201507230424
Version: 1.1.0.0
Path:      /home/unityadmin/IBM/LogAnalysis/unity_content/
DB2InsightPack_v1.1.0.2/extractors/ruleset/commonDB2/lib

UnityAqlUDFDate.jar
Build ID: 201507230424
Version: 1.1.0.0
Path:      /home/unityadmin/IBM/LogAnalysis/unity_content/
GAInsightPack_v1.1.1.3/extractors/ruleset/dateTimeSplitter/lib

UnityAqlUDFDate.jar
Build ID: 201507230424
Version: 1.1.0.0
Path:      /home/unityadmin/IBM/LogAnalysis/unity_content/
GAInsightPack_v1.1.1.3/extractors/ruleset/
normalizedAmericanDateTimeSplitter/lib

UnityAqlUDFDate.jar
Build ID: 201507230424
Version: 1.1.0.0
Path:      /home/unityadmin/IBM/LogAnalysis/unity_content/
GAInsightPack_v1.1.1.3/extractors/ruleset/
normalizedEuropeanDateTimeSplitter/lib

UnityAqlUDFDate.jar
Build ID: 201507230424
Version: 1.1.0.0
Path:      /home/unityadmin/IBM/LogAnalysis/unity_content/
GAInsightPack_v1.1.1.3/extractors/ruleset/
normalizedYearFirstDateTimeSplitter/lib

UnityAqlUDFDate.jar
Build ID: 201507230424
Version: 1.1.0.0
Path:      /home/unityadmin/IBM/LogAnalysis/unity_content/
GAInsightPack_v1.1.1.3/extractors/ruleset/
timeOnlySplitter/lib

UnityAqlUDFDate.jar
Build ID: 201306070311
Version: 1.1.0.0
Path:      /home/unityadmin/IBM/LogAnalysis/unity_content/
Sample_weblogInsightpack_v1.0.0.0/extractors/ruleset/dateTimeSplitter/lib

UnityAqlUDFPatternMatcher.jar
Build ID: 201507230424
Version: 1.1.0.0
Path:      /home/unityadmin/IBM/LogAnalysis/unity_content/
DB2InsightPack_v1.1.0.2/extractors/ruleset/common/lib

UnityAqlUDFPatternMatcher.jar
Build ID: 201507230424
Version: 1.1.0.0

```

```

Path: /home/unityadmin/IBM/LogAnalysis/unity_content/
GAInsightPack_v1.1.1.3/extractors/ruleset/common/lib

UnityAqlUDFPatternMatcher.jar
Build ID: 201306070311
Version: 1.1.0.0
Path: /home/unityadmin/IBM/LogAnalysis/unity_content/
Sample_weblogInsightpack_v1.0.0.0/extractors/ruleset/common/lib

UnityAqlUDFPatternMatcher.jar
Build ID: 201507230424
Version: 1.1.0.0
Path: /home/unityadmin/IBM/LogAnalysis/unity_content/
SyslogInsightPack_v1.1.0.3/extractors/ruleset/common/lib

UnityAqlUDFPatternMatcher.jar
Build ID: 201507230424
Version: 1.1.0.0
Path: /home/unityadmin/IBM/LogAnalysis/unity_content/
WASInsightPack_v1.1.0.3/extractors/ruleset/common/lib

UnityAqlUDFSyslogDate.jar
Build ID: 201507230424
Version: 1.1.0.0
Path: /home/unityadmin/IBM/LogAnalysis/unity_content/
SyslogInsightPack_v1.1.0.3/extractors/ruleset/dateTimeSplitter/lib

UnityEIFReceiver.jar
Build ID: 201507240538
Version: 1.3.2.0
Path: /home/unityadmin/IBM/LogAnalysis/UnityEIFReceiver/jars

datacollector-client.jar
Build ID: 201507240538
Version: 1.3.2.0
Path: /home/unityadmin/IBM/LogAnalysis/utilities/
datacollector-client

db2-content.jar
Build ID: 201507240538
Version: 1.3.2.0
Path: /home/unityadmin/IBM/LogAnalysis/DataCollector/
annotators/jars

db2-content.jar
Build ID: 201507240538
Version: 1.3.2.0
Path: /home/unityadmin/IBM/LogAnalysis/work_files/
Configurations/UnityContent/annotator/java

logAAqlUDFs.jar
Build ID: 201507230424
Version: 1.1.0.0
Path: /home/unityadmin/IBM/LogAnalysis/unity_content/
DB2InsightPack_v1.1.0.2/extractors/ruleset/common/lib

logAAqlUDFs.jar
Build ID: 201507230424
Version: 1.1.0.0
Path: /home/unityadmin/IBM/LogAnalysis/unity_content/
GAInsightPack_v1.1.1.3/extractors/ruleset/common/lib

logAAqlUDFs.jar
Build ID: 201306070311
Version: 1.1.0.0
Path: /home/unityadmin/IBM/LogAnalysis/unity_content/
Sample_weblogInsightpack_v1.0.0.0/extractors/ruleset/common/lib

logAAqlUDFs.jar
Build ID: 201507230424
Version: 1.1.0.0
Path: /home/unityadmin/IBM/LogAnalysis/unity_content/
SyslogInsightPack_v1.1.0.3/extractors/ruleset/common/lib

logAAqlUDFs.jar
Build ID: 201507230424
Version: 1.1.0.0
Path: /home/unityadmin/IBM/LogAnalysis/unity_content/
WASInsightPack_v1.1.0.3/extractors/ruleset/common/lib

unity-analytics-framework.jar
Build ID: 201507240538

```

```

Version: 1.3.2.0
Path: /home/unityadmin/IBM/LogAnalysis/UnityEIFReceiver/jars
=====
Solr Information
=====

nc9118042236.in.ibm.com: -
solr-spec : 5.2.1
solr-impl : 5.2.1

```

Supported time zone names

IBM Operations Analytics - Log Analysis uses Coordinated Universal Time as the default timezone.

If you need to change the default timezone, you must change the setting after you install IBM Operations Analytics - Log Analysis but before you load any data, including the sample data that is provided on the **Getting Started** page. You cannot change the timezone after you load data.

You must use the full timezone name rather than the timezone abbreviation in the timezone parameter. For example, to change the timezone to Central European Time, edit the parameter as follows:

```
UNITY_TIME_ZONE=Europe/Paris
```

For a full list of the supported time zones see the table below.

<i>Table 24. Supported timezone names</i>
Pacific/Midway
Pacific/Niue
Pacific/Pago_Pago
Pacific/Samoa
US/Samoa
America/Adak
America/Atka
Pacific/Honolulu
Pacific/Johnston
Pacific/Rarotonga
Pacific/Tahiti
US/Aleutian
US/Hawaii
Pacific/Marquesas
America/Anchorage
America/Juneau
America/Nome
America/Sitka
America/Yakutat
Pacific/Gambier
US/Alaska
America/Dawson

<i>Table 24. Supported timezone names (continued)</i>
America/Ensenada
America/Los_Angeles
America/Metlakatla
America/Santa_Isabel
America/Tijuana
America/Vancouver
America/Whitehorse
Canada/Pacific
Canada/Yukon
Mexico/BajaNorte
Pacific/Pitcairn
US/Pacific
US/Pacific-New
America/Boise
America/Cambridge_Bay
America/Chihuahua
America/Creston
America/Dawson_Creek
America/Denver
America/Edmonton
America/Hermosillo
America/Inuvik
America/Mazatlan
America/Ojinaga
America/Phoenix
America/Shiprock
America/Yellowknife
Canada/Mountain
Mexico/BajaSur
Navajo
US/Arizona
US/Mountain
America/Bahia_Banderas
America/Belize
America/Cancun

<i>Table 24. Supported timezone names (continued)</i>
America/Chicago
America/Costa_Rica
America/El_Salvador
America/Guatemala
America/Indiana/Knox
America/Indiana/Tell_City
America/Knox_IN
America/Managua
America/Matamoros
America/Menominee
America/Merida
America/Mexico_City
America/Monterrey
America/North_Dakota/Beulah
America/North_Dakota/Center
America/North_Dakota/New_Salem
America/Rainy_River
America/Rankin_Inlet
America/Regina
America/Resolute
America/Swift_Current
America/Tegucigalpa
America/Winnipeg
Canada/Central
Canada/East-Saskatchewan
Canada/Saskatchewan
Chile/EasterIsland
Mexico/General
Pacific/Easter
Pacific/Galapagos
US/Central
US/Indiana-Starke
America/Atikokan
America/Bogota
America/Cayman

<i>Table 24. Supported timezone names (continued)</i>
America/Coral_Harbour
America/Detroit
America/Eirunepe
America/Fort_Wayne
America/Grand_Turk
America/Guayaquil
America/Havana
America/Indiana/Indianapolis
America/Indiana/Marengo
America/Indiana/Petersburg
America/Indiana/Vevay
America/Indiana/Vincennes
America/Indiana/Winamac
America/Indianapolis
America/Iqaluit
America/Jamaica
America/Kentucky/Louisville
America/Kentucky/Monticello
America/Lima
America/Louisville
America/Montreal
America/Nassau
America/New_York
America/Nipigon
America/Panama
America/Pangnirtung
America/Port-au-Prince
America/Porto_Acre
America/Rio_Branco
America/Thunder_Bay
America/Toronto
Brazil/Acre
Canada/Eastern
Cuba
Jamaica

<i>Table 24. Supported timezone names (continued)</i>
US/East-Indiana
US/Eastern
US/Michigan
America/Caracas
America/Anguilla
America/Antigua
America/Aruba
America/Asuncion
America/Barbados
America/Blanc-Sablon
America/Boa_Vista
America/Campo_Grande
America/Cuiaba
America/Curacao
America/Dominica
America/Glace_Bay
America/Goose_Bay
America/Grenada
America/Guadeloupe
America/Guyana
America/Halifax
America/Kralendijk
America/La_Paz
America/Lower_Princes
America/Manaus
America/Marigot
America/Martinique
America/Moncton
America/Montserrat
America/Port_of_Spain
America/Porto_Velho
America/Puerto_Rico
America/Santiago
America/Santo_Domingo
America/St_Barthlemy

<i>Table 24. Supported timezone names (continued)</i>
America/St_Kitts
America/St_Lucia
America/St_Thomas
America/St_Vincent
America/Thule
America/Tortola
America/Virgin
Antarctica/Palmer
Atlantic/Bermuda
Brazil/West
Canada/Atlantic
Chile/Continental
America/St_Johns
Canada/Newfoundland
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/Salta
America/Argentina/San_Juan
America/Argentina/San_Luis
America/Argentina/Tucuman
America/Argentina/Ushuaia
America/Bahia
America/Belem
America/Buenos_Aires
America/Catamarca
America/Cayenne
America/Cordoba
America/Fortaleza

<i>Table 24. Supported timezone names (continued)</i>
America/Godthab
America/Jujuy
America/Maceio
America/Mendoza
America/Miquelon
America/Montevideo
America/Paramaribo
America/Recife
America/Rosario
America/Santarem
America/Sao_Paulo
Antarctica/Rothera
Atlantic/Stanley
Brazil/East
America/Noronha
Atlantic/South_Georgia
Brazil/DeNoronha
America/Scoresbysund
Atlantic/Azores
Atlantic/Cape_Verde
Africa/Abidjan
Africa/Accra
Africa/Bamako
Africa/Banjul
Africa/Bissau
Africa/Casablanca
Africa/Conakry
Africa/Dakar
Africa/El_Aaiun
Africa/Freetown
Africa/Lome
Africa/Monrovia
Africa/Nouakchott
Africa/Ouagadougou
Africa/Sao_Tome

<i>Table 24. Supported timezone names (continued)</i>
Africa/Timbuktu
America/Danmarkshavn
Antarctica/Troll
Atlantic/Canary
Atlantic/Faeroe
Atlantic/Faroe
Atlantic/Madeira
Atlantic/Reykjavik
Atlantic/St_Helena
Eire
Europe/Belfast
Europe/Dublin
Europe/Guernsey
Europe/Isle_of_Man
Europe/Jersey
Europe/Lisbon
Europe/London
GB
GB-Eire
Greenwich
Iceland
Portugal
Universal
Zulu
Africa/Algiers
Africa/Bangui
Africa/Brazzaville
Africa/Ceuta
Africa/Douala
Africa/Kinshasa
Africa/Lagos
Africa/Libreville
Africa/Luanda
Africa/Malabo
Africa/Ndjamena

<i>Table 24. Supported timezone names (continued)</i>
Africa/Niamey
Africa/Porto-Novo
Africa/Tunis
Africa/Windhoek
Arctic/Longyearbyen
Atlantic/Jan_Mayen
Europe/Amsterdam
Europe/Andorra
Europe/Belgrade
Europe/Berlin
Europe/Bratislava
Europe/Brussels
Europe/Budapest
Europe/Busingen
Europe/Copenhagen
Europe/Gibraltar
Europe/Ljubljana
Europe/Luxembourg
Europe/Madrid
Europe/Malta
Europe/Monaco
Europe/Oslo
Europe/Paris
Europe/Podgorica
Europe/Prague
Europe/Rome
Europe/San_Marino
Europe/Sarajevo
Europe/Skopje
Europe/Stockholm
Europe/Tirane
Europe/Vaduz
Europe/Vatican
Europe/Vienna
Europe/Warsaw

<i>Table 24. Supported timezone names (continued)</i>
Europe/Zagreb
Europe/Zurich
Poland
Africa/Blantyre
Africa/Bujumbura
Africa/Cairo
Africa/Gaborone
Africa/Harare
Africa/Johannesburg
Africa/Kigali
Africa/Lubumbashi
Africa/Lusaka
Africa/Maputo
Africa/Maseru
Africa/Mbabane
Africa/Tripoli
Asia/Amman
Asia/Beirut
Asia/Damascus
Asia/Gaza
Asia/Hebron
Asia/Istanbul
Asia/Jerusalem
Asia/Nicosia
Asia/Tel_Aviv
Egypt
Europe/Athens
Europe/Bucharest
Europe/Chisinau
Europe/Helsinki
Europe/Istanbul
Europe/Kiev
Europe/Mariehamn
Europe/Nicosia
Europe/Riga

<i>Table 24. Supported timezone names (continued)</i>
Europe/Sofia
Europe/Tallinn
Europe/Tiraspol
Europe/Uzhgorod
Europe/Vilnius
Europe/Zaporozhye
Israel
Libya
Turkey
Africa/Addis_Ababa
Africa/Asmara
Africa/Asmera
Africa/Dar_es_Salaam
Africa/Djibouti
Africa/Juba
Africa/Kampala
Africa/Khartoum
Africa/Mogadishu
Africa/Nairobi
Antarctica/Syowa
Asia/Aden
Asia/Baghdad
Asia/Bahrain
Asia/Kuwait
Asia/Qatar
Asia/Riyadh
Europe/Kaliningrad
Europe/Minsk
Indian/Antananarivo
Indian/Comoro
Indian/Mayotte
Asia/Riyadh87
Asia/Riyadh88
Asia/Riyadh89
Mideast/Riyadh87

<i>Table 24. Supported timezone names (continued)</i>
Mideast/Riyadh88
Mideast/Riyadh89
Asia/Tehran
Iran
Asia/Baku
Asia/Dubai
Asia/Muscat
Asia/Tbilisi
Asia/Yerevan
Europe/Moscow
Europe/Samara
Europe/Simferopol
Europe/Volgograd
Indian/Mahe
Indian/Mauritius
Indian/Reunion
Asia/Kabul
Antarctica/Mawson
Asia/Aqtau
Asia/Aqtobe
Asia/Ashgabat
Asia/Ashkhabad
Asia/Dushanbe
Asia/Karachi
Asia/Oral
Asia/Samarkand
Asia/Tashkent
Indian/Kerguelen
Indian/Maldives
Asia/Calcutta
Asia/Colombo
Asia/Kolkata
Asia/Kathmandu
Asia/Katmandu
Antarctica/Vostok

<i>Table 24. Supported timezone names (continued)</i>
Asia/Almaty
Asia/Bishkek
Asia/Dacca
Asia/Dhaka
Asia/Qyzylorda
Asia/Thimbu
Asia/Thimphu
Asia/Yekaterinburg
Indian/Chagos
Asia/Rangoon
Indian/Cocos
Antarctica/Davis
Asia/Bangkok
Asia/Ho_Chi_Minh
Asia/Hovd
Asia/Jakarta
Asia/Novokuznetsk
Asia/Novosibirsk
Asia/Omsk
Asia/Phnom_Penh
Asia/Pontianak
Asia/Saigon
Asia/Vientiane
Indian/Christmas
Antarctica/Casey
Asia/Brunei
Asia/Choibalsan
Asia/Chongqing
Asia/Chungking
Asia/Harbin
Asia/Hong_Kong
Asia/Kashgar
Asia/Krasnoyarsk
Asia/Kuala_Lumpur
Asia/Kuching

<i>Table 24. Supported timezone names (continued)</i>
Asia/Macao
Asia/Macau
Asia/Makassar
Asia/Manila
Asia/Shanghai
Asia/Singapore
Asia/Taipei
Asia/Ujung_Pandang
Asia/Ulaanbaatar
Asia/Ulan_Bator
Asia/Urumqi
Australia/Perth
Australia/West
Hongkong
Singapore
Australia/Eucla
Asia/Dili
Asia/Irkutsk
Asia/Jayapura
Asia/Pyongyang
Asia/Seoul
Asia/Tokyo
Japan
Pacific/Palau
Australia/Adelaide
Australia/Broken_Hill
Australia/Darwin
Australia/North
Australia/South
Australia/Yancowinna
Antarctica/DumontDUrville
Asia/Khandyga
Asia/Yakutsk
Australia/ACT
Australia/Brisbane

<i>Table 24. Supported timezone names (continued)</i>
Australia/Canberra
Australia/Currie
Australia/Hobart
Australia/Lindeman
Australia/Melbourne
Australia/NSW
Australia/Queensland
Australia/Sydney
Australia/Tasmania
Australia/Victoria
Pacific/Chuuk
Pacific/Guam
Pacific/Port_Moresby
Pacific/Saipan
Pacific/Truk
Pacific/Yap
Australia/LHI
Australia/Lord_Howe
Antarctica/Macquarie
Asia/Sakhalin
Asia/Ust-Nera
Asia/Vladivostok
Pacific/Efate
Pacific/Guadalcanal
Pacific/Kosrae
Pacific/Noumea
Pacific/Pohnpei
Pacific/Ponape
Pacific/Norfolk
Antarctica/McMurdo
Antarctica/South_Pole
Asia/Anadyr
Asia/Kamchatka
Asia/Magadan
Kwajalein

<i>Table 24. Supported timezone names (continued)</i>
Pacific/Auckland
Pacific/Fiji
Pacific/Funafuti
Pacific/Kwajalein
Pacific/Majuro
Pacific/Nauru
Pacific/Tarawa
Pacific/Wake
Pacific/Wallis
NZ-CHAT
Pacific/Chatham
Pacific/Apia
Pacific/Enderbury
Pacific/Fakaofu
Pacific/Tongatapu
Pacific/Kiritimati

Supported languages

IBM Operations Analytics - Log Analysis supports multiple languages.

Refer to the following table of languages codes for supported languages.

<i>Table 25.</i>	
Language	Language code
Brazilian-Portuguese	pt-BR
French	fr
German	de
Italian	it
Japanese	ja
Korean	ko
Russian	ru
Spanish	es
Simplified Chinese	zh-cn
Traditional Chinese	zh-tw

Chapter 5. Data tiering and storage

After you install and configure Log Analysis, you can configure how Log Analysis stores data.

You can divide data that Log Analysis stores into the current and archive tiers.

Standard Standard Edition users can integrate Log Analysis with Hadoop to store long term data.

Configuring the data archive

To optimize performance, configure the length of time that IBM Operations Analytics - Log Analysis stores data in the archive.

About this task

You must configure the archive period before you load any data.

To facilitate query performance, IBM Operations Analytics - Log Analysis uses data tiers to prioritize the retrieval of the most recent information.

IBM Operations Analytics - Log Analysis divides data into two tiers, the current tier and the archive tier. Data is held in the current tier for the specified period. After this time elapses and more data is loaded, it is moved to the archive tier. Data that is stored in the current tier is stored in memory. Therefore, the queries can access this data more quickly than data in the archive tier.

You use the `HOT_TIER_PERIOD` parameter to specify the number of days that data is held in the current tier. For example, if the `HOT_TIER_PERIOD` parameter is set to 2, data is held in the current tier for two days until the next ingestion of data.

The length of the time period that you specify for the `HOT_TIER_PERIOD` parameter affects the amount of memory that IBM Operations Analytics - Log Analysis uses. The longer the period, the greater the memory used.

Procedure

1. To stop the IBM Operations Analytics - Log Analysis server, use the following command:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
```

2. Open the `unitysetup.properties` file that is in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF` directory.
3. To specify the number of days that data is stored in the current tier, change the value for the `HOT_TIER_PERIOD` parameter. The default value is 2:

```
HOT_TIER_PERIOD=2
```

4. Save the file.
5. To start the IBM Operations Analytics - Log Analysis server, use the following command:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -start
```

Results

Data is held in the current tier until the next data ingestion for the specified period.

Standard If you use the Standard Edition of Log Analysis, you can integrate Log Analysis with Hadoop to store long-term data.

Before you can use Hadoop for long-term data storage, you must create DataNode and NameNode connections to integrate Log Analysis with your Hadoop installation.

If you want to integrate IBM BigInsights, Hortonworks, or Huawei FusionInsight HD you must use the UI to create the connections automatically.

Fix Pack 1 If you want to use Huawei FusionInsight HD, you must install IBM Operations Analytics - Log Analysis 1.3.3 Fix Pack 1.

If you want to integrate Cloudera Hadoop, you must configure it manually.

After the integration is complete, Log Analysis stores data from the archive tier in Hadoop. You can use Log Analysis to search this data when you need to.

For more information about the supported versions of Hadoop, see [“Supported versions of Hadoop” on page 137](#).

Why should I use Hadoop?

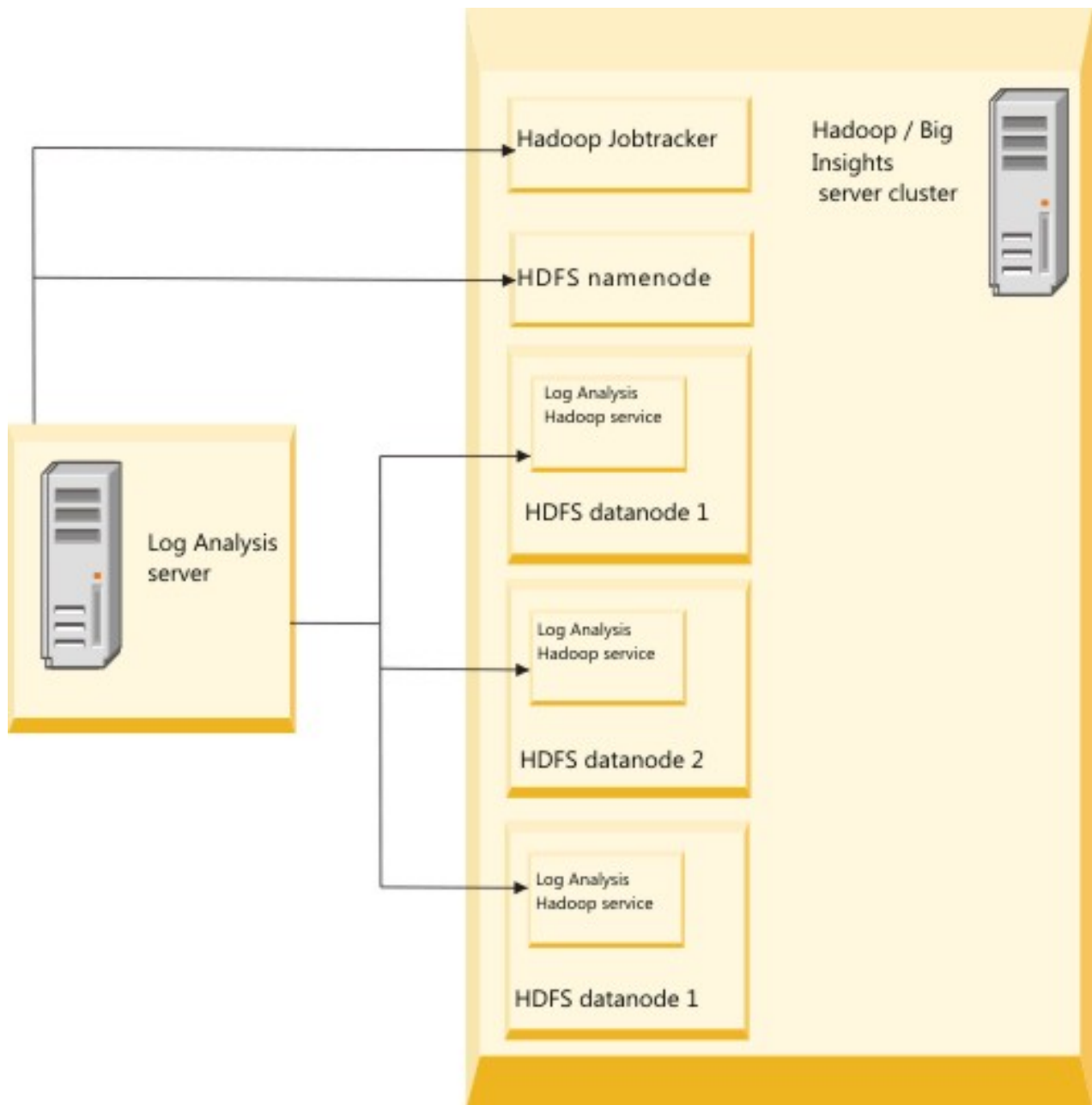
Hadoop offers a more efficient method for long-term data storage that you can use to store long-term data from annotated log files. The integration with Log Analysis is facilitated by a service that is bundled with Log Analysis and ensures that you can continue to search this data without need to store the data in the main database.

Architecture

Log Analysis stores the log data on to the Hadoop cluster and allows users to search data that is stored in Hadoop. Log Analysis, with the help of the Log Analysis service that is installed on each datanode, writes the data to the Hadoop cluster. The data is written in the avro object container file format. For more information about object container files, see <http://avro.apache.org/docs/1.7.4/spec.html#Object+Container+Files>. Data is then written to each DataNode where the service is installed. You can use IBM Operations Analytics - Log Analysis to search this data.

You can also run Log Analysis searches on the data stored on the Hadoop cluster.

The following graphic displays an overview of the service architecture:



Supported versions of Hadoop

Ensure that your version of Hadoop is supported by Log Analysis.

Log Analysis supports integrations with the following versions of Hadoop:

IBM Open Platform with Apache Hadoop for IBM BigInsights 4.0 and 4.1

IBM InfoSphere BigInsights® delivers a rich set of advanced analytics capabilities that allows enterprises to analyze massive volumes of structured and unstructured data in its native format. For more information, see the IBM InfoSphere BigInsights documentation at http://www-01.ibm.com/support/knowledgecenter/SSPT3X/SSPT3X_welcome.html.

IBM BigInsights consists of a number of modules. To integrate with Log Analysis, you must download and install IBM Open Platform with Apache Hadoop. For more information about how to download this component, see <http://www-01.ibm.com/support/docview.wss?uid=swg24040517>.

IBM InfoSphere BigInsights 3.0

IBM InfoSphere BigInsights 3.0 is supported on Log Analysis 1.3.3, however you must configure the server connections manually.

Fix Pack 1 IBM InfoSphere BigInsights 3.0 is not supported on Log Analysis 1.3.3.1, or higher.

Cloudera Hadoop 5.3.0

Cloudera provides a scalable, flexible, integrated platform that makes it easy to manage rapidly increasing volumes and varieties of data in your enterprise. For more information, see the Cloudera documentation at <http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/introduction.html>.

Fix Pack 1

Hortonworks Hadoop 2.x

The Hortonworks product, Hortonworks Data Platform provides storage of large volumes of data, and processing and analyzing of that data. It is designed to process and analyze data from various sources and formats. For more information, see the Hortonworks documentation at <http://hortonworks.com/hdp/>.

Fix Pack 1

Huawei Hadoop 2.x

Huawei FusionInsight HD provides storage of large volumes of data, and processing and analyzing of that data. For more information, see the Huawei FusionInsight HD documentation at <http://support.huawei.com/enterprise/product-21110924-en.html>.

Prerequisite tasks

Before you can create the connections to your Hadoop servers, ensure that the prerequisites are met.

Ensure that IBM Open Platform with Apache Hadoop, Cloudera Hadoop, Hortonworks, or Huawei FusionInsight HD is installed in your environment. For more information, see [“Supported versions of Hadoop”](#) on page 137.:

The following tasks are only relevant if you are manually configuring the integration:

- Create a user for each DataNode on the Hadoop cluster.
- Ensure that the username for each DataNode is the same as the IBM Operations Analytics - Log Analysis username on the IBM Operations Analytics - Log Analysis server.

Standard Configuring long term data storage automatically

If you want to integrate IBM Open Platform with Apache Hadoop, Hortonworks Hadoop, or Huawei FusionInsight HD to store your long term data, you must use the Log Analysis UI to create the required connections.

Fix Pack 1 IBM InfoSphere BigInsights 3.0 is not supported on Log Analysis 1.3.3.1, or higher.

If you want to integrate Cloudera Hadoop or IBM InfoSphere BigInsights 3.0, you must configure it manually. For more information, see [“Manually configuring long term data storage”](#) on page 140.

If you use IBM Open Platform with Apache Hadoop 4.0 or 4.1, you cannot configure the server connections manually. You must use the UI to integrate it automatically as described in this section.

Standard Creating a NameNode server connection

Before you can create DataNode server connections, you must create a NameNode server connection.

About this task

Only Standard Edition users can use this feature. You can only create one NameNode server connection.

Procedure


1. Click the **Hadoop Integration** tab.
2. Click  (Create Name Node icon).
3. Complete the fields as described in the table.

Table 26. NameNode server connection editor fields	
Field	Description
Hadoop	Select your Hadoop distribution from the drop-down list.
NameNode Server	Specify either the host name or IP address of the server where you installed the NameNode server.
NameNode Root Password	Specify the root password for the NameNode server.
Namenode SSH Port	Specify the port number that is used by the NameNode server for SSH connections to Log Analysis. The default is 22.
LA Service Port	Specify the port number that is used by the Log Analysis service.
HDFS UserName	Specify the Hadoop user name.
LA Directory Location on HDFS	Specify the path to the directory where you want to store the Log Analysis files on the Hadoop Distributed File System (HDFS). For example / ioala-root.
Hadoop Install Path	Specify the path to the directory where you installed Hadoop on the NameNode server.
Fix Pack 1 Fix Pack 1 Hadoop configuration file location directory (Huawei FusionInsight HD only)	Specify the path to the directory where the Hadoop configuration is located. For more information about configuring the file location directory, see Specifying the Huawei FusionInsight HD configuration file location directory .
Fix Pack 1 Fix Pack 1 Enable Kerberos	To enable Kerberos, select the Enable Kerberos check box. For more information about enabling Kerberos, see Enabling Kerberos .
Fix Pack 1 Fix Pack 1 Kerberos user principal	Specify the principal Kerberos user name and domain. For more information about specifying the user and domain, see Specifying the Kerberos user name and domain .
Fix Pack 1 Fix Pack 1 Kerberos key tab	Browse to the Kerberos key tab. For more information about Kerberos key tab, see Downloading the Kerberos keytab file

4. Click **Add**.

What to do next

After you configure the NameNode server connection, you can configure the Data Node server connections.

Standard **Creating a DataNode server connection**

After you create a NameNode server connection, you must create the associated DataNode server connections.

About this task

Only Standard Edition users can use this feature.

Procedure


1. Click the **Hadoop Integration** tab.
2. Click on the radio that is beside the NameNode connection.
3. Click  (Create DataNode icon).
4. Complete the fields as described in the table.

Table 27. DataNode server connection editor fields	
Field	Description
Hostname	Specify the host name or IP address of the DataNode server.
Root Password	Specify the root password for the DataNode server.
SSH Port	Specify the port that the DataNode server uses to connect to Log Analysis for SSH connections. The default value is 22.
LA Service Home	Specify the path to the directory where the Log Analysis service is deployed on the DataNode server.
Java Home Path	Specify the full path to the directory on the DataNode server where Java is installed.
LA Service Username	The user name that the Log Analysis service uses to connect to Log Analysis to copy the required Java Archive files to the DataNode server. This field is automatically filled with the user name that installed Log Analysis. You cannot change this field.
LA Service User Password	Specify the password for LA service user name. If this user already exists, the password is changed to the value that you specify in this field.

5. Click **Add**.
6. Add any other DataNode server connections repeating the previous steps.
For example, if you use a Hadoop cluster, you can add the remaining DataNode connections.
7. After you add all the required connections, restart Log Analysis.

Standard **Manually configuring long term data storage**

If you want to integrate Cloudera Hadoop with Log Analysis, you must configure the integration manually.

If you use IBM InfoSphere BigInsights 3.0, you must configure the server connections manually.

Fix Pack 1 IBM InfoSphere BigInsights 3.0 is not supported on Log Analysis 1.3.3.1, or higher.

If you use IBM Open Platform with Apache Hadoop 4.0 or 4.1, Hortonworks Hadoop, or Huawei FusionInsight HD you cannot configure the server connections manually. You must use the UI to integrate it. For more information, see “[Configuring long term data storage automatically](#)” on page 138.

Standard Integrating the Hadoop client

Before you can enable the Hadoop tier, you must prepare the Hadoop client.

Procedure

1. Create a folder that is called `hadoop-jars` on the IBM Operations Analytics - Log Analysis server.
2. Copy the Hadoop client from the Hadoop cluster. Choose one of the following options for your chosen Hadoop integration for IBM Operations Analytics - Log Analysis.

- IBM BigInsights 3.0

- a. Open the IBM InfoSphere BigInsights administration webpage. For example, `http://<BI_Cluster_Manager_Host>:8080`
- b. Click **Download client library and development software**.
- c. To download the client package, select **Job Submission API package**.
- d. Download and extract the client package.
- e. Copy the `jar` and `xml` files from the client package to the `hadoop-jars` folder created in step 1.

Note: If IBM Operations Analytics - Log Analysis is installed on Linux on System z based operating system with IBM InfoSphere BigInsights 3.0.0 on x-86, you must replace the `hadoop-core.jar` with the `hadoop-core-2.2.0-mr1.jar`. The `hadoop-core.jar` was copied from the IBM BigInsights cluster. The `hadoop-core-2.2.0-mr1.jar` is supplied with IBM Operations Analytics - Log Analysis, and located here: `<HOME>/IBM/LogAnalysis/utilities/hadoop/BigInsights_3.0.0`

- Cloudera Hadoop

- a. Navigate to the Cloudera parcels folder on the Cloudera Hadoop cluster. For example: `/opt/cloudera/parcels/CDH-5.3.0-1.cd5.3.0.p0.30/lib/hadoop/client`
- b. Copy the `jar` files from the Cloudera parcels folder to the `hadoop-jars` folder created in step 1. For information about obtaining `jar` files, see the Cloudera documentation: http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/cdh_vd_hadoop_api_dependencies.html
- c. Open the cluster managers webpage. For example, `http://<CDH_Cluster_Manager_Host>:7180/cmf/home`.
- d. Download the **Client Configuration Files** for YARN services. For more information about downloading the client configuration files, see the Cloudera documentation: http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/cm_mc_client_config.html
- e. Extract and copy the `xml` files to the `hadoop-jars` folder created in step 1.

Standard Setting up the Hadoop tier

Before you can enable the Hadoop tier, you must set up the Hadoop tier.

Procedure

1. Create a folder that is called `LA_HADOOP_TIER` on the Hadoop Distributed File System (HDFS).

For example, to use the command-line utility to create the folder, run the following command:

```
hadoop fs -mkdir /<la-hadoop-tier>
```

This folder is exclusively used by the IBM Operations Analytics - Log Analysis Hadoop tier.

2. Create the following folders in the LA_HADOOP_TIER folder.

LA_HADOOP_TIER

This folder is used to ingest the data.

LA_HADOOP_TIER/jars

This folder is used to store the jar files required by the map-reduce job.

LA_HADOOP_TIER/output

This folder is used as temporary storage during a IBM Operations Analytics - Log Analysis search query execution over the Hadoop tier.

For example, to use the command-line utility to create these folders, enter the following commands in the command-line:

```
hadoop fs -mkdir /la-hadoop-tier/data
hadoop fs -mkdir /la-hadoop-tier/jars
hadoop fs -mkdir /la-hadoop-tier/output
```

3. Change the ownership of these folders to the LA use.

For example, to use the command-line utility to change the ownership of the folders created in step 2 to the LA user, enter the following command:

```
./hadoop fs -chown -R LA:LA /la-hadoop-tier
```

4. Verify the creation and ownership of these folders.

For example, to use the command-line utility to verify folder details, enter the following commands:

```
hadoop fs -ls /
hadoop fs -ls /la-hadoop-tier
```

5. Copy and extract the search.zip from <HOME>/IBM/LogAnalysis/utilities/hadoop to a temporary folder on the Hadoop cluster.

For example: /tmp/la-search-jars

6. Upload the jars from the temporary folder to HDFS.

For example, to use the command-line utility to load the files from the temporary folder to HDFS, enter the following command:

```
hadoop fs -copyFromLocal /tmp/la-search-jars/*.jar /la-hadoop-tier/jars
```

7. To ensure that the LA user can launch a MapReduce job to the Hadoop cluster, log in as the LA user and launch a test map-reduce job.

For more information about how to launch a map-reduce test on the Cloudera Hadoop, see *Running a MapReduce Job* in the Cloudera documentation: http://www.cloudera.com/content/cloudera/en/documentation/core/latest/topics/cm_ig_testing_the_install.html

Installing the Log Analysis service

To use the Hadoop tier, you must install the Log Analysis service on each Data Node server in the Hadoop cluster.

About this task

The Log Analysis server pushes the log data to the Log Analysis service. The data is then written to Hadoop.

Procedure

To install the IBM Operations Analytics - Log Analysis service, complete the following steps.

1. Log in as the LA user to one of the Data Nodes in the Hadoop cluster.
2. Create a folder that is called LA_SERVICE_HOME.
For example, <HOME>/IBM/LogAnalysis/LA_SERVICE_HOME
3. Copy and extract the service.zip from <HOME>/IBM/LogAnalysis/utilities/hadoop folder to the LA_SERVICE_HOME folder.

4. Copy the LA_HADOOP_TIER/jars folder to the LA_SERVICE_HOME/lib folder.
5. Review, and modify if required, the values for the environment variables in LA_SERVICE_HOME/bin/env.sh script.
6. Copy the LA_SERVICE_HOME folder to all the Data Node servers in the Hadoop cluster.

Standard **Configuring Log Analysis**

Before you can enable the Hadoop tier, you must configure Log Analysis.

Procedure

1. Stop the Log Analysis server:

```
./ unity.sh -stop
```

2. Open the *unitysetup.properties* in the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF directory. Change the following properties:

- HADOOP_TIER_HDFS_BASE_DIR

Specify the value of <LA_HADOOP_TIER> on HDFS.

- INDEX_IMPLEMENTATION=INDEXING_ENGINE

Add Hadoop to the INDEX_IMPLEMENTATION property. For example, change

INDEX_IMPLEMENTATION=INDEXING_ENGINE to

INDEX_IMPLEMENTATION=INDEXING_ENGINE, HADOOP

Update the default HADOOP_TIER.properties file if it is different from your environment. Ignore the HADOOP_TIER_JOB_TRACKER_URI property.

3. Copy the hadoop-jars folder to the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/lib directory
4. Start the Log Analysis server:

```
./ unity.sh -start
```

Standard **Testing the Hadoop integration**

To ensure that the Hadoop integration configured correctly, you can run some basic tests.

Procedure

1. Ingest log data to the Log Analysis server.

- a. You can install sample data on the Log Analysis UI.
- b. You can also use data that had been loaded with an Insight Pack.
- c. Loaded data is written to the avro files in the <LA_HADOOP_TIER>/data folder in Hadoop in the following format:

```
<UnityCollection_Timestamp>/<DATA_SOURCE_NAME>/<DAY_BASED_ON_TIMESTAMP_IN_LOG_RECORDS>/<counter>.avro
```

2. Use the Search UI to search for data that is stored in Hadoop.

- a. Prepend the search query in the UI with [_hq]. For example, [_hq]*
- b. The search on the Hadoop tier is run via a map reduce job on the Hadoop cluster.
- c. Prepend the same search query in the UI with [_sq], for example, [_sq]*, and perform the query on an Indexing Engine.
- d. Compare the search results.

3. To identify errors, open the following files:

- UnityApplication.log on the Log Analysis server.

- <HOME>/IBM/LogAnalysis/logs/hadooptier.log on the Log Analysis server.
- <LA_SERVICE_HOME>/logs on the server where the Log Analysis service is installed.

Standard **Hadoop Integration tab**

Use the **Hadoop Integration** tab to view the status of your current data storage integration and the associated processes.

Only Standard Edition users can use this feature.

The tab consists of 2 tabs:

Configuration tab

Displays the NameNode and DataNode server connections. You can create, edit, and delete connections here. You can also view details for specific connections.

Integration Processes tab

Displays an overview of the processes that are running as part of Log Analysis. The processes that are listed there includes the local EIF Receiver instance, main Log Analysis process, Indexing Engine nodes (Apache Solr nodes), and any connections to Hadoop. You can start, stop, and restart these processes.

Note: You cannot start, stop or restart the Log Analysis process or the Apache Solr processes.

Managing the Log Analysis server

You can stop, start, and restart some of the Log Analysis processes on the UI.

About this task

You can only use this UI to manage the Log Analysis server's core process.

Note: Start, stop, and restart actions are not supported for IBM Operations Analytics - Log Analysis main processes or Apache Solr processes.

Procedure

1. Open the IBM Operations Analytics - Log Analysis UI and click **Administrative settings**.
2. Click the **Hadoop Integration** tab.
3. Click the **Integration Processes** tab.
4. Hover over the core process you want to stop, start, or restart in the Core Process list to display the **Start**, **Stop**, and **Restart** icons.
5. To complete the start, stop, or restart action, click the appropriate icon.

Standard **Maintaining automatically configured connections**

You can delete NameNode and DataNode server connections that you created in the Log Analysis UI.

You can also edit the **LA Service User Password** field for a DataNode server connection. You cannot edit any other fields.

You cannot edit a NameNode server connection after you create it.

Fix Pack 1 Editing a NameNode server connection

After you create a NameNode server connection, you can edit it to enable or disable the Kerberos configuration.

Procedure

1. Click the **Hadoop Integration** tab.

2. Click the Edit NameNode icon.
3. Specify the NameNode root password, Kerberos related properties.
4. Click **Save**.

Standard **Editing a DataNode server connection**

After you create a DataNode server connection, you can edit it.

Procedure

1. Click the **Hadoop Integration** tab.
2. Click the Edit DataNode icon.
3. Edit the LA Service User Password field, as described in the table. You cannot edit any other fields.
4. Click **Save**.


Standard **Deleting a NameNode server connection**

You can delete a NameNode server connection.

Before you begin

You must delete the DataNode server connections that are associated with the NameNode server connection before you can delete the NameNode server connection.

Procedure

1. Click the **Hadoop Integration** tab.
2. Click  (Delete NameNode icon).
3. Click **Ok**.
4. Restart Log Analysis.


What to do next

This action does not delete the data that is stored in Hadoop. Only the connection details are deleted. To delete the stored log file records from Hadoop, you must use the Log Analysis deletion utility. For more information, see [“Deleting data” on page 330](#).

Standard **Deleting a DataNode server connection**

You can delete a DataNode server connection.

Procedure

1. Click the **Hadoop Integration** tab.
2. Click  (Delete DataNode icon).
3. Click **Delete**.
4. Restart Log Analysis.

Standard **Maintaining manually configured DataNode connections**

After you configure the Hadoop service, you can use the `server.sh` script to manage the service.

Procedure

You can choose to manage the service on an individual DataNode server or manage all of the service instances together.

- To manage the service on an individual DataNode, use the LA user that you created when you configured the service to log in to a Hadoop Data Node server.
 - a. Run the `<LA_SERVICE_HOME>/bin/server.sh` script with one of the following parameters:
 - Start**
Starts the service on the Hadoop DataNode server.
 - Stop**
Stops the service on the Hadoop DataNode server.
 - Status**
Retrieves the status for the Hadoop DataNode server.
- To manage all of the instances, select one DataNode to act as a `LA_Service_Controller_Node`. This will manage the service on all of the DataNodes.
 - a. (Optional) Create password-less SSH for the LA user from this DataNode to all of the DataNodes, including this DataNode, in the Hadoop cluster.
 - b. Use the LA user to login to the `LA_Service_Controller_Node` Data Node.
 - c. Run the `<LA_SERVICE_HOME>/bin/server.sh` script with one of the following parameters:
 - clusterStart**
Starts the service on each Hadoop DataNode server.
 - clusterStop**
Stops the service on each Hadoop DataNode server.
 - clusterStatus**
Retrieves the status for each Hadoop DataNode server.

If you do not configure password-less SSH connections during the configuration, you are prompted for the password for each DataNode server.

Standard **Sharing a Hadoop cluster across multiple Log Analysis instances**

You can use the same Hadoop cluster for multiple instances of Log Analysis.

Procedure

1. To share a Hadoop cluster across multiple Log Analysis instances, you must integrate the Hadoop service for each Log Analysis instance.

For more information, see the [“Integrating the Hadoop client” on page 141](#).

- a) You must use a different value for each of the following folders:

```
<LA_HADOOP_TIER>
```

on the DataNode server.

```
<LA_SERVICE_HOME>
```

on the Log Analysis server.

As an alternative to repeating this step for each Log Analysis instance, you can create a copy of the resultant folders from a Log Analysis instance of the same version.

2. Modify the `PORT` and `PROCESS_ID` values in the `<LA_SERVICE_HOME>/bin/env.sh` file

Standard DataNode server connection editor reference

Use the DataNode server connection editor to create DataNode server connections. You can have multiple DataNode connections. All fields are required.

After you create a DataNode server connection, you can only edit the **LA Service Username** field.

Table 28. DataNode server connection editor fields	
Field	Description
Hostname	Specify the host name or IP address of the DataNode server.
Root Password	Specify the root password for the DataNode server.
SSH Port	Specify the port that the DataNode server uses to connect to Log Analysis for SSH connections. The default value is 22.
LA Service Home	Specify the path to the directory where the Log Analysis service is deployed on the DataNode server.
Java Home Path	Specify the full path to the directory on the DataNode server where Java is installed.
LA Service Username	The user name that the Log Analysis service uses to connect to Log Analysis to copy the required Java Archive files to the DataNode server. This field is automatically filled with the user name that installed Log Analysis. You cannot change this field.
LA Service User Password	Specify the password for LA service user name. If this user already exists, the password is changed to the value that you specify in this field.

Standard NameNode server connection editor reference

Use the NameNode server connection editor to create the connection between the NameNode server and Log Analysis. You can only have one Name Node connection.

Table 29. NameNode server connection editor fields	
Field	Description
Hadoop	Select your Hadoop distribution from the drop-down list.
NameNode Server	Specify either the host name or IP address of the server where you installed the NameNode server.
NameNode Root Password	Specify the root password for the NameNode server.
Namenode SSH Port	Specify the port number that is used by the NameNode server for SSH connections to Log Analysis. The default is 22.
LA Service Port	Specify the port number that is used by the Log Analysis service.

Table 29. NameNode server connection editor fields (continued)

Field	Description
HDFS UserName	Specify the Hadoop user name.
LA Directory Location on HDFS	Specify the path to the directory where you want to store the Log Analysis files on the Hadoop Distributed File System (HDFS). For example / ioala-root.
Hadoop Install Path	Specify the path to the directory where you installed Hadoop on the NameNode server.
Fix Pack 1 Fix Pack 1 Hadoop configuration file location directory (Huawei FusionInsight HD only)	Specify the path to the directory where the Hadoop configuration is located. For more information about configuring the file location directory, see Specifying the Huawei FusionInsight HD configuration file location directory .
Fix Pack 1 Fix Pack 1 Enable Kerberos	To enable Kerberos, select the Enable Kerberos check box. For more information about enabling Kerberos, see Enabling Kerberos .
Fix Pack 1 Fix Pack 1 Kerberos user principal	Specify the principal Kerberos user name and domain. For more information about specifying the user and domain, see Specifying the Kerberos user name and domain .
Fix Pack 1 Fix Pack 1 Kerberos key tab	Browse to the Kerberos key tab. For more information about Kerberos key tab, see Downloading the Kerberos keytab file

Chapter 6. Loading and streaming data

Before you can perform a search on log or other data, you must first load the data into IBM Operations Analytics - Log Analysis. When the file is loaded the data is indexed and is then available to be searched.

There are two main scenarios for loading data:

- Batch loading historic data. For example, you may want to ingest historic log data in a single batch for analysis or for testing. For more information, see [“Loading batches of data” on page 178](#).
- Streaming data from a monitored application. You may want to load data that is streamed from a local or remote server. You can use a number of different tools to stream data.

For more information about the tools that you can use to load data, see [“Data collection tools” on page 149](#).

Note: You must create a Data Source before you configure data loading. For information about creating a Data Source, see [“Data Source creation” on page 309](#).

Limitations

You cannot use IBM Operations Analytics - Log Analysis to index log records that contain non-ASCII characters. If your log records contain non-ASCII characters, the records are not added when you use the IBM Tivoli Monitoring Log File Agent or the Data Collector client. When you use the Data Collector client errors that relate to non-ASCII characters are added to the Generic Receiver log.

Data collection tools

You can load data in batches or you can stream data from local or remote servers. You can use different tools for each.

The following table outlines a number of example scenarios to help illustrate how you use the different tools for different scenarios.

Table 30. Example data loading scenarios	
Example	Tool
I want to load a batch of historic log data to test the environment.	Data Collector client
I want to monitor an application on a remote server.	IBM Tivoli Monitoring Log File Agent
I want to use logstash to monitor log files on a remote server.	logstash
I want to load a batch of historic log data in the JSON format.	Generic Receiver

You can use a number of different methods to load log data into Log Analysis:

Data Collector client

Use the Data Collector client to load a batch of data. This method is the easiest method if you want to load a large log file for historic analysis if you want to test your Log Analysis configuration before you attempting the more complex IBM Tivoli Monitoring Log File Agent configuration. The Data Collector client is not designed for loading data from remote sources. If you want to load a batch of historical data from a remote source, use the IBM Tivoli Monitoring Log File Agent.

For more information, see [“Data Collector client” on page 178](#).

IBM Tivoli Monitoring Log File Agent

Use the IBM Tivoli Monitoring Log File Agent for scenarios where you want to stream log data from your production environment or to stream data from a remote server.

For more information about how to stream data, see [“Streaming data with the IBM Tivoli Monitoring Log File Agent” on page 186](#).

For more information about how to load a batch of data, see [“Loading batches of historic data with the IBM Tivoli Monitoring Log File Agent” on page 183](#) and [“Loading a batch of log files with the LFA” on page 183](#)

logstash

logstash can be used as a method to collect and load data into Log Analysis.

For more information, see [“Streaming data with logstash” on page 218](#).

Generic Receiver

Use the Generic Receiver to load data from the REST interface into Log Analysis.

For more information, see [“Generic Receiver” on page 181](#).

IBM Performance Management OS agent

If you also use IBM Performance Management, you can configure the OS agent to stream data from IBM Performance Management to Log Analysis.

For more information, see [“Streaming data with the IBM Performance Management OS agent” on page 212](#).

This is explained in the *Intended uses of data loading components* table. Log Analysis is installed with an internal version of the IBM Tivoli Monitoring Log File Agent. However, IBM Operations Analytics - Log Analysis can also load data from a separate installation of the IBM Tivoli Monitoring Log File Agent, which is known as an external IBM Tivoli Monitoring Log File Agent.

Table 31. Intended uses of data loading tools				
	Load batch of historic data		Stream data	
Tool	Local	Remote	Local	Remote
Data Collector client	Yes	Yes	No	No
Internal IBM Tivoli Monitoring Log File Agent	Yes	Yes	Yes	Yes
External IBM Tivoli Monitoring Log File Agent	Yes	Yes	Yes	Yes
logstash	Yes	Yes	Yes	Yes
Generic Receiver	Yes	Yes	No	No
IBM Performance Management OS agent	No	No	Yes	Yes

Supported operating systems for data collection

The supported operating systems for remote data collection differ, depending on the tool that you use.

If you want to use the Data Collector client to load a batch of data, the supported operating systems are the same as the ones that are supported by Log Analysis.

However, the supported operating systems might vary depending on tool that you use to stream data. For more information, see the *Supported operating systems for data streaming* table.

Table 32. Supported operating systems for data streaming

Scenario	Feature	Supported operating systems
Use the internal IBM Tivoli Monitoring Log File Agent to stream data	Internal IBM Tivoli Monitoring Log File Agent	See the <i>Requirements for the monitoring agent</i> topic in the documentation for IBM Tivoli Monitoring 6.2.3.1 at: Tivoli Monitoring documentation
Use an external IBM Tivoli Monitoring Log File Agent to stream data	External IBM Tivoli Monitoring Log File Agent	See the <i>Requirements for the monitoring agent</i> topic in the documentation for your version of IBM Tivoli Monitoring at: Tivoli Monitoring documentation

Fix Pack 1 Deploying scalable data collection architecture

Read this section to understand how you can configure your data collection architecture to collect and load data. This architecture can also be easily scaled.

The scalable data collection architecture consists of the following components:

- Apache Kafka
- Logstash
- IBM Tivoli Monitoring Log File Agent
- HAProxy

For an overview of each of these components and how they interact, see “[Components](#)” on page 153.

For more information about the architecture, see “[Data collection architecture](#)” on page 152.

For more information about how to configure these components, see “[Configuring scalable data collection](#)” on page 158.

This scenario assumes that you are using IBM Operations Analytics - Log Analysis 1.3.3 Fix Pack 1.

Fix Pack 1 Planning your deployment

Before you configure your data collection architecture, plan your deployment.

Review the suggested architecture, the description of the components and concepts, and the supported versions of these components. You also need to make yourself familiar with these components and how they work. For more information about the individual components, see the following table.

Prerequisites

Complete the following prerequisites:

- Install IBM Operations Analytics - Log Analysis 1.3.3 Fix Pack 1.
- Install the LFA patch that is part of Fix Pack 1. For more information, see [TCP connection changes to CLOSE_WAIT when Logstash is restarted](#).
- Install the Interim Fix for APAR IV85642. For more information, see <https://ibm.biz/Bd4Uqz>.

Further information

Table 33. Further information for data collection components	
Component	Link
Apache Kafka	http://kafka.apache.org/documentation.html#gettingStarted
Logstash	https://www.elastic.co/guide/en/logstash/current/getting-started-with-logstash.html If you are using the version of Logstash which is bundled with Log Analysis, see “ Streaming data with logstash ” on page 218.
HAProxy	http://www.haproxy.org/download/1.5/doc/configuration.txt
IBM Tivoli Monitoring Log File Agent	For more information, see “ Streaming data with the IBM Tivoli Monitoring Log File Agent ” on page 186.

Fix Pack 1 Data collection architecture

Review the data collection architecture before you deploy and configure the components.

Log file data is generated in the various environments in your organization or data center. This data is collected by the IBM Tivoli Monitoring Log File Agents (LFAs). The LFAs send the data to the Receiver cluster. The Receiver cluster sends it to Apache Kafka where it is queued on the message store. The Sender cluster pulls the data from Apache Kafka and processes it. After this final processing, the data is sent to Log Analysis.

The following graphic summarizes the architecture.

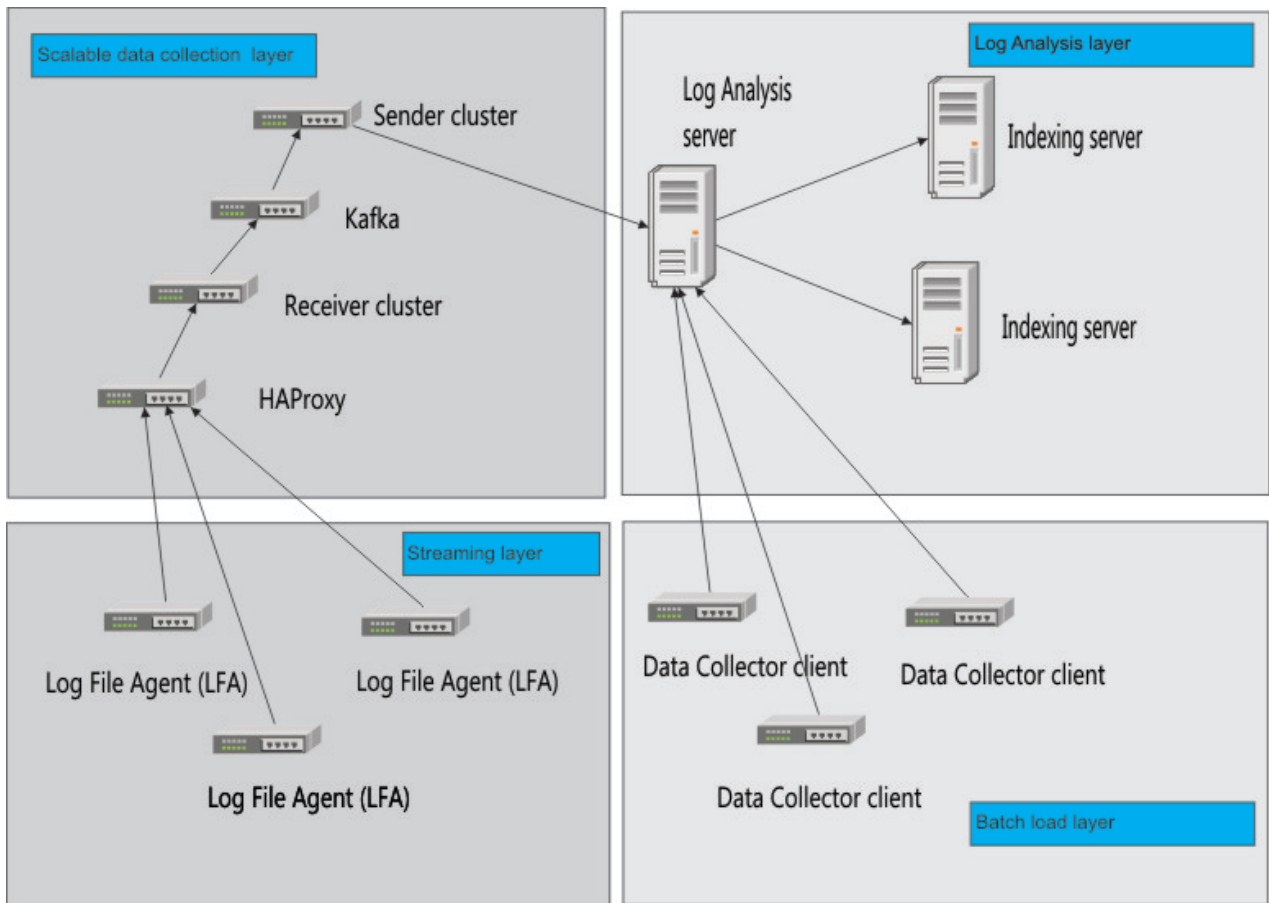


Figure 1. Scalable data collection architecture

Fix Pack 1 Components

Read about the components that make up the scalable data collection architecture.

The scalable data collection configuration consists of the following components.

- IBM Tivoli Monitoring Log File Agent
- Receiver cluster
- Sender cluster
- HAProxy
- Apache Kafka

Fix Pack 1 Apache Kafka cluster and components

Apache Kafka is a high-throughput distributed messaging system that you can use to facilitate scalable data collection.

Apache Kafka is bundled with Log Analysis in the <HOME>/IBM/LogAnalysis/kafka directory.

An installation of Apache Kafka consists of a number of brokers that run on individual servers that are coordinated by an instance of Apache ZooKeeper. You can start by creating a single broker and you can add more as you scale your data collection architecture.

In the scalable data collection architecture, the Receiver cluster writes data to Apache Kafka topics and partitions, based on the data sources. The Sender cluster reads data from Apache Kafka, does some processing and sends the data to Log Analysis.

Apache ZooKeeper

Apache Kafka uses Apache ZooKeeper to maintain and coordinate the Apache Kafka brokers.

A version of Apache ZooKeeper is bundled with Apache Kafka.

Topics, partitions, and consumer groups

The basic objects in Apache Kafka are *topics*, *partitions*, and *consumer groups*.

Topics are divided into partitions. Partitions are distributed across all the Apache Kafka brokers. LFAs

Create one partition for every two physical processors on the server where the broker is installed. For example, if you use eight core processors, create four partitions in the Apache Kafka broker. You specify the number of partitions in your Apache Kafka configuration. For more information, see [“Configuring Apache Kafka brokers”](#) on page 160.

You do not need to manually create topics or consumer groups. You only need to specify the correct values in the configuration for the LFA, Sender, and Receiver clusters. The appropriate topics and partitions are created for you.

In the Receiver configuration, you configure Logstash to receive data from the LFAs and send it to the Apache Kafka brokers. The configuration maps the logical data source attributes that are specified in the LFA configuration to the `topic_id` and `message_key` parameters in Apache Kafka. This configuration ensures that data from each physical data source is mapped to a partition in Apache Kafka. For more information, see [“Configuring the Receiver cluster”](#) on page 163.

In the Sender configuration, you configure Logstash to read data from a specific topic or in the consumer group. This configuration is based on the `group_id` and `topic_id` that you specify. The `topic_id` is the same as the name of the logical data source. For more information, see [“Configuring the Sender cluster”](#) on page 162.

Apache Kafka brokers

The configuration parameters for each Apache Kafka servers are specified in the `<kafka_install_dir>/kafka_2.9.1-0.8.2.2/config/server.properties` file. You need to specify the broker ID, the port, the directory where the log files are stored and the Apache ZooKeeper host name and port in this file. For example:

```
broker.id=1
port=17991
log.dirs=/tmp/kafka-logs-server_0
zookeeper.connect=example.com:12345
```

You can find a sample configuration file in the `<HOME>/IBM/LogAnalysis/kafka/test-configs` directory.

If you want to implement high availability in a production environment, the Apache Kafka server cluster must consist of multiple servers. You can also use these servers to configure replication and retention period. However, when you add new brokers to the cluster, the existing topics are not distributed automatically across new brokers. For more information about how to fix this issue, see <https://kafka.apache.org/081/ops.html>.

Fix Pack 1 **Receiver cluster**

Use a cluster of Logstash servers to receive log file data, separate the incoming traffic, and add tags for further processing

The Receiver cluster can include one or more instances of Logstash.

As part of deploying scalable data ingestion architecture, you implement a basic configuration of Logstash that can receive log data and store it in a reliable data store without doing any further processing. By limiting the amount of processing, you ensure that you avoid increased processor usage in your data collection pipe line by not blocking the LFAs when they send data.

You can run multiple instances of Logstash with the same configuration in a cluster, which sits behind the high availability proxy servers.

You need to decide whether you want to process single or multiple line log files or both. You need to use different instances of Logstash for both types of log file. For more information, see [“Single and multi-line log files”](#) on page 156.

Fix Pack 1 **Sender cluster**

Configure the Sender cluster to read data from Apache Kafka, process it and send it to Log Analysis.

The Sender cluster can include one or more instances of Logstash.

Before you deploy the Sender cluster, decide whether you want to process single or multiple line log files or both. You need to use different instances of Logstash for both types of log file. For more information, see [“Single and multi-line log files”](#) on page 156.

You can configure the Logstash to parse and annotate data as part of your scalable data collection architecture. For more information, see [“Configuring the Sender cluster”](#) on page 162.

Fix Pack 1 **HAProxy**

Use HAProxy to act as a load balancer and to provide high availability for the Logstash instances in your Receiver cluster.

HAProxy is an open source tcp or http load balancer. You add this component to add high availability and load balancing capabilities to your data collection architecture. You can also use it to protect the LFA from any changes that are made in the Receiver cluster.

HAProxy is an operating system package that you need to install with a tool such as "yum" or "yast".

For more information about HAProxy, see <http://www.haproxy.org/#docs>

After you configure it, it runs as a service in your operating system. The configuration file for HAProxy is `haproxy.cfg`.

If you are monitoring log files with multiple lines, you must configure HAProxy with affinity for agents that is based on the IP information. For more information, see [“Single and multi-line log files”](#) on page 156.

You can also modify your HAProxy configuration to enable the statistics page. For more information, see <http://www.haproxy.org/download/1.5/doc/configuration.txt>.

Fix Pack 1 **IBM Tivoli Monitoring Log File Agent (LFA)**

Use the IBM Tivoli Monitoring Log File Agent (LFA) to collect the log file data and add the required metadata before sending it through HAProxy to the Logstash instances in the Receiver cluster.

Before you can use the LFA, you must apply the required patch to all your LFA instances. For more information, see [“Updating the LFA for scalable data collection”](#) on page 166.

You can create new LFAs or you can integrate existing LFAs. For more information about how to deploy the LFA that is bundled with Log Analysis, see [“Streaming data with the IBM Tivoli Monitoring Log File Agent”](#) on page 186.

To use the LFA, you need to configure it for scalable data collection. For more information, see [“Configuring the LFA”](#) on page 167.

After you configure the LFA, you need to create new custom data sources in Log Analysis.

To facilitate efficient processing, create LFA subnodes. For more information, see [“Configuring LFA subnodes”](#) on page 199.

Fix Pack 1 **Data Sources**

To facilitate scalable data collection, you must create new data sources.

For more information about how to create a data source for scalable data ingestion, see [“Configuring data sources in Log Analysis”](#) on page 167.

For scalable data collection, create a logical data source into which data is streamed from multiple physical data sources, which are defined in the LFA configuration.

For an example of how to create a data source for scalable data collection, see [“Data Source configuration example” on page 168](#).

You need to ensure that the correct data sources are specified in the configuration of the other components such as Logstash. The information that is required to do so is included in the relevant sections of this documentation.

Logical and physical data sources

It is also important that you understand the difference between logical and physical data sources.

A logical data source is a data source that you create in Log Analysis. A physical data source is source of data in your environment. In the context of scalable data collection, the physical data sources are defined by you in the LFA configuration. To add a new physical data source, configure the LFA to monitor a new file.

The log file types that you stream in a logical data source must be the same for each physical data source. For example, you cannot use the same logical data source to stream WAS and DB2 files.

You can use Log Analysis to create a logical data source that you can use to stream data from multiple, physical sources of data. For example, if you have multiple servers where you store log files of the same type, you can use a data source in Log Analysis to stream data from the servers.

For an example of how to use a logical data source to stream data from multiple physical data sources, see [“LFA example: Streaming data from a single remote host” on page 202](#).

To increase the volume of data, you can create a new physical or logical data source. For more information, see [“Increasing the volume of data” on page 178](#).

Fix Pack 1 Supported component versions

The scalable data ingestion architecture uses a number of components. Ensure that you are using the correct version.

For more information about the supported versions of the software components that make up the scalable data collection architecture, see the following table.

Table 34. Version information for components	
Component	Version
IBM Operations Analytics - Log Analysis	IBM Operations Analytics - Log Analysis 1.3.3 Fix Pack 1
Apache Kafka	0.8.2.2
Logstash	2.2.1
HAProxy	1.5.4

Fix Pack 1 Single and multi-line log files

When you deploy the scalable data collection architecture, you need to consider whether you want to load single or multi-line log files or both.

You need to modify your Sender and Receiver clusters and your HAProxy configuration to facilitate multi-line log files.

You can run multiple instances of Logstash on the same system to scale to available CPU resources. For more information about how to tune these resources, see [“Tuning considerations” on page 158](#).

Sender and Receiver clusters

If you decide to run both types of log file, you need to configure separate instances of Logstash in both the Sender and Receiver clusters.

To process single-line log files, install and configure a single instance of Logstash. Ensure that this instance runs with multi-threading enabled. To enable multi-threading, use a `-w` argument to run the instance.

To process multi-line log files, install and configure one or more instances of Logstash. Ensure that each instance of Logstash runs with in single-threading enabled. This configuration helps to ensure that the lines in the log file are processed sequentially, maintaining order within and between records.

To process multi-line log files, you also need to include the `"metadata_fields"` section in the LA Logstash plug-in configuration. This configuration ensures that records are segregated and that batches are created at the physical data source level, preserving the order of the lines in the log files.

HAProxy

To monitor multi-line log files, configure HAProxy with affinity for agents that is based on the IP information. To configure HAProxy, edit the `haproxy.cfg` file.

Fix Pack 1 Sizing hardware for scalable data collection

Before you implement this architecture, consider the hardware requirements for the components that make up your scalable data collection architecture.

The servers can be metal or virtualized. If you use a virtualized server, ensure that hyper threading is disabled and that physical cores are allocated and reserved.

For an example deployment, see [“Example sizing” on page 157](#).

This document outlines the hardware requirements for your scalable data collection components. For more information about the general hardware requirements for Log Analysis, see [Hardware and software requirements](#).

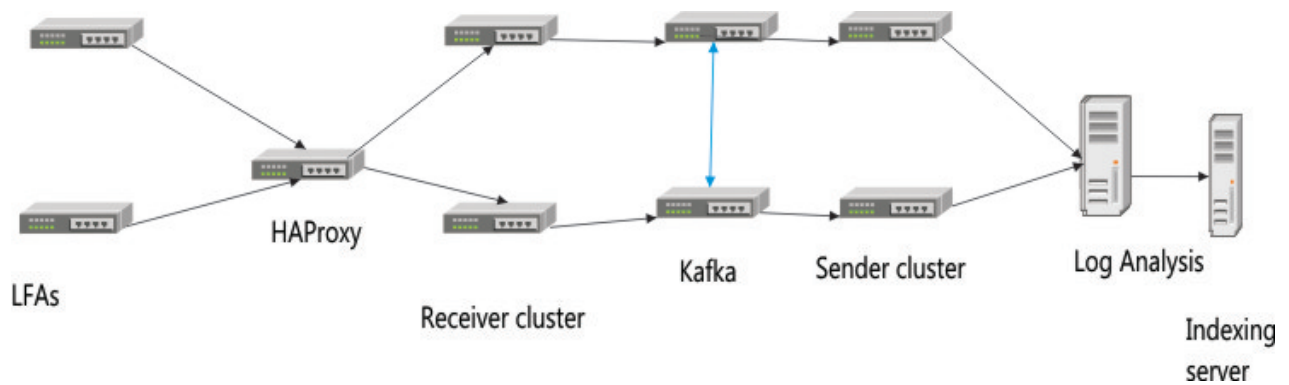
The following table summarizes the hardware requirements for each server and the recommended maximum number of servers.

<i>Table 35. Hardware requirements</i>				
Component	Daily data ingestion per server in GB	Processors per server	RAM per server in GB	Recommended number of servers
IBM Tivoli Monitoring Log File Agent	200	8	4	2
HAProxy	400	2	4	1
Receiver cluster	200	8	8	2
Apache Kafka	200	4	8	2
Sender cluster	200	8	8	2
Log Analysis server	400	8	16	1
Indexing Engine server	400	8	48	1

Fix Pack 1 Example sizing

This example lists the hardware requirements for an example deployment where 400 GB of data is streamed into Log Analysis daily.

The following graphic illustrates this deployment:



The following lists the servers and components in a typical deployment. It assumes that you are streaming 400GB of data daily in total. Each Apache Kafka broker streams an aggregated 400 GB of data daily:

LFA servers

Two servers. 200 GB daily ingestion per server. Each server has eight physical processors and 4 GB of RAM.

HAProxy server

One server. The server has two physical processors and 4 GB of RAM.

Receiver cluster (Logstash instance)

Two servers. 200 GB daily data ingestion per server. Each server has eight physical processors and 8 GB of RAM.

Sender cluster (Logstash instance)

Two servers. 200 GB daily data ingestion per server. Each server has eight physical processors and 8 GB of RAM.

Log Analysis server

One server. The server has 8 physical processors and 16 GB of RAM.

Indexing Engine server

One server. The server has 8 physical processors and 48 GB of RAM.

Fix Pack 1 Tuning considerations

You can tune the components in your data collection architecture to ensure that data is processed efficiently.

In most cases, the default memory and other performance-related settings are sufficient for scalable data collection. For more information about tuning a component such as Logstash or Apache Kafka, see <https://developer.ibm.com/itoa/docs/log-analysis/performance-tuning>.

You can also make the following modifications to the standard configuration:

- To minimize the number of connections that are used by the LFA to connect to HAProxy, create LFA subnodes. For more information, see [“Configuring LFA subnodes” on page 199](#).

Fix Pack 1 Configuring scalable data collection

Before you can use the scalable data collection architecture, you need to install and configure the components.

Before you complete the steps below, plan your deployment. For more information, see [“Planning your deployment” on page 151](#).

To configure scalable data collection, complete the following steps:

1. Install Logstash and create the required utility scripts.
2. Install and configure Apache Kafka.
3. Configure the instances of Logstash in the Sender cluster.

4. Configure the instances of Logstash in the Receiver cluster.
5. Install and configure HAProxy.
6. Configure the LFA.
7. Create the logical data sources in Log Analysis.

Fix Pack 1 Sample files for scalable data collection

You can find sample scripts in the <HOME>/IBM/LogAnalysis/kafka/test-configs folder.

The files are listed in the following table:

<i>Table 36. Sample files for scalable data collection</i>	
File	Description
HAProxy	
haproxy.cfg	HAProxy configuration file.
Apache Kafka	
server_0.properties	Apache Kafka broker configuration file.
zookeeper.properties	Apache ZooKeeper properties file,
LFA	
DB2-Diag.conf	LFA configuration file for DB2® logs.
DB2-Diag.fmt	LFA format file for DB2 logs.
WASSystemOut.conf	LFA configuration file for WebSphere Application Server logs.
WASSystemOut.fmt	LFA format file for WebSphere Application Server logs.
Logstash	
receiver-logstash1.conf	Configuration file for the receiver-logstash1 Logstash receiver instance.
receiver-logstash1-util.sh	Script for running the receiver-logstash1 Logstash receiver instance.
receiver-logstash2.conf	Configuration file for the receiver-logstash2 Logstash receiver instance.
receiver-logstash2-util.sh	Script for running the receiver-logstash2 Logstash receiver instance.
SCALAPATTERNS	Pattern file. This file specifies the required patterns for parsing data sent from the LFA. Copy this file to the patterns directory in your installation. For example <logstash_install>/Logstash/patterns. Add the path to this directory to your Logstash configuration.
sender-logstash-multi_line.conf	Configuration file for the Logstash sender instance for multiple line log files.
sender-logstash-multi_line-util.sh	Script for running the Logstash sender instance for multiple line log files.
sender-logstash-single_line.conf	Configuration file for the Logstash sender instance for single line log files.

Table 36. Sample files for scalable data collection (continued)

File	Description
sender-logstash-single_line-util.sh	Script for running the Logstash sender instance for single line log files.

Fix Pack 1 **Configuring Apache Kafka brokers**

To implement scalable data loading, you must configure at least one Apache Kafka broker.

About this task

Apache Kafka is bundled with Log Analysis. The sample configuration files for Apache Kafka are in the <HOME>/IBM/LogAnalysis/kafka/test-configs/kafka-configs directory.

Create one partition per topic for every two physical processors on the server where the broker is installed. For example, if you use eight core processors, create four partitions per topic in the Apache Kafka broker.

To implement High Availability messaging, you must create multiple brokers on different servers. To set up multiple brokers, update the configuration files as described in step 3. Set it to the same Apache ZooKeeper server and update the broker ID so that it is unique for each broker.

Procedure

- Copy the kafka_2.9.2-0.8.2.1.tgz to an appropriate directory on the server where you want to install Apache Kafka.
- Extract the kafka_2.9.2-0.8.2.1.tgz file.
- Update the Apache Kafka configuration files.
 - To add the directory path and the port number that is used by Apache ZooKeeper, edit the following properties in the zookeeper.properties file:

```
dataDir=<zookeeper_data_directory>
clientPort=<zookeeper_port>
```

where <zookeeper_data_directory> is the directory where the snapshot of Apache ZooKeeper is stored. <zookeeper_port> is the port used by Apache ZooKeeper.

- To update the key parameters in the Apache Kafka broker, edit the following key properties in the server.properties file:

```
broker.id=0
port=<kafka_broker_port>
log.dirs=/tmp/kafka-logs-server0
num.partitions=<num_partitions>
zookeeper.connect=<zookeeper_server>:<zookeeper_port>
```

where <kafka_broker_port> is the port used for Apache Kafka. <zookeeper_server> and <zookeeper_port> are the host name and port used by Apache ZooKeeper. <num_partitions> is half the number of physical processors on the server.

- Start Apache ZooKeeper.

To start it in console mode, enter the following command:

```
<Kafka_home>/bin/zookeeper-server-start.sh config/zookeeper.properties
```

- Start the Apache Kafka broker. Enter the following command:

```
<Kafka_home>/bin/kafka-server-start.sh -daemon config/server0.properties
```

To stop the broker, enter the following command:

```
<Kafka_home>/bin/kafka-server-stop.sh
```


To stop Apache ZooKeeper, enter the following command:

```
<Kafka_home>/bin/zookeeper-server-stop.sh
```

Fix Pack 1 Installing Logstash and the utility script

To implement a scalable data collection architecture, install and configure both a Sender and Receiver cluster that is composed of a number of Logstash instances.

Before you begin

Use the utility that is provided in Log Analysis to install Logstash on the remote servers for the sender and receiver clusters. For more information, see [“Installing logstash on a remote node”](#) on page 220.

Procedure

1. To create and configure the configuration files:
 - a. Copy the default configuration file, `<logstash_install>/Logstash/logstash-2.2.1/logstash-scala/logstash/config/logstash-scala.conf`.
 - b. Create a configuration file for each Logstash instance in your Receiver cluster. Rename the copied files to match your deployment. For example `receiver-logstash1.conf`.
 - c. Configure the Receiver file. For more information, see [“Configuring the Receiver cluster”](#) on page 163.
 - d. Create a configuration file for each Logstash instance in your Sender cluster. Rename the copied files to match your deployment. For example `sender-logstash1.conf`.
 - e. Configure the Sender file. For more information, see [“Configuring the Sender cluster”](#) on page 162.
 - f. Save the files in the `<logstash_install>/Logstash/logstash-2.2.1/logstash-scala/logstash/config/` directory.
2. Create a script to manage Logstash:
 - a. Copy the `<logstash_install>/utilities/logstash-util.sh` script.
 - b. Rename it to `receiver-logstash1.sh`
 - c. Update the following parameters to match the configuration file that you created in step 1. For example:

```
logstash_conf="${<logstash_install>}/logstash-scala/logstash/  
config/receiver-logstash1.conf"
```

- d. Update the log file for the particular instance. For example:

```
logstash_log="${<logstash_install>}/../logs/receiver-logstash1.log"
```

- e. Update the `find_logstash_process()` function to search for your configuration file:

```
find_logstash_process () {  
    PIDTEMP=`ps ux | grep logstash | grep /vendor/jruby  
    | grep config/receiver-logstash1.conf | grep java | awk '{ print $2 }'`
```

- f. To create a script to manage the Sender cluster, repeat the previous steps, changing `receiver-logstash1` to `sender-logstash1`.

Results

You can now use the following commands to start and stop your Receiver cluster: instances:

```
<logstash_install_dir>/utilities/receiver-logstash1.sh start | stop | status
```

You can now use the following commands to start and stop your Sender cluster: instances:

```
<logstash_install_dir>/utilities/sender-logstash1.sh start | stop | status
```

Fix Pack 1 **Configuring the Sender cluster**

To implement a scalable data collection architecture, install and configure a cluster of Logstash servers to send data to Log Analysis.

Before you begin

You need to decide whether you want to process single or multiple line log files or both. You need to use different instances of Logstash for both types of log file. For more information, see [“Single and multi-line log files”](#) on page 156.

Install Logstash on the servers and create the required utility script. For more information, see [“Installing Logstash and the utility script”](#) on page 161.

Procedure

1. Stop the Logstash instance.
2. Edit the `<logstash_install>/logstash/logstash-2.2.1/logstash-scala/logstash/config/<logstash_instance>.config` file.
3. To ensure that the Logstash instance can read data that is sent from the Apache Kafka cluster, add the following information in the `input` section:

```
input {
  kafka {
    zk_connect => "<zookeeper_server:port>"
    group_id => "<group_id>"
    topic_id => "<topic_id>"
    consumer_threads => <num_partitions>
    consumer_restart_on_error => true
    consumer_restart_sleep_ms => 100
    decorate_events => true
  }
}
```

where `<zookeeper_server:port>` is the Apache ZooKeeper server and port. The `group_id` identifies the groups of consumers. `<num_partitions>` is half the number of physical processors on the server.

The Sender Logstash instance reads data from the topic or partition that you specify in the `input` section. The important parameters are `group_id`, `topic_id`, and `consumer_threads`.

The `topic_id` identifies the topic that consumes the messages. Use the same name for the `topic_id` and the `group_id`. For example:

```
input {
  kafka {
    zk_connect => "zookeeper1.example.com:17981"
    group_id => "MY_WAS_SystemOut"
    topic_id => "MY_WAS_SystemOut"
    consumer_threads => 4
    consumer_restart_on_error => true
    consumer_restart_sleep_ms => 100
    decorate_events => true
  }
}
```

Ensure that the `consumer_threads` parameter matches the number of partitions that are specified in the Apache Kafka. The `consumer_threads` parameter specifies the number of consumers that are created in a consumer group. Each thread or consumer maps to a partition for the specified topic or logical data source. This ensures that data is processed concurrently. If you specify fewer partitions than consumer threads, some threads remain idle while they wait for an available partition.

If you are running multiple Logstash servers in your Receiver cluster, ensure that no two instances of Logstash are reading data from the same `topic_id`. Each instance must read data from a different `topic_id`. The `topic_id` is specified in the `input` section of the Apache Kafka configuration file.

4. To ensure that the Logstash instance tags the log files with the required data before it is sent to Log Analysis, update the `filter` section. This example adds host and path information that Log Analysis uses to stream data into the appropriate data source:

```

filter {
  if "grok_lfa" in [tags] {
    mutate {
      replace => { "host" => "%{LFA_SITE}_%{LFA_MODULE}" }
      add_field => { "path" => "%{LFA_TYPE}" }
    }
  }
}

```

The `grok_lfa` tag is added to the message by the receiver Logstash cluster. "host" and "path" are required tags. They are used to identify the data source in Log Analysis. The tags are created and added to the event by the Receiver Logstash cluster during processing.

5. Review the output section and ensure that everything is correct. For example, if you install Logstash with the remote tool, the output section is as follows:

```

output {
  scala {
    scala_url => "https://<ioala-server>:9987/Unity/DataCollector"
    scala_user => "unityadmin"
    scala_password => "{aes}E2EC7F376F4837F2AA041A7364CC1F9A"
    scala_keystore_path => "<Logstash_install_location>/store/unity.ks"
    batch_size => 500000
    idle_flush_time => 5
    sequential_flush => true
    num_concurrent_writers => 20
    use_structured_api => false
    disk_cache_path => "<Logstash_install_location>/Logstash/cache-dir"
    date_format_string => "yyyy-MM-dd'T'HH:mm:ssX"
    log_file => "<Logstash_install_location>/Logstash/
logs/scala_ml_logstash.log"
    log_level => "info"
  }
}

```

For multi-line log files, add the `metadata_fields` section to the output plug-in after the `log_level` line. Specify the host name and file path for each logical data source which consumes multi-line log files in the `<DataSource_Host1>` and `<DataSource_Path1>`. Repeat this for each logical data source.

For example:

```

metadata_fields => {
  "<DataSource_Host1>@<DataSource_Path1>" => {
    "field_names" => "resourceID"
    "field_paths" => "resourceID"
  }
  "<DataSource_Host2>@<DataSource_Path2>" => {
    "field_names" => "resourceID"
    "field_paths" => "resourceID"
  }
}

```

`<DataSource_Host1>` is the host name as defined in the logical data source. `<DataSource_Path1>` is the file path as defined in the logical data source. `field_name` specifies the name of the field that is added to the event log. `field_path` is the path in the event message.

This configuration specifies a field in the log event that can be used to uniquely identify the physical data source. This example uses the `resourceID` field.

6. Start the Logstash cluster.

Fix Pack 1 **Configuring the Receiver cluster**

To implement a scalable data collection architecture, install and configure a cluster of Logstash servers to receive data from the LFA and write it to Apache Kafka.

Before you begin

Install Logstash on the remote servers and create the required utility script. For more information, see [“Installing Logstash and the utility script” on page 161](#).

About this task

You must create at least one Logstash server to act as a receiver. In a production environment, you need to use more than one instance of Logstash in a cluster.

You need to complete this task for each instance of Logstash in your cluster.

Procedure

1. Stop the Logstash server.
2. Edit the `<logstash_install>/logstash/logstash-2.2.1/logstash-scala/logstash/config/<logstash_instance>.config` file.
3. To allow Logstash to receive data from the LFA, add the receiver port information to the `input` section:

```
input {
  tcp {
    port => <receiver_port>
    type => "lfa"
  }
}
```

The type is referenced in the `filter` section to add the required fields and process the message before the message is sent to the Apache Kafka server.

4. To process messages from the LFA and add the required fields, update the `filter` section:

```
filter {
  if [type] == "lfa" {
    grok {
      patterns_dir => "<Logstash_install_location>
/<patterns_directory>"
      match => [ "message", "%{LFAMESSAGE}" ]
      add_tag => ["grok_lfa"]
    }
    if "grok_lfa" in [tags] {
      mutate {
        replace => ["message", "%{LFA_ORIG_MSG}"]
        add_field => [ "datasource",
"%{LFA_SITE}_%{LFA_MODULE}_%{LFA_TYPE}" ]
        add_field => [ "resourceID",
"%{LFA_HOSTNAME}_%{LFA_LOGNAME}_1" ]
      }
    }
  }
}
```

where `<Logstash_install_location>` is the directory where you installed Logstash. `<patterns_directory>` is the directory where you stored the patterns used by Logstash.

The "datasource" field is used to create the topic in Apache Kafka. The "resourceID" field is used to map the data to partitions.

The filter section matches each agent specific log record with an appropriate pattern. You can find an example file called SCALAPATTERNS in the `<HOME>/IBM/LogAnalysis/utilities/kafka/test-configs` directory. Copy this file to the patterns directory in your Logstash instances. For example, the patterns directory can be `<logstash_install>/Logstash/patterns`.

The filter also extracts the fields that are required to define the logical and physical data source. The filter uses these fields to add two further fields, `datasource` and `resourceID` to the message. These fields are used to send the data to the appropriate partition or topic that is specified in the output section of your Apache Kafka configuration.

5. Create a new patterns file or use the `<HOME>/IBM/LogAnalysis/kafka/test-configs/SCALAPATTERNS` sample file. Save the file in the patterns directory. The following example is based on this file. It is broken up over a number of lines. When you create your own pattern, the code must be entered as a single line.

```
LFAMESSAGE
<START>.*type='%{DATA:LFA_TYPE}';
```

```

text='%{DATA:LFA_ORIG_MSG}';
RemoteHost='%{DATA:LFA_REMOTE_HOST}';
site='%{DATA:LFA_SITE}';instance='%{DATA:LFA_INSTANCE}';
hostname='%{DATA:LFA_HOSTNAME}';
cluster='.*';module='%{DATA:LFA_MODULE}';
env='%{DATA:LFA_ENVIRONMENTNAME}';
logpath='%{DATA:LFA_LOGNAME}';
functional='%{DATA:LFA_FUNCTIONALNAME}';END
LALFAMESSAGE
<START>.*text='%{DATA:LFA_OrigMsg}';
RemoteHost='.*';
hostname='%{DATA:LFA_HOSTNAME}';
env='%{DATA:LFA_ENVIRONMENTNAME}';
logpath='%{DATA:LFA_LOGNAME}';END

```

6. To send messages that use the `grok_lfa` tag to the Apache Kafka cluster, add the following information to the output section.

```

output {
  if ("grok_lfa" in [tags]) and ! ("_grokparsefailure" in [tags]) {
    kafka {
      bootstrap_servers => "<kafka_broker1>:<broker1_port>,<kafka_broker2>:<broker2_port>"
      topic_id => "%{datasource}"
      message_key => "%{resourceID}"
    }
  }
}

```

7. Start the Logstash instance.
 8. Set up extra receiver configurations for availability and failover if required.
- For more information, see <https://www.elastic.co/guide/en/logstash/current/getting-started-with-logstash.html>.

Example

The following example processes events where the type is `lfa` and matches these to the patterns. The `datasource` and `resourceID` are also added based on the metadata in the event.

```

filter {
  if [type] == "lfa" {
    grok {
      patterns_dir => "home/la/logstash/patterns"
      match => [ "message", "%{LFAMESSAGE}" ]
      add_tag => ["grok_lfa"]
    }
  }
  if "grok_lfa" in [tags] {
    mutate {
      replace => ["message", "%{LFA_ORIG_MSG}"]
      add_field => [ "datasource",
"%{LFA_SITE}_%{LFA_MODULE}_%{LFA_TYPE}" ]
      add_field => [ "resourceID",
"%{LFA_HOSTNAME}_%{LFA_LOGNAME}_1" ]
    }
  }
}

```

The output section writes data to the Apache Kafka cluster while mapping the data source to the `topic_id` parameter. This configuration ensures that one topic is created for the logical data source. It also ensures that data from each physical data source is written to the same partition within the topic. For example:

```

output{
  if ("grok_lfa" in [tags]) and ! ("_grokparsefailure" in [tags]) {
    kafka {
      bootstrap_servers => "kafkabroker1.example.com:17911,
kafkabroker2.example.com:17911"
      topic_id => "%{datasource}"
      message_key => "%{resourceID}"
    }
  }
}

```

Fix Pack 1 Installing and configuring HAProxy

HAProxy is a native Linux operating system package that you can use to help you to balance loads, facilitate high availability, and provide failover support in case one of the Logstash servers is not available.

About this task

For more information about HAProxy, see <http://www.haproxy.org/#docs>

Procedure

1. To install HAProxy, use a system tool like "yum" or "yast" . The package is part of both Red Hat Enterprise for Linux and SuSE Linux Enterprise server.
2. To add the LFA cluster and specify the receiver Logstash instance details, edit the `haproxy.cfg` file and add the required information as follows:

```
listen LFA_Cluster <haproxy_hostname>:<haproxy_port>
    mode tcp
    balance source
    hash-type consistent
    server receiver-logstash1
    <logstash1-host>:<logstash1-port>
    check inter 1s fall 2 rise 3
    server receiver-logstash2 <logstash2-host>:<logstash2-port>
    check inter 1s fall 2 rise 3
```

where `<haproxy_hostname>:<haproxy_port>` is the host name and port used by HAProxy. `<logstash1-host>:<logstash1-port>` is the host name and server used by the first instance of the receiver Logstash cluster. `<logstash2-host>:<logstash2-port>` is the host name and server used by the second instance. Add each instance as required.

If you are monitoring log files with multiple lines, you must configure HAProxy with affinity for agents that is based on the IP information. This is specified by the following lines:

```
balance source
hash-type consistent
```

3. To restart HAProxy, enter the following command:

```
service haproxy restart
```

Fix Pack 1 Updating the LFA for scalable data collection

Before you can use the LFA, install the required patch.

About this task

To facilitate scalable data collection, you need to deploy the patch for APAR IV812217 for IBM Tivoli Monitoring version 06.30.05.00 and IBM Tivoli Monitoring Log File Agent version 06.30.00.04. This patch is bundled with IBM Operations Analytics - Log Analysis 1.3.3 Fix Pack 1.

Procedure

1. Extract the `6.3.0-TIV-ITM-FP0005-IV81217.zip` file.
2. To install the patch, follow the instructions in the readme file.
3. Repeat steps one and two for each LFA instance that you want to include in your scalable data collection architecture.

What to do next

Restart the LFA instances.

Fix Pack 1 **Configuring the LFA**

Configure the LFA to add the metadata and attributes such as type, site or module, which Log Analysis uses to map to the logical data source and Resource ID. The Resource ID is used to uniquely identify the physical data source.

Before you begin

You can install a new LFA or you can integrate an existing LFA. For more information about installing and configuring the LFA, see [“Streaming data with the IBM Tivoli Monitoring Log File Agent” on page 186](#).

Procedure

1. To add the required information to the configuration, edit the configuration or `.conf` file. The required parameters are:

```
LogSources=<log_path>
FileComparisonMode=CompareByAllMatches
ServerLocation=<haproxy_server_host>
ServerPort=<haproxy_port>
```

where `<log_path>` is the path to log files. `<haproxy_server_host>` and `<haproxy_port>` are the host name and proxy port used by HAProxy.

2. To populate the fields that are required for each event, update the format or `.fmt` file with the specific values that are required for your deployment. For example:

```
REGEX AllRecords
(.* )
hostname LABEL
-file FILENAME
RemoteHost DEFAULT
logpath PRINTF("%s",file)
type SystemOut
instance testInstance
cluster NONE
module WAS
env DEV
functional NONE
site PUNE
text $1
END
```

3. Restart the LFA. For example, if you are using the internal LFA, enter the following command:

```
<LFA_install_directory>/utilities/lfautil.sh
```

What to do next

To facilitate efficient processing, create LFA subnodes. For more information, see [“Configuring LFA subnodes” on page 199](#).

Fix Pack 1 **Configuring data sources in Log Analysis**

After you configure the LFA, create new custom data sources in Log Analysis to facilitate scalable data collection.

About this task

Create a logical data source that streams data from multiple physical data sources.

For an example of how to create a data source for scalable data collection, see [“Data Source configuration example” on page 168](#).

For more information about creating a data source in Log Analysis, see [“Data Source creation” on page 309](#).

Procedure

1. Use the **unityadmin** user to log in to Log Analysis.
2. Create a data source with the values outlined in the following table.

Table 37. Data source for scalable data collection	
Field	User input
Type	Select Custom .
Host name	Enter a generic name. For example MY_WAS .
File path	Enter a generic file path, for example, SystemOut. This path does not need to correspond to any actual path.
Type	Select the correct type for your log files. The type defaults to WASSystemOut when you include a string like SystemOut in the file path.
Name	Enter a suitable name. For example MY_WAS_SystemOut..

3. Save the data source.

Fix Pack 1 Data Source configuration example

This example helps you to include the required information in your data source configuration.

In it, the user has three attributes, which are used to define the logical data source in Log Analysis. The attributes and possible value are listed in the following table.

Table 38. LFA attributes and values for logical data source	
Attribute	Value
site	PUNE
module	WAS
type	SystemOut

Log Analysis uses these attributes to construct the logical data source name. You create a data source in Log Analysis to capture these attributes. An example data source configuration is outlined in the following table:

Table 39. Log Analysis data source attributes	
Field	User entry
Name	PUNE_WAS_SystemOut
Type	Custom
Host name	PUNE_WAS
File path	SystemOut

The LFA format file that gathers and sends this information to the Receiver cluster is as follows:

```
REGEX AllRecords
(.*)
hostname LABEL
-file FILENAME
//-file /home/user/LoadGenerator/load-dump/db2diag-2.log
RemoteHost DEFAULT
logpath PRINTF("%s",file)
type SystemOut
```



```
instance testInstance
cluster NONE
module WAS
env DEV
functional NONE
site PUNE
text $1
END
```

Fix Pack 1 **Configuring logging for troubleshooting**

To ensure that you can troubleshoot issues with Apache Kafka or Logstash, enable logging for those components.

About this task

Use the logging function only in Proof of Concept or test systems. For production scale volumes of data, disable logging.

Procedure

Configuring logging for Logstash

1. To enable logging for Logstash, add the following code to the output section of the configuration:

```
output {
  file {
    path => "<path_to_log_directory>/rubydebug.log"
    codec => rubydebug
  }
}
```

where *<path_to_log_directory>* is the path to the directory where you want to store the log files.

This code enables the `rubydebug` codec, which logs all the messages that Logstash receives. You can use these logs to check to see whether the appropriate tags are applied and if the processing is correct.

2. To generate log files for the Log Analysis Logstash plug-in, add the full path for the log file to the following property in the Log Analysis output plug-in section of your Sender configuration:

```
log_file => "<log_file_and_path>"
```

For example:

```
log_file => "/data/loala/Logstash/logs/scala_logstash.log"
log_level => "warn"
```

If you do enable logging in production systems, the plug-in needs to be configured for the warning level.

Configuring Insight Packs

If you want to integrate your Insight Packs with the scalable data collection architecture, you need to update the scalable data collection configuration.

Insight Packs use various different tools and methods to collect and annotate data. Data can be collected by the IBM Tivoli Monitoring Log File Agent or Logstash. Annotation can be done in Logstash or in Log Analysis. Some Insight Packs use the LFA to collect data and annotate it in Logstash before it is sent to Log Analysis.

The Insight Packs that are available for Log Analysis can be grouped into the following types based on the data collection tools that they use and where the data is annotated:

IBM Tivoli Monitoring Log File Agent (LFA) collects data and Log Analysis annotates it

These types of Insight Pack use local or remote instances of the LFA to stream log file data to Log Analysis. The data is annotated according to the logic defined in the Insight Pack on the Log Analysis server. The WebSphere Application Server Insight Pack is one example of this type.

For more information, see [“Configuring Insight Packs that use the LFA to stream data and Log Analysis to annotate it”](#) on page 172.

LFA collects data and Logstash annotates it

These types of Insight Packs use both the LFA and Logstash. The LFA sends the log file data to Logstash, where it is annotated according to the logic defined in the Logstash configuration and sent to Log Analysis. The SAP HANA Insight Pack is one example of this type.

For more information, see [“Configuring Insight Packs that use LFA to load data and Logstash to annotate it”](#) on page 170.

Logstash collects data and annotates it

These types of Insight Packs use Logstash to stream data to Log Analysis. The data is annotated according to the logic defined in the Logstash configuration for the Insight Pack. The Oracle DB Alert Insight Pack is one example of this type.

For more information, see [“Configuring Insight Packs that use Logstash to stream data and annotate it”](#) on page 175.

Custom data collection agent collects data and Log Analysis annotates it

These types of Insight Packs use custom built agents to collect data. The data is sent to Log Analysis through the REST API. The data is annotated according to the logic defined in the Insight Pack on the Log Analysis server. The NetApp OnTap Insight Pack is one example of this type. This type of Insight Pack is not supported by the scalable data collection architecture.

If you do not integrate your Insight Packs with the scalable data collection architecture, they will continue to function. However, they will not enjoy the benefits of the architecture.

Configuring Insight Packs that use LFA to load data and Logstash to annotate it

To integrate the scalable data collection architecture with any Insight Packs that use the LFA and Logstash to stream data, annotate it and send it to Log Analysis, you need to adapt the configuration to make it compatible with scalable data collection.

Before you begin

Configure the scalable data collection architecture. For more information, see [“Configuring scalable data collection”](#) on page 158.

About this task

This configuration is intended for Insight Packs that use the LFA to stream data to Logstash where it is processed and sent to Log Analysis. For example, the Generic Receiver Insight Pack, which is part of the Log Analysis health pack.

You copy the Logstash configurations that are included in the Insight Packs to the Sender and Receiver cluster configurations in your scalable data collection architecture. First, copy the `input` section from the Logstash configuration file for the Insight Pack to the Receiver configuration file. You also need to add some configuration for processing the metadata fields and sending messages to the topics and partitions in Apache Kafka. Next, you copy the filtering and processing logic from the Logstash configuration for the Insight Pack to the Sender cluster configuration. These configuration settings help process the log files and forward them to Log Analysis.

Procedure

1. Update the LFA configuration for the Insight Pack so that it can send data to HAProxy or the Receiver cluster.

Add the server and port information for the Receiver cluster instance or the HAProxy to the LFA configuration or `.conf` file. For example:

```
ServerLocation=<HAProxy_or_receiver_cluster_server>  
ServerPort=<HAProxy_or_receiver_cluster_port>
```

For more information, see [“Configuring the LFA” on page 167](#).

2. Configure the tcp input section of the Receiver cluster configurations so that it can receive data that is sent by Logstash.

For example:

```
input {
  tcp {
    port => <Logstash_Port>
    type => "LFA"
    codec => line { charset => "US-ASCII" }
  }
}
```

For more information, see [“Configuring the Receiver cluster” on page 163](#).

3. Update the Receiver cluster to process data and send it to the Apache Kafka brokers.

To update the Receiver cluster:

- a. Copy the matching patterns from the Insight Pack configuration to the Logstash servers in the Receiver cluster.
- b. Update the input section with the input section from the Logstash configuration for the Insight Pack.
- c. Update the filter section with the matching logic from the Logstash configuration files for the Insight Pack and add fields that are mapped to the Apache Kafka topic or partition. For example:

```
filter {
  if [type] == "LFA"{
    mutate {
      strip => ["message"]
    }
    grok {
      match => [ "message", "%{LFAMESSAGE}" ]
      patterns_dir => [ "<Patterns_directory>" ]
      add_tag => ["grok_lfa_prod"]
    } # end LFA grok
    if "grok_lfa_prod" not in [tags]{
      grok{
        match => [ "message", "%{LALFAMESSAGE}" ]
        patterns_dir => [ "<Patterns_directory>" ]
        add_tag => ["grok_lfa"]
      }
    }
  }

  if ( "grok_lfa_prod" in [tags] or "grok_lfa" in [tags] ) {
    mutate {
      add_field => [ "datasource", "LA_Health_Pack" ]
      add_field => [ "resourceID", "%{LFA_HOSTNAME}_%{LFA_LOGNAME}" ]
    }
  }
}
```

This example shows the output section:

```
output {
  if ("grok_lfa" in [tags]) and ! ("_grokparsefailure" in [tags]) {
    kafka {
      bootstrap_servers =>
      "<Kafka_broker_server1>:<kafka_broker_port1>,..>"
      topic_id => "%{datasource}"
      message_key => "%{resourceID}"
    }
  }
}
```

The datasource field is mapped to a topic in Apache Kafka. The resourceID is mapped to a partition.

4. Copy the remaining portions of the Logstash configuration file for the Insight Pack to the Sender configuration file.
5. Update the Sender configuration file so that it can read data from the topics and partitions in Apache Kafka and send it to Log Analysis.

To update the Sender configuration:

- a. Copy the matching patterns from the Logstash configuration file for the Insight Pack to the Logstash servers in the Sender cluster.
- b. Update the Input section of the Sender cluster configuration to read data from the topic or partition in Apache Kafka. For example:

```
input {  
  kafka {  
    zk_connect => "<Zookeeper_Host>:<Zookeeper_Port>"  
    group_id => "LA_Health_Pack"  
    topic_id => "LA_Health_Pack"  
    consumer_threads => 4  
    consumer_restart_on_error => true  
    consumer_restart_sleep_ms => 100  
    decorate_events => true  
  }  
} #end inputs
```

- c. Update the filter section of the Sender configuration with the remaining configuration from the Logstash configuration file for the Insight Pack.
- d. Update the Output section to send data to Log Analysis.

For more information, see [“Streaming data with logstash”](#) on page 218.

Configuring Insight Packs that use the LFA to stream data and Log Analysis to annotate it

To integrate the scalable data collection architecture with any Insight Packs that use the LFA to stream data and Log Analysis to annotate it, you need to adapt the configuration to make it compatible with scalable data collection.

Before you begin

Configure the scalable data collection architecture. For more information, see [“Configuring scalable data collection”](#) on page 158.

About this task

The configuration described in this topic is intended for Insight Packs that use the LFA to stream data and Log Analysis to annotate it. For example, the WebSphere Application Server Insight Pack is one such Insight Pack.

In this task, you update the Receiver cluster configuration so that data is sent from the LFA to Apache Kafka. You also update the Sender cluster configuration to pull the data from Apache Kafka and send it to Log Analysis.

Procedure

1. Create a custom data source in Log Analysis. Choose an appropriate type, for example **WASSystemOut**, and complete the other fields.

For example:

Table 40. Example data source	
Data source field	Input
Host name	PUNE_WAS

Table 40. Example data source (continued)	
Data source field	Input
File path	SystemOut
Type	WASSystemOut
Name	PUNE_WAS_SystemOut

2. Configure the LFA.

Update the format or .fmt file to add the metadata fields for processing. For example

```
REGEX AllRecords
(.* )
hostname LABEL
-file FILENAME
RemoteHost DEFAULT
logpath PRINTF("%s",file)
type SystemOut
module WAS
site PUNE
text $1
END
```

Add the Receiver cluster or the HAProxy server and port information to the LFA's configuration or .conf file. For example:

```
ServerLocation=<HAProxy_or_receiver_cluster_server>
ServerPort=<HAProxy_or_receiver_cluster_port>
```

For more information, see [“Configuring the LFA” on page 167](#).

3. Update the Receiver cluster configuration.

You need to specify the required metadata information to facilitate the creation of topics and partitions in Apache Kafka. For more information, see [“Configuring the Receiver cluster” on page 163](#).

To update the Receiver cluster, complete these steps:

- Configure the matching pattern in the `<Logstash_install_location>/<patterns_directory>` directory where you store your patterns. This pattern matches the message and extracts the metadata fields. For example:

```
WASLFAMESSAGE
<START>.*type='%{DATA:LFA_TYPE}';text='%{DATA:LFA_ORIG_MSG}'
;RemoteHost='%{DATA:LFA_REMOTE_HOST}';site='%{DATA:LFA_SITE}'
;hostname='%{DATA:LFA_HOSTNAME}';module='%{DATA:LFA_MODULE}'
;logpath='%{DATA:LFA_LOGNAME}';END
```

This pattern needs to be specified on a single line in the patterns file

- Update the Receiver cluster configuration to match the message and create the topic and partition information for Apache Kafka. For example:

```
filter {
  if [type] == "lfa" {
    grok {
      patterns_dir => "<patterns_directory>"
      match => [ "message", "%{WASLFAMESSAGE}" ]
      add_tag => ["grok_lfa"]
    }
    if "grok_lfa" in [tags] {
      mutate {
        replace => ["message", "%{LFA_ORIG_MSG}"]
        add_field => [ "datasource", "%{LFA_SITE}_%{LFA_MODULE}_%{LFA_TYPE}" ]
        add_field => [ "resourceID", "%{LFA_HOSTNAME}_%{LFA_LOGNAME}_1" ]
      }
    }
  }
}
```

- c. Update the output section of the Receiver cluster configuration to send data to the Apache Kafka brokers. For example:

```
output {
  if ("grok_lfa" in [tags]) and ! ("_grokparsefailure" in [tags]) {
    kafka {
      bootstrap_servers =>
"<Kafka_broker_server1>:<kafka_broker_port1>,..."
      topic_id => "%{datasource}"
      message_key => "%{resourceID}"
    }
  }
}
```

The `datasource` field is `PUNE_WAS_SystemOut`. The `resourceID` field is composed of the host name and absolute file path, which are unique for a specific log file. The `datasource` and `resourceID` fields are mapped to topics and partitions in Apache Kafka.

4. Update the Sender cluster configuration.

- a. Update the input section of the Sender cluster configuration so that it can receive data that is sent from the topic in Apache Kafka. For example:

```
input {
  kafka {
    zk_connect => "<Zookeeper_host>:<Zookeeper_port>"
    group_id => "<Kafka_group_id>"
    topic_id => "<Kafka_topic_id>"
    consumer_threads => 5
    consumer_restart_on_error => true
    consumer_restart_sleep_ms => 100
  }
}
```

The `group_id` and the `topic_id` must match the values that are specified in the metadata.

- b. Update the `filter` section of the Sender configuration. Add the `host` and `path` fields to the message so that the message is mapped to the data source that is specified in Log Analysis. For example:

```
filter {
  mutate {
    add_tag => ["NO_OP"]
  }
  if "grok_lfa" in [tags] {
    mutate {
      replace => { "host" => "%{LFA_SITE}_%{LFA_MODULE}" }
      add_field => { "path" => "%{LFA_TYPE}" }
    }
  }
}
```

The `host` and `path` fields must match the **Hostname** and **File Path** that you specified when you created the custom data source in step 1.

- c. Update the output section of the Sender cluster configuration with the Log Analysis plug-in information so that it can communicate with the Log Analysis server. For more information, see [“Streaming data with logstash” on page 218](#).

For more information, see [“Configuring the Sender cluster” on page 162](#).

Configuring Insight Packs that use Logstash to stream data and annotate it

To integrate the scalable data collection architecture with any Insight Packs that use Logstash to stream and annotate data before it is sent to Log Analysis, you need to adapt the configuration to make it compatible with scalable data collection.

Before you begin

Configure the scalable data collection architecture. For more information, see [“Configuring scalable data collection”](#) on page 158.

About this task

The configuration that is described here is intended for Insight Packs that use Logstash to stream data to Log Analysis for processing and indexing. For example, the Oracle DB Alert Insight Pack is one such Insight Pack.

In this task, you copy the Logstash configuration for the Insight Pack to your scalable data collection architecture. Copy the `input` section from the Logstash configuration for the Insight Pack to your Receiver cluster configuration. You also update the Sender cluster configuration to read data from Apache Kafka and send it to Log Analysis.

Procedure

1. Deploy Logstash on the server where the logs that you want to process are generated.
2. Configure the Logstash instances that you deployed to collect log files as instances in your Receiver cluster.

To configure these instances as part of your Receiver cluster, complete these steps:

- a. Update the `input` section of the Receiver cluster configuration with the `input` section from the Logstash configuration file for the Insight Pack. For example:

```
input {
  file {
    type => "BlueMedoraOracleDBAlert"
    path => ["<path_to_log_file>"]
    start_position => "beginning"

    codec => multiline {
      patterns_dir => "<path_to_patterns_directory>"
      pattern => "%{ORACLEDBTIMESTAMP1}"
      negate => true
      what => previous
    } # end codec
  } # end file i/p
}
```

- b. Update the `filter` section so that the metadata fields are added which map the logs to the topics and partitions in Apache Kafka. For example:

```
filter {
  mutate {
    add_field => ["LFA_SITE", "PUNE"]
    add_field => ["LFA_TYPE", "Alert"]
    add_field => ["LFA_INSTANCE", "testInstance"]
    add_field => ["LFA_HOSTNAME", "1a12345.example.com"]
    add_field => ["LFA_MODULE", "ORACLE"]
    add_field => ["LFA_ENVIRONMENTNAME", "DEV"]
    add_field => ["LFA_LOGNAME", "<path_to_log_file>"]
    add_field => ["LFA_FUNCTIONALNAME", "NONE"]

    add_tag => ["mock_lfa"]

    add_field => [ "datasource", "%{LFA_SITE}_%{LFA_MODULE}_%{LFA_TYPE}" ]
    add_field => [ "resourceID", "%{LFA_HOSTNAME}_%{LFA_LOGNAME}_1" ]
  } # end mutate
}
```

c. Update the output section to send data to Apache Kafka. For example:

```
output {
  if ("mock_lfa" in [tags] and !("_grokparsefailure" in [tags])) {
    kafka {
      bootstrap_servers =>
      "<Kafka_broker_server1>:<kafka_broker_port1>,.. "
      topic_id => "%{datasource}"
      message_key => "%{resourceID}"
    } # end kafka
  } # end if
} # end output
```

3. Update the Sender cluster configuration to read data from the topic in Apache Kafka and process the logs based on the Logstash configuration file for the Insight Pack.

To update the Sender configuration, complete the following steps:

- a. Copy the matching patterns from the Logstash configuration file for the Insight Pack to the Sender cluster.
- b. Update the input section of the Sender cluster configurations to read data from the topic or partition in Apache Kafka.
- c. Copy the remaining configuration from the Logstash configuration file for the Insight Pack to the filter section of the Sender cluster configurations.
- d. Update the output section to send data to the Log Analysis server.

Managing your data collection components

Read about the commands which you can use to start and stop your data collection components.

Ensure that you start and stop components in the correct order to avoid data inconsistencies.

Starting and stopping your cluster in the right order

To avoid exceptions during processing, start and stop the components of your Apache Kafka cluster in the specified order.

When you want to start the cluster, start the components in the following order:

1. Apache ZooKeeper servers
2. Apache Kafka brokers
3. Receiver Logstash servers
4. HAProxy service
5. Sender Logstash servers

When you want to stop the cluster, stop the components in the following order:

1. Sender Logstash servers
2. Receiver Logstash servers
3. HAProxy service
4. Apache Kafka brokers
5. Apache ZooKeeper servers

Commands for managing components

Use these commands to manage the components of your scalable data ingestion architecture.

Apache ZooKeeper

Table 41. Apache ZooKeeper commands	
Action	Command
Console mode	
Start	<code><Kafka_dir>/bin/zookeeper-server-start.sh config/zookeeper.properties</code>
Stop	<code><Kafka_dir>/bin/zookeeper-server-stop.sh</code>
Daemon mode	
Start	<code><Kafka_dir>/bin/zookeeper-server-start.sh -daemon config/zookeeper.properties</code>
Stop	<code><Kafka_dir>/bin/zookeeper-server-stop.sh</code>

Apache Kafka broker

Table 42. Apache Kafka commands	
Action	Command
Start in daemon mode	<code><Kafka_dir>/bin/kafka-server-start.sh -daemon config/server0.properties</code>
Stop	<code><Kafka_dir>/bin/kafka-server-stop.sh</code>

Logstash

You create a custom script to manage Logstash. This script is based on the script used in Log Analysis. For more information, see [“Installing Logstash and the utility script” on page 161](#).

For more information about how to use this script, see [“Logstash operations” on page 231](#)

Troubleshooting scalable data collection

You may need to troubleshoot issues with the data collection architecture.

Before you can troubleshoot Logstash, you need to enable logging. For more information, see [“Configuring logging for troubleshooting” on page 169](#).

Troubleshooting Logstash

After you enable logging, you can find the log files in the directory that you specified in the configuration.

You can also find the log files in the path you specified for the `log_file` parameter in the Log Analysis output plugin in your Sender configuration.

Troubleshooting Apache Kafka

Apache Kafka logs are automatically generated and are stored in the `<Kafka_home>/logs` directory.

Apache Kafka includes some utilities which you can use to monitor topics and data. You can use these utilities to help you to monitor events for a specified data source.

To check the messages that have been received by the Apache Kafka cluster for a specific topic, enter the following command:

```
<Kafka_home>/bin/kafka-console-consumer.sh  
--zookeeper <zookeeper_host>:<zookeeper_port>  
-topic <topic>
```

where *<topic>* is the ID of the topic that you want to check.

In some cases, Apache Kafka may not start if you have not enabled verbose logging. For more information, see [Cannot start Apache Kafka broker](#).

Increasing the volume of data

Over time, you will want to increase the volume of data that you stream into Log Analysis.

There are two scenarios for increasing the volume of data that is collected and processed by Log Analysis.

- You can increase the volume of data by adding an additional physical data source to an existing logical data source in Log Analysis. To do this, add a new file to the DataSources filed in your existing LFA configuration file. For more information, see [“Enabling the LFA” on page 192](#).
- You can also create a new logical data source and add this to your scalable data collection architecture. To stream more data with a new logical data source, create a new data source in Log Analysis. For more information, see [“Data Sources” on page 155](#).

Loading batches of data

In addition to streaming data directly, you can also load batches of historic data for test or other purposes.

Data Collector client

Use the Data Collector client to load batches of data into Log Analysis.

Batch loading historic log data with the Data Collector client

Use the Data Collector client to ingest data in batch mode. Use this method to review historic log data. This is the easiest method if you want to ingest large log files for historic analysis.

Before you begin

If you want to use the Data Collector client to load data from remote sources, you must configure the data collector on the remote host before you can configure the local data collector as described here. For more information, see [“Configuring the Data Collector client to ingest data from remote hosts” on page 180](#).

About this task

If you want to load a log file that does not include time stamp information, ensure that the values for `timestamp` and `timestampFormat` are configured in `javaDatacollector.properties`. IBM Operations Analytics - Log Analysis cannot index log files without a time stamp, but if no time stamp information is found in a log file, the value that is configured in `javaDatacollector.properties` is used.

Procedure

To use the Data Collector client to load log file information, complete the following steps:

1. In the Administrative Settings page, define an appropriate log file source.
2. At the command line, navigate to the `<HOME>/utilities/datacollector-client` directory.

3. Update the configuration file that is used by the Data Collector client, `javaDatacollector.properties`.

Set the following properties, as appropriate:

logFile

The full path of the file you want to ingest.

servletURL

The URL of the Data Collector service.

userid

The user ID for the Data Collector service.

password

The password for the Data Collector service.

datasource

The datasource that you want to use to load data.

timestamp

The time stamp to use if a time stamp is not found in the log file.

batchsize

The number of BYTES of logs that are sent in one batch. The default value is 500,000.

keystore

The full path to the keystore file.

inputType

The valid input types are: LOGS, CONFIGFILES, SUPPORTDOCS. The default value is LOGS.

flush flag

If the default `true` is set, the client sends a flush signal to the Generic Receiver for the last batch of the file. If set to `false`, no flush signal is sent when the end of file is reached.

The following sample `javaDatacollector.properties` file displays the configuration for loading the `SystemOut.log` log file.

```
#Full path of the file you want to read and upload to Unity
logFile = SystemOut.log
#The URL of the REST service. Update the host/port information if required
servletURL = https://hostname:9987/Unity/DataCollector
#The user ID to use to access the unity rest service
userid=unityuser
#The password to use to access the unity rest service
password=password
datasource=Systemout
#Time stamp to use if your content can not find a time stamp in log record.
The same time stamp would be used for all records
timestamp = 01/16/2013 17:27:23:964 GMT+05:30
#The number of BYTES of logs sent in one batch to Unity
batchsize = 500000
#The full path to the keystore file
keystore = /home/unity/IBM/LogAnalysisTest/wlp/usr/servers/Unity/
keystore/unity.ks
#input data type - LOGS, CONFIGFILES, SUPPORTDOCS
inputType = LOGS
#flush flag:
#true : (default) if the client should send a flush signal to the Generic
Receiver for the last batch of this file
#false : if no flush signal to be sent upon reaching eod-of-file
flushflag = true
#Other properties (name/value pairs, e.g. middleware = WAS) that you want
to add to all json records
#These properties need to be appropriately added to the index configuration
```

4. Ensure that the Data Collector client JAR file, `datacollector-client.jar`, has execute permissions.
5. Use the following command to run the Data Collector client with the correct inputs:

```
<HOME>/ibm-java/bin/java
-jar datacollector-client.jar
```

Results

After the task completes, the log file is indexed and can be searched in the **Search** workspace.

Configuring the Data Collector client to ingest data from remote hosts

If you want to use the Data Collector client to collect data from a remote server and return it to the local machine, you must configure the data collector on the remote host.

Before you begin

You must use the instance of IBM Java™ Runtime Engine (JRE) 1.8 that is installed by the remote installer. Before you configure the data collector, you must use the remote installer to install at least one instance of IBM Tivoli Monitoring Log File Agent or the EIF Receiver on a remote machine. For more information, see the *Configuring data collection for scalability on multiple remote nodes* topic in the Installation Guide.

About this task

To configure the Data Collector on the remote host, copy the data collector client files from your local version of the data collector files to the remote host.

Procedure

1. Copy the <HOME>/utilities/datacollector-client directory and all the files that are contained in it from the local installation of IBM Operations Analytics - Log Analysis to the remote machine.
2. Add the location of the log and keystore files to the javaDatacollector.properties file in the directory that you copied the data collector to in the previous step.

The keystore file is named unity.ks and it is available in the <Remote_install_dir>/LogAnalysis/store/ directory on the remote machine. Where <Remote_install_dir> is the directory where you installed the remote instance as described in the *Prerequisites* section here.

Results

After you complete the configuration, you must complete the Data Collector configuration. For more information about how to do this, see [“Batch loading historic log data with the Data Collector client” on page 178](#). You must ensure that the remote installation uses the IBM Java Runtime Engine (JRE) 1.8 that is installed by the remote installer. IBM Java Runtime Engine (JRE) 1.8 is stored in the <Remote_install_dir>/LogAnalysis/ibm-java/ directory.

Data Collector properties

Before you can use the data collector to stream data or load a batch of historic data, edit the javaDatacollector.props file.

The javaDatacollector.props file is in the <HOME>/IBM/LogAnalysis/utilitiesdatacollector-client folder.

The logFile, hostname, logpath, and keystore parameters are required.

The userid, password, and keystore parameters are automatically populated with the default values that are created during the installation. If you want, you can change these but you do not need to.

Table 43. Data Collector properties	
Parameter	Value
logFile	The full path of the file you want to load.
servletURL	The URL of the Data Collector service.
userid	The user ID for the Data Collector service.
password	The password for the Data Collector service.

Table 43. Data Collector properties (continued)	
Parameter	Value
datasource	The datasource that you want to use to load data.
timestamp	The time stamp to use if a time stamp is not found in the log file.
batchsize	The number of BYTES of logs sent in one batch. The default value is 500,000.
keystore	The full path to the keystore file.
inputType	The valid input type is LOGS.
flush flag	If the default true is set, the client sends a flush signal to the Generic Receiver for the last batch of the file. If set to false no flush signal is sent when the end-of-file is reached.

Generic Receiver

The Generic Receiver is a component of IBM Operations Analytics - Log Analysis that supports the REST interface for loading data into IBM Operations Analytics - Log Analysis. The REST API uses JSON (JavaScript Object Notation) as an input and returns JSON as an output after the incoming logs are processed. If an error occurs, the API returns an error code and a message.

Processing a batch

Invoking the Generic Receiver API initiates the processing of a batch that is contained in the Input JSON. Buffer a set of log records to create a batch and send data in batches to IBM Operations Analytics - Log Analysis. The batches must be sent in the order in which logs are generated for a specific data source. The size of each batch must be less than the batch size (500000 bytes) supported by IBM Operations Analytics - Log Analysis. At the minimum, you can send data for a single log record in a batch. The Generic Receiver processes a batch by:

- Splitting the batch into multiple log records by using the Splitter that was specified during the creation of the SourceType from the Admin UI corresponding to the data source
- Annotates every log record that is found by the Splitter by using the Annotator that is specified during the creation of the SourceType from the Admin UI corresponding to the data source
- Indexing the annotated log record in the back-end search engine

As special cases, split and annotated steps are skipped if the Splitter or Annotator is specified as null in the SourceType. Even if there is no data to split, you need to send an empty string in the text field of the Input JSON.

Batching of data at the client might lead to an incomplete log record at the end of the batch. This incomplete log record gets buffered in IBM Operations Analytics - Log Analysis and stitched with the remaining data in the subsequent batch to form a complete log record. This stitching assumes that you are maintaining the log record order of the data that is sent to IBM Operations Analytics - Log Analysis. If the order is not maintained, then logs are not correctly split into log records.

Input JSON

The basic structure of an Input JSON file is:

```
{
  "hostname": " ",      (String)
  "logpath": " ",      (String)
  "batchsize": " ",    (String)
  "inputType": " ",    // Optional (String) "LOGS";
  "flush": " ",       // Optional (boolean)
  "payload": " ",     // (JSONObject)
```

```

{
  "name1": "value1",           // Optional
  ...
  ...
  "nameN": "valueN",          // Optional
  text : "log record 1 log record 2 ..." (String)
}
}

```

The following parameters in the Input JSON are mandatory:

hostname

The host name that corresponds to the data source for which you want to ingest data.

logpath

The log path that corresponds to the data source for which you want to ingest data.

batchsize

The number of BYTES of logs that are sent in one batch to IBM Operations Analytics - Log Analysis (less than 500,000).

inputType

The type of input data: LOGS.

flush flag

A flag that indicates to the Generic Receiver whether the last record in the batch is a complete log record. Typically, this flag would be set to true in the last batch upon reaching the end of file.

payload.txt

This text contains the actual log records to be split, annotated, and indexed into IBM Operations Analytics - Log Analysis. The text portion is split into log records by the Splitter, annotated by the Annotator, and then indexed. If you do not have any log records, but want to index only structured (name-value pairs) data, you can specify this mandatory field as an empty string.

More metadata (optional) to be indexed with every log record of the batch can be specified as name-value pairs in the input JSON or the payload within the input JSON. This metadata is applicable at the batch level. For posting distinct metadata for each log record, send 1 log record at a time in the batch.

Post the input JSON to the following URL:

```
http://<UNITY_HOST_NAME>:<UNITY_PORT>/Unity/DataCollector
```

where <UNITY_HOST_NAME> is the machine on which you installed IBM Operations Analytics - Log Analysis and <UNITY_PORT> is the port on which it is running. The default port is 9988. The client (Java or Script) sending data into IBM Operations Analytics - Log Analysis needs to authenticate by using the form-based mechanism that is implemented in IBM Operations Analytics - Log Analysis before the Data Collector API is invoked. Refer to the authentication and security design document for details.

Output JSON

The output that is sent by the Generic Receiver after indexing logs contains the count and detailed information on the failure cases in a JSON Array. The details include the actual logRecord, specific error message, and any exception. The basic structure of an Output JSON file is:

```

{
  "batchSize" : ,           // (int)
  "numFailures" : ,         // (int)
  "failures" :              // (JSONArray)
  [
    {
      "logRecord" : ,        // (JSONObject)
      "errorMessage" : ,     // (String)
      "exception" : ,        // (JSONArray)
    },
    .
    .
    .
  ]
}

```

```
} ]
```

Serviceability

As you send data into IBM Operations Analytics - Log Analysis, you might encounter errors that occur before the incoming batch gets processed or errors that occur during processing of batch and indexing log records.

If errors occur before the incoming batch gets processed, the Generic receiver returns an error code and message. To correct the problem, process the error code, make any required changes, and resend the data.

Possible causes for error code 400 (HttpServletResponse.SC_BAD_REQUEST) include:

- Invalid input JSON
- Input batch size is greater than what is supported (500000 bytes)
- No data source is configured from the Admin UI for the host name and log path combination that is sent in the input JSON
- The input type (LOGS) specified in the batch does not match the value that is specified in the logsource that is configured from the Admin UI

Possible causes for error code 500 (HttpServletResponse.SC_INTERNAL_SERVER_ERROR) include:

- An exception that is encountered in any of the steps of the ingestion pipeline (for example, during splitting of a batch).
- An internal IBM Operations Analytics - Log Analysis database-related error.
- Any other exception in IBM Operations Analytics - Log Analysis.

If errors occur during processing of batch and indexing log records, the output JSON provides details of indexing failure. To correct the problem, process the error code, make any required changes, and resend only the affected log records. Sending the same log record twice to IBM Operations Analytics - Log Analysis results in duplicate records in the back-end index and duplicate records in the search results.

Loading batches of historic data with the IBM Tivoli Monitoring Log File Agent

You can use the IBM Tivoli Monitoring Log File Agent to load batches of historic data for testing and other purposes.

Procedure

1. Copy the log files that you want to load to a temporary directory on the IBM Operations Analytics - Log Analysis server. For example, to upload a batch of log files from an installation of WebSphere Application Server, you copy the `SampleSystemOut.log` file to the `/tmp/logs/` directory.
2. Create a custom data source.
3. Copy the log file to the directory that you specified in the `logpath` parameter when you created the data source.

Loading a batch of log files with the LFA

Use this example to help you to understand how to use the LFA to load log a batch of files.

Before you begin

Consider the size of the log files that you want to load. If a log file is in the region of 50 MB, or more, in size, increase the size of the log file agent cache. In the appropriate configuration file, set `BufEvtMaxSize=102400`. For WAS log files, update `<HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo/WASInsightPack-lfawas.conf`. For DB2 log files, update `<HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo/DB2InsightPack-lfadb2.conf`.

You must delete the appropriate existing cache file. For WAS log files, delete <HOME>/IBM/LogAnalysis/logs/lfa-WASInsightPack.cache and for DB2 log files, delete <HOME>/IBM/LogAnalysis/logs/lfa-DB2InsightPack.cache

For very large log files, update the cache size of the EIF receiver. In the <HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/eif.conf file, increase the value of the BufEvtMaxSize property.

Lines in a log that are longer than 4096 characters are, by default, ignored by the LFA. To force it to read lines longer than 4096 characters, add the EventMaxSize=<length_of_longest_line> property to the .conf file that will be used while loading the log.

For WAS update \$UNITY_HOME/IBM-LFA-6.30/config/lo/WASInsightPack-lfawas.conf file.
DB2 update \$UNITY_HOME/IBM-LFA-6.30/config/lo/DB2InsightPack-lfadb2.conf file.

If you make any changes to the configuration, you must restart the service for the changes to take effect. To restart the service, from the <HOME>/IBM/LogAnalysis/utilities directory, run the following commands:

- <HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
- <HOME>/IBM/LogAnalysis/utilities/unity.sh -start

About this task

The LFA might be on the same server as IBM Operations Analytics - Log Analysis and monitoring a local directory. In this scenario, the installation of IBM Operations Analytics - Log Analysis completes all of the configuration required.

If the LFA is on the same server as IBM Operations Analytics - Log Analysis, but monitoring remote directories, some additional configuration is required. If you want to monitor log files on remote servers, you must make some specific settings changes. For more information about these specific settings, see the *Configuring remote monitoring that uses the predefined configuration files* topic under *IBM Tivoli Log File Agent Configuration* in the *Extending IBM Operations Analytics - Log Analysis* section.

If your configuration requires it, you can use a remote LFA. In this scenario, install and configure the LFA based on the your requirements. For more information, see the IBM Tivoli Monitoring documentation: http://www-01.ibm.com/support/knowledgecenter/SSTFXA_6.3.0/com.ibm.itm.doc_6.3/welcome.htm

Procedure

To use the log file agent to load log information, complete the following steps:

1. In the Administrative Settings page, define an appropriate log file source.
2. Ensure that the log file you want to add is in the appropriate directory.

For WAS logs, place the log file in the following directory:

```
<HOME>/IBM/LogAnalysis/logsources/WASInsightPack
```

For DB2 logs, place the log file in the following directory:

```
<HOME>/IBM/LogAnalysis/logsources/DB2InsightPack
```

For Generic annotator log files, place the log file in the following directory:

```
$UNITY_HOME/logsources/GAInsightPack
```

The log file is automatically picked up and analyzed. Depending on the size of the log file, processing it could take some time.

3. Optional: To monitor progress, check the following log files:
 - <HOME>/IBM/LogAnalysis/logs/GenericReceiver.log
 - <HOME>/IBM/LogAnalysis/logs/UnityEifReceiver.log

When you are using the LFA to perform data collection, monitor the `UnityEIFReceiver.log` and `GenericReceiver.log` log files located in the `$UNITY_HOME/logs` directory to ensure that the data ingestion has completed correctly.

This example illustrates the addition of a batch of log records. The result is indicated in the `RESPONSE MESSAGE` section of the log file:

```
~~~~~
2013-04-20 04:43:10,032 [pool-5-thread-1] INFO - LogEventPoster : -
Posting Event to UNITY DATA COLLECTOR -
https://nc9118041070:9987/Unity/DataCollector
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO - LogEventPoster :
+++++++ RESPONSE MESSAGE ++++++
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO - LogEventPoster : OK
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO - LogEventPoster :
{ "batchSize": 2078,
"failures": [ ], "numFailures": 0 }
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO - LogEventPoster :
+++++++
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO - LogEventPoster :
EIF event delivery to Generic Receiver -- SUCCESS
~~~~~
```

In this log, the number of log records processed is indicated in the line:

```
{ "batchSize": 2078, "failures": [ ], "numFailures": 0 }
```

2078 log records were successfully ingested. The `numFailures` value indicates the number of failures in the ingestion of the log records. For example, a value of 5 for the `numFailures` value indicates that 5 log records were not ingested.

When data collection has completed, if the EIF Receiver buffer is partially filled, any remaining log records are posted to the Generic Receiver. This is recorded in the log as a `TIMEOUT FLUSH` event. These events are added to the log file at the end of the session of data collection:

```
~~~~~
2013-04-20 04:54:26,341 [pool-4-thread-1] INFO - LogEventService :
TIMEOUT FLUSH for logsource:nc9118041070::
/home/example/LogAnalytics/logsources/
WASInsightPack/TipTrace5.log
2013-04-20 04:54:26,359 [pool-5-thread-1] INFO - LogEventPoster : ---
Posting Event to UNITY DATA COLLECTOR -
https://nc9118041070:9987/Unity/DataCollector
2013-04-20 04:54:38,581 [pool-5-thread-1] INFO - LogEventPoster :
+++++++ RESPONSE MESSAGE ++++++
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO - LogEventPoster : OK
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO - LogEventPoster :
{ "batchSize": 1714,
"failures": [ ], "numFailures": 0 }
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO - LogEventPoster :
+++++++
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO - LogEventPoster :
EIF event delivery to Generic Receiver -- SUCCESS
2013-04-20 04:54:38,583 [pool-4-thread-1] INFO - LogEventService :
POST RESULT:
{"failures": [], "batchSize": 1714, "numFailures": 0}
~~~~~
```

To calculate the number of events that have been processed, calculate the sum of all of the `batchSize` values. To calculate the number of events ingested, calculate the sum of all of the `batchSize` values and deduct the total sum of `numFailure` values.

If the ingestion fails, an error message is recorded in the `UnityEIFReceiver.log`:

```
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO - LogEventPoster :
+++++++ RESPONSE MESSAGE ++++++
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO - LogEventPoster : Not Found
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO - LogEventPoster :
{"BATCH_STATUS": "NONE", "RESPONSE_MESSAGE":
"CTGLA0401E : Missing log source ", "RESPONSE_CODE": 404}
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO - LogEventPoster :
+++++++
```

```
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO - LogEventPoster :  
FAILURE - ResponseCode:404 ResponseMessage:Not Found
```

Additional HTTP response codes are as follows:

413

Request Entity Too Large: Displayed if a batch size is greater than the Generic Receiver default value set in the \$UNITY_HOME/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties.

500

Internal Server Error: Displayed when there is any issue with IBM Operations Analytics - Log Analysis such as a database error or any other runtime error.

404

Not Found: Displayed when a Log Source is not found for a hostname and log path combination in the request.

409

Conflict: Displayed if the data batch is posted for a Log Source that is in an inactive state or if there is a conflict between the data posted and the data expected by the server. For example, the inputType field in the request JSON does not match the inputType field in the Collection for the requested hostname and log path combination.

200

OK: Displayed when the request is processed by the server. The status of the processed batch of records is returned with the total number of records ingested, how many failed records are present and which failed.

400

Bad Request: Displayed when the request JSON does not contain the required fields expected by the Generic Receiver or where the JSON is not properly formed.

Results

After the task completes, the log file is indexed and can be searched using the **Search** field on the IBM Operations Analytics - Log Analysis Dashboard.

Streaming data with the IBM Tivoli Monitoring Log File Agent

To stream data to Log Analysis, you can configure and use the internal IBM Tivoli Monitoring Log File Agent (LFA) that is installed during the installation of Log Analysis or you configure an external LFA.

For more information about how to configure both types of LFA to stream data, see [“Configuring the LFA” on page 187](#).

If you use an internal LFA, you can use the remote install utility to install an LFA instance on a remote server. The utility also helps you to configure Secure Shell (SSH) authentication. For more information, see [“Streaming data from multiple remote sources across a network” on page 213](#).

You can also use other versions of the IBM Tivoli Monitoring Log File Agent to stream data from Windows or AIX operating systems. This is referred to as an external LFA. For more information, see [“IBM Tivoli Monitoring Log File Agent configuration scenarios” on page 188](#).

If you want to group an explicit collection of configurations, you can also create subnodes. For more information, see [“Configuring LFA subnodes” on page 199](#)

Considerations when using the LFA

Before you use the LFA, read this topic to help you to understand some important considerations and limitations.

Log file size

If your log files are likely to exceed 50 MB, increase the size of the LFA cache: In the appropriate configuration file, set `BufEvtMaxSize=102400`. For WAS log files, update `<HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo/WASInsightPack-lfawas.conf`. For DB2 log files, update `<HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo/DB2InsightPack-lfadb2.conf`.

You must delete the appropriate existing cache file. For WAS log files, delete `<HOME>/IBM/LogAnalysis/logs/lfa-WASInsightPack.cache` and for DB2 log files, delete `<HOME>/IBM/LogAnalysis/logs/lfa-DB2InsightPack.cache`

For very large log files, update the cache size of the EIF receiver. In the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/eif.conf` file, increase the value of the `BufEvtMaxSize` property.

For WAS, update `<HOME>/IBM-LFA-6.30/config/lo/WASInsightPack-lfawas.conf` file. DB2 update `<HOME>/IBM-LFA-6.30/config/lo/DB2InsightPack-lfadb2.conf` file.

If you make any changes to the configuration, you must restart the service for the changes to take effect. To restart the service, from the `<HOME>/IBM/LogAnalysis/utilities` directory, run the following commands:

- `unity.sh -stop`
- `unity.sh -start`

Maximum log line length

The LFA monitors each log file line. The default maximum line length that can be processed by the LFA is 4096 bytes. This is equivalent to 4096 ASCII characters. This limitation is related to the log line and not the log record. If a log record consists of multiple log lines, such as in the case of a stack trace, the limit applies to each line. This is a limitation of the LFA and does not apply if you use an alternative data collection mechanism.

Performance implications of using the LFA

Loading logs using the LFA is a CPU bound process. If your system does not meet the minimum requirements you will need to increase the `MaxEventQueueDepth`. On some systems, altering this value may produce a noticeable impact on performance. This will buffer additional LFA events while they are waiting to be processed. The required value for `MaxEventQueueDepth` may vary depending on the size of the rolled log and the number/speed of your CPU's. If you choose not to increase this value, then older events may be replaced on the event queue by newer events and not sent to the Log Analysis server.

To minimize the chance of data loss due to CPU bottlenecks, and to reduce the latency between when a log record is written to the file and when it is loaded, we recommend that the maximum size of a log be small enough so that your system does not fall behind while processing the logs.

Configuring the LFA

Before you can stream data into Log Analysis, you need to configure the LFA.

You can stream data locally, that is from an LFA on the same machine as Log Analysis or you can stream data remotely.

If you installed the LFA at the same time as IBM Tivoli Monitoring Log File Agent, you are using an internal LFA. If you installed the LFA before IBM Tivoli Monitoring Log File Agent, you are using an external LFA. For more information about installing the internal LFA, see [“Installing and configuring the IBM Tivoli Monitoring Log File Agent”](#) on page 21.

To stream data with an LFA, complete the following steps:

1. Review the possible data loading scenarios and decide whether you want to use an internal or external LFA. For more information, see [“IBM Tivoli Monitoring Log File Agent configuration scenarios”](#) on page 188.
2. Configure the required configuration file, format file, and data source. The required parameters for the configuration file are different for internal and external LFAs. For more information, see [“Enabling the LFA”](#) on page 192.
3. Deploy the LFA. For more information, see [“Configuring and deploying LFAs in the command line”](#) on page 198.

IBM Tivoli Monitoring Log File Agent configuration scenarios

You can use the internal IBM Tivoli Monitoring Log File Agent that is installed with IBM Operations Analytics - Log Analysis or you can use an external IBM Tivoli Monitoring Log File Agent to stream data from local or remote servers.

You can also use the IBM Tivoli Monitoring Log File Agent to upload a batch of historic data. For more information, see [“Loading batches of historic data with the IBM Tivoli Monitoring Log File Agent”](#) on page 183.

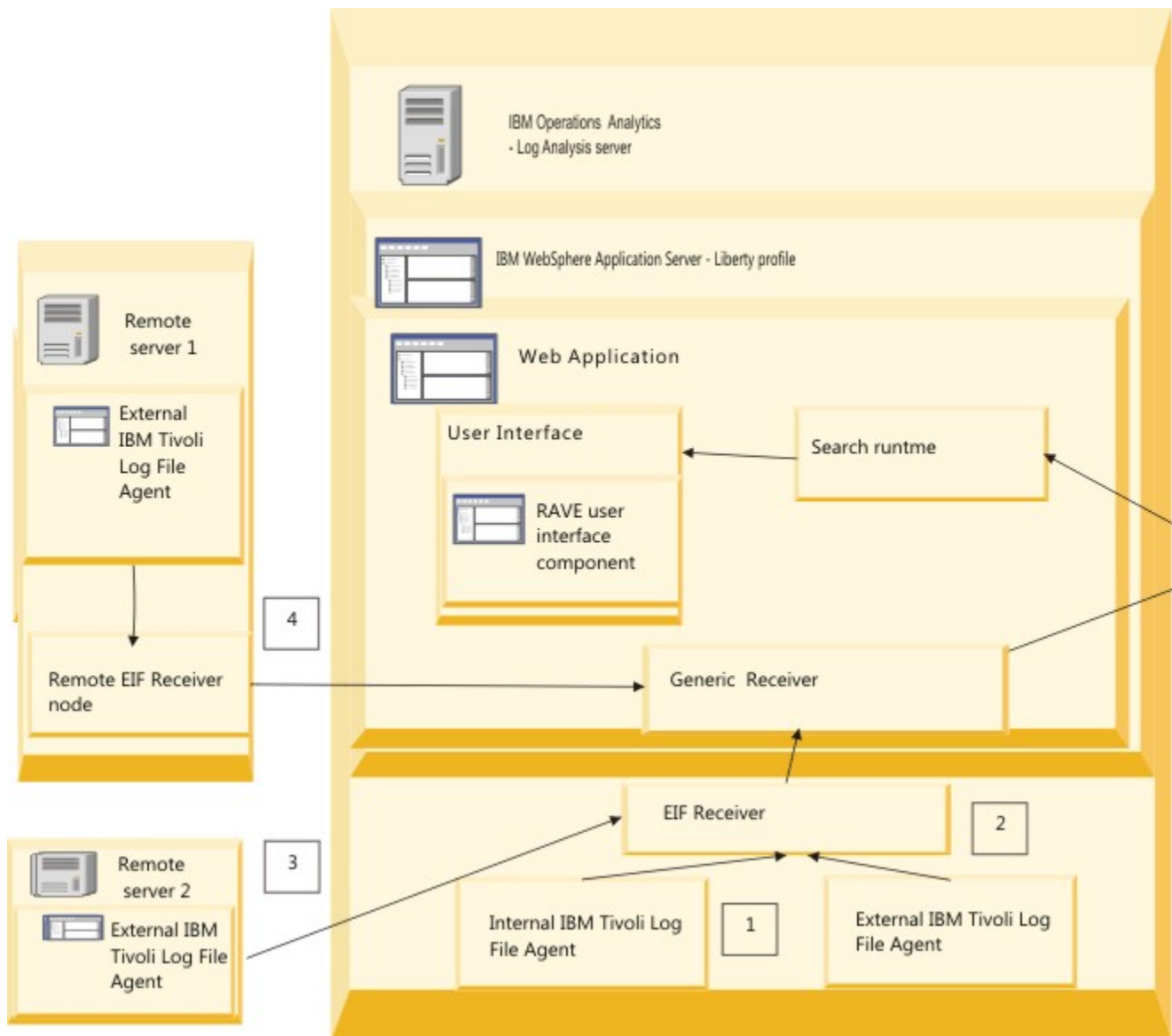
You can integrate the IBM Tivoli Monitoring Log File Agent with IBM Operations Analytics - Log Analysis in two ways.

You can use it with the version of the IBM Tivoli Monitoring Log File Agent that is installed with IBM Operations Analytics - Log Analysis. This is known as the internal IBM Tivoli Monitoring Log File Agent.

You can also use it with an IBM Tivoli Monitoring Log File Agent that has been installed separately as part of another installation.

You can use local and remote versions of both types of IBM Tivoli Monitoring Log File Agent.

The following graphic illustrates these possibilities:



The following possible scenarios are illustrated in the graphic:

1. Internal IBM Tivoli Monitoring Log File Agent on a local server

In this scenario, you use the version of the IBM Tivoli Monitoring Log File Agent that is installed with IBM Operations Analytics - Log Analysis to load data from the local installation of IBM Tivoli Monitoring to IBM Operations Analytics - Log Analysis.

2. External IBM Tivoli Monitoring Log File Agent on a local server

In this scenario, you use a version of the IBM Tivoli Monitoring Log File Agent that was not installed with IBM Operations Analytics - Log Analysis but that is installed on the same server as IBM Operations Analytics - Log Analysis.

3. External IBM Tivoli Monitoring Log File Agent on a remote server

In this scenario, you use an installation of an external IBM Tivoli Monitoring Log File Agent to push data to IBM Operations Analytics - Log Analysis. To facilitate this integration, you modify the properties of the IBM Tivoli Monitoring Log File Agent.

4. Remote instance of the internal IBM Tivoli Monitoring Log File Agent

In this scenario, you use the remote installer tool to install a remote instance of the internal IBM Tivoli Monitoring Log File Agent. For more information about to use the tool, see [“Deploying the LFA or EIF on remote servers”](#) on page 215.

The following table summarizes the different configurations that are required for the scenarios.

Table 44. Configuration for data streaming scenarios

Data streaming scenario	IBM Tivoli Monitoring Log File Agent type	Log file location	Data source type	Required parameters in .conf file
1	Internal and local. The LFA is bundled with Log Analysis and is installed on the same server as Log Analysis	Local	Local	LogSources
2	External and local. The LFA is installed on the same server as Log Analysis but it is not the LFA that is bundled with Log Analysis.	Local	Local	LogSources, ServerLocation, ServerPort, BufEvtMaxSize.
3	External and remote. The LFA is installed on the remote server and it is not the LFA that is bundled with Log Analysis.	Remote	Custom	LogSources
4	Internal and remote. You use the remote installer to create a remote instance of the LFA that is bundled with Log Analysis.	Remote	Custom	LogSources, SshAuthType, SshHostList, SshPassword, SshPort, SshPrivKeyfile, SshPubKeyfile, SshUserid.

LFA configuration and format files

If you use an internal or external LFA, you can edit the configuration and property files to suit your specific installation.

The LFA configuration for a particular data source is defined in the following files:

- A <name>.conf file that contains the properties that are used by the LFA for processing the log files.
- A <name>.fmt file that contains an expression and format that is used by the agent to identify matching log file records and to identify the properties to include in the Event Integration Format (EIF) record. The EIF is sent from the agent to the receiving server. The receiving server is the server where the Log Analysis server is installed. The <name>.fmt file uses a regular expression to determine matching records in the log file and to send each matching record to the LFA server in an EIF event.

If you want to use the LFA to send your log files to the LFA server, you must customize the regular expression and define your own stanza in the <name>.fmt file to capture the log records that are to be sent. The event record format must include the host name, file name, log path, and text message. The LFA server uses these values to process the logs. For more information about the IBM Tivoli 6.3 Log File Agent and the configuration files and properties, see [Tivoli Log File Agent User's Guide](#).

The file names must be identical for both files. For example, WASContentPack_v1.1.0-1fawas.conf and WASContentPack_v1.1.0-1fawas.fmt.

After you modify the configuration files as required, you use the LFA to load the data into LFA. For a general description of how to do this, see [“Loading a batch of log files with the LFA” on page 183](#)

If you use an external instance of the LFA to load data into the Log Analysis server, you must install the configuration files into the agent. This configuration ensures that the agent knows where the log files for a data source are located, how to process the records in the log file, and the server to which records are sent.

Requirements

Ensure that the configuration and format files that you create or modify, meet the following requirements:

- Ensure that the configuration file that you create contains a line separator between each property and that the file uses the `.conf` file extension.
- The format file must use the `.fmt` extension.
- The names of the configuration and format files must be identical. For example, `WASContentPack_v1.1.0-lfawas.conf` and `WASContentPack_v1.1.0-lfawas.fmt`.

Sample configuration and format files for Insight Packs

Log Analysis includes sample configuration and format files in the `<HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo` directory. You can use these files with the included Insight Packs to load data.

Note: If you want to use an internal or external LFA that is installed on a remote server to load data, you need to manually create the required `/lo` directory, the configuration, and format files.

LFA configuration file examples

The following example shows the files that are installed as part of the WebSphere Insight Pack that is included as standard with Log Analysis.

The `WASContentPack_v1.1.0-lfawas.conf` file contains many properties, including the following examples:

```
# Files to monitor. The single file /tmp/regextest.log, or any file like
/tmp/foo-1.log or /tmp/foo-a.log.
LogSources=/home/unityadm/IBM/LogAnalysis/logsources
           /WASInsightPack/*

# Our EIF receiver host and port.
ServerLocation=<EIF Receiver host name>
ServerPort=5529
```

The `WASContentPack_v1.1.0-lfawas.fmt` file contains the following regular expression that matches any record within a monitored log file. In this example, the regular expression matches all the log records in the file and to the Operations Analytics server as an EIF event. The EIF event contains the host name where the agent is running, the file name of the log file, the log file path of the log file, and the log file record itself.

```
// Matches records for any Log file:
//
REGEX AllRecords
(.*?)
hostname LABEL
-file FILENAME
logpath PRINTF("%s",file)
text $1
END
```

Creating data sources for the LFA

Before you can use the LFA to stream data, you must create a data source in Log Analysis.

The type of data source differs depending on the data streaming scenario:

- If you are streaming data locally with the internal LFA that is installed on the same server as Log Analysis during installation, you can create a local data source.
- If you are streaming data from a remote server with the local, internal LFA, create a remote data source. The local LFA loads the data directly from the remote server.
- If you are streaming data from a remote server with a remote instance of the internal LFA or you are using an external LFA, create a custom data source.

Ensure that you use the host name rather than the fully qualified domain name.

Ensure that the information that the log path value that you specify in the data source matches the log sources that you specify in the LFA configuration file. For more information, see [“LFA configuration and format files” on page 190](#).

Note: For local and remote data sources, format files (*<filename>.fmt*) are automatically created. If you want to modify the regular expression to process data further, manually edit the format file and restart Log Analysis.

For information about creating a data source, see [“Data Source creation” on page 309](#).

Enabling the LFA

Before you use an internal or external LFA, you must create or modify the required configuration and format files. You must also create a data source to load the data into Log Analysis.

Before you begin

Create a data source. Ensure that you use the host name rather than the fully qualified domain name. For more information, see [“Creating data sources for the LFA” on page 191](#).

Ensure that the configuration and format files that you create or modify meet the requirements. For more information, see [“LFA configuration and format files” on page 190](#).

The `<HOME>/IBM-LFA-6.30/config/lo` directory contains some example configuration and format files. These files are designed to help with common types of log file records. You can use these sample files to help you to create your own files or you can modify the files themselves and use them directly. Ensure that you specify the correct file names in your configuration. For more information, see [“LFA configuration and format files” on page 190](#).

About this task

Procedure

1. Open the configuration file that you want to use. The location and name of this file differs depending on the data loading scenario:
 - If you are streaming data locally or remotely with the internal LFA, you can modify one of the sample files in the `<HOME>/IBM-LFA-6.30/config/lo` directory directly. You only need to add the data source information.
 - If you are streaming data from a remote instance of the internal LFA that you installed with the remote installation utility, you can copy the `<HOME>/IBM-LFA-6.30/config/lo` directory to the remote server and modify the files that you want to use. You only need to add the data source, server location, and port information. Delete the files that you do not need. The utility does not copy the sample files.
 - If you are streaming data from a remote instance of an external LFA, you can use the existing configuration and format files. Ensure that you add the required parameters to the relevant configuration file.
2. Define the required parameters in the configuration file. The required parameters are different depending on the data loading scenario.

Table 45. Parameters for LFA configuration file		
Required for	Parameter	Description
All	LogSources	Specify the data source that you want to monitor. If you are specifying multiple data sources, they must be comma-separated and without spaces. When you configure a remote directory in the LFA configuration file, the directory you specify must not contain any subdirectories.
Internal LFAs installed on remote servers	ServerLocation,	Specify the server location for the EIF receiver server. For example, for a server that is at 111.222.333.444, specify the following value: ServerLocation=111.222.333.444
Internal LFAs installed on remote servers	ServerPort	Specify the port that the EIF receiver uses. For example: ServerPort=5529
Internal LFAs installed on remote servers	BufEvtMaxSize	Specify the maximum buffer size for the LFA. This parameter is the maximum size that the cache is allowed to be. If the cache is full, events are dropped and performance can decline. The value that you enter here is in kilobytes. For example: BufEvtMaxSize=102400

Table 45. Parameters for LFA configuration file (continued)

Required for	Parameter	Description
External LFAs installed on remote servers	SshAuthType	<p>You must set this value to either PASSWORD or PUBLICKEY.</p> <p>If you set this value to PASSWORD, IBM Operations Analytics - Log Analysis uses the value that is entered for SshPassword as the password for Secure Shell (SSH) authentication with all remote systems.</p> <p>If you set this value to PUBLICKEY, IBM Operations Analytics - Log Analysis uses the value that is entered for SshPassword as pass phrase that controls access to the private key file.</p>
External LFAs installed on remote servers	SshHostList	<p>You use the SshHostList value to specify the hosts where the remotely monitored log files are generated. IBM Operations Analytics - Log Analysis monitors all the log files that are specified in the LogSources or RegexLogSources statements in each remote system.</p> <p>If you specify the local machine as a value for this parameter, the LFA monitors the files directly on the local system. If you specify that the localhost SSH is not used to access the files on the system, IBM Operations Analytics - Log Analysis reads the files directly.</p>
External LFAs installed on remote servers	SshPassword	<p>If the value of the SshAuthType parameter is PASSWORD, enter the account password for the user that is specified in the SshUserId parameter as the value for the SshPassword parameter.</p> <p>If the value of the SshAuthType parameter is PUBLICKEY, enter the pass phrase that decrypts the private key that is specified in the SshPrivKeyfile parameter.</p>

Table 45. Parameters for LFA configuration file (continued)		
Required for	Parameter	Description
External LFAs installed on remote servers	SshPort	You specify the TCP port that is used for SSH connections. If you do not enter anything, this value is defaulted to 22.
External LFAs installed on remote servers	SshPrivKeyfile	<p>If the value of the SshAuthType parameter is set to PUBLICKEY, enter the directory path to the file that contains the private key of the user that is specified in the SshUserid parameter as the value for this parameter.</p> <p>If the value of the SshAuthType parameter is not set to PUBLICKEY, this value is not required.</p>
External LFAs installed on remote servers	SshPubKeyfile	<p>If the value of the SshAuthType parameter is set to PUBLICKEY, enter the directory path to the file that contains the public key of the user that is specified in the SshUserid parameter as the value for this parameter.</p> <p>If the value of the SshAuthType parameter is not set to PUBLICKEY, this value is not required.</p>
External LFAs installed on remote servers	SshUserid	Enter the user name from the remote system that the agent uses for SSH authentication.

3. Define the format file as required.

4. (Optional) If you want to monitor log files type where the log files rotate resulting in multiple log files, update the .fmt file for each rotating log type to allow for the appropriate name change. Open the .fmt file, and edit the line:

```
-file FILENAME
```

to reflect the file name rotation. For example, for SystemOut log files where a number is appended to the file name for each additional log, the FILENAME must read:

```
-file SystemOut*.log
```

5. Save your changes.

Example

For example:

```
=====
SshHostList=host1,host2,host3
```

```

SshUserid=loguser
SshAuthType=PASSWORD
SshPassword=<password>

=====
SshHostList=host1,host2,host3
SshUserid=loguser
SshAuthType=PUBLICKEY
SshPrivKeyfile = <SshUserid_Private_Key_File_Path>
(Or)
SshPubKeyfile = <SshUserid_Private_Key_File_Path>

=====

```

where <password> is the password that you want to use.

<SshUserid_Private_Key_File_Path> is the full path for the file that contains the private key of the user that is specified in the SshUserid user. For example, if you save the password to a file called password.txt in the <HOME>/utilities directory, the full path is as follows:

```
SshPrivKeyfile = <HOME>/utilities/password.txt
```

What to do next

Configure and deploy the LFA. For more information, see [“Configuring and deploying LFAs in the command line”](#) on page 198.

LFA configuration file parameters

The IBM Tivoli Monitoring Log File Agent uses the information that is specified in the configuration file to process log file information.

Table 1 explains that parameters that you can modify in this file.

Table 46. Parameters for LFA configuration file		
Required for	Parameter	Description
All	LogSources	Specify the data source that you want to monitor. If you are specifying multiple data sources, they must be comma-separated and without spaces. When you configure a remote directory in the LFA configuration file, the directory you specify must not contain any subdirectories.
Internal LFAs installed on remote servers	ServerLocation,	Specify the server location for the EIF receiver server. For example, for a server that is at 111.222.333.444, specify the following value: ServerLocation=111.222.333.444
Internal LFAs installed on remote servers	ServerPort	Specify the port that the EIF receiver uses. For example: ServerPort=5529

Table 46. Parameters for LFA configuration file (continued)

Required for	Parameter	Description
Internal LFAs installed on remote servers	BufEvtMaxSize	<p>Specify the maximum buffer size for the LFA. This parameter is the maximum size that the cache is allowed to be. If the cache is full, events are dropped and performance can decline. The value that you enter here is in kilobytes. For example:</p> <pre>BufEvtMaxSize=102400</pre>
External LFAs installed on remote servers	SshAuthType	<p>You must set this value to either PASSWORD or PUBLICKEY.</p> <p>If you set this value to PASSWORD, IBM Operations Analytics - Log Analysis uses the value that is entered for SshPassword as the password for Secure Shell (SSH) authentication with all remote systems.</p> <p>If you set this value to PUBLICKEY, IBM Operations Analytics - Log Analysis uses the value that is entered for SshPassword as pass phrase that controls access to the private key file.</p>
External LFAs installed on remote servers	SshHostList	<p>You use the SshHostList value to specify the hosts where the remotely monitored log files are generated. IBM Operations Analytics - Log Analysis monitors all the log files that are specified in the LogSources or RegexLogSources statements in each remote system.</p> <p>If you specify the local machine as a value for this parameter, the LFA monitors the files directly on the local system. If you specify that the localhost SSH is not used to access the files on the system, IBM Operations Analytics - Log Analysis reads the files directly.</p>

Table 46. Parameters for LFA configuration file (continued)

Required for	Parameter	Description
External LFAs installed on remote servers	SshPassword	<p>If the value of the SshAuthType parameter is PASSWORD, enter the account password for the user that is specified in the SshUserid parameter as the value for the SshPassword parameter.</p> <p>If the value of the SshAuthType parameter is PUBLICKEY, enter the pass phrase that decrypts the private key that is specified in the SshPrivKeyfile parameter.</p>
External LFAs installed on remote servers	SshPort	You specify the TCP port that is used for SSH connections. If you do not enter anything, this value is defaulted to 22.
External LFAs installed on remote servers	SshPrivKeyfile	<p>If the value of the SshAuthType parameter is set to PUBLICKEY, enter the directory path to the file that contains the private key of the user that is specified in the SshUserid parameter as the value for this parameter.</p> <p>If the value of the SshAuthType parameter is not set to PUBLICKEY, this value is not required.</p>
External LFAs installed on remote servers	SshPubKeyfile	<p>If the value of the SshAuthType parameter is set to PUBLICKEY, enter the directory path to the file that contains the public key of the user that is specified in the SshUserid parameter as the value for this parameter.</p> <p>If the value of the SshAuthType parameter is not set to PUBLICKEY, this value is not required.</p>
External LFAs installed on remote servers	SshUserid	Enter the user name from the remote system that the agent uses for SSH authentication.

Configuring and deploying LFAs in the command line

You can use the command line to configure and deploy LFAs.

Before you begin

Before you can configure and deploy an LFA. You need to create the configuration and format files.

For more information about how configuration and format files are used, see [“LFA configuration and format files”](#) on page 190.

For more information about the required parameters in the configuration file, see [“LFA configuration file parameters”](#) on page 104.

About this task



CAUTION:

You cannot use non-ASCII characters in the installation path. The installation path cannot exceed 80 characters.

For more information, about this and about how to configure the monitoring agent in step 3 see:

[Configuring the monitoring agent](#)

Procedure

1. To configure the LFA, run the command:

```
./itmcmd config -A pc
```

where *pc* is the product code for your agent.

For example: `./itmcmd config -A lo`.

2. You are prompted to supply the following information:

Enter instance name (default is:):

Enter the instance name. For example, *rhelagent*.

Conf file (default is:):

Enter the configuration file path. For example, */unity/IBM/ITM/config/lo/*.

Format File (default is:):

Enter the format file path. For example, */unity/IBM/ITM/config/lo/*.

Note: All fields must be completed. Blank fields can cause the LFA to fail.

3. Where prompted, provide the monitoring agent configuration information.
4. To start the LFA, run the command

```
./itmcmd agent -o instance name start lo
```

Configuring LFA subnodes

Create an LFA subnode to group an explicit set of configurations that the LFA uses to identify and process a log event.

About this task

The subnode consists of a format (*.fmt*) file and a configuration (*.conf*) file. A single instance of the LFA can have multiple subnodes. Each subnode behaves like a single thread that is running in the same instance of the LFA.

You can create subnodes for the following use cases:

Improve performance by making generic format settings more specific

To improve overall performance, you can create specific configurations to replace more generic ones. For example, you can specify the same regular expression (REGEX) in a generic *.fmt* file to parse both WebSphere Application Server (WAS) and DB2 log files. However, as the content of the log files differs, this operation can become inefficient. To improve performance, replace the single *.fmt* files with two new files that contain two specific REGEXs for WAS and DB2 in two new subnodes.

Improve performance by making generic configuration settings more specific

Similarly, you can improve performance by replacing generic configurations with more specific ones. For example, you can specify the same rollover behavior in a generic *.conf* to process both WAS and

DB2 log files. However, as the rollover behavior in the log files differs, this configuration results in some of the logs not being processed correctly and is inefficient. To improve performance, replace the single `.conf` with two new files in two new subnodes.

Improve performance for many data sources

If you use many data sources to monitor the log events, you can create subnodes to spread the workload.

Remote monitoring with IBM Tivoli Monitoring Log File Agent 6.3

With IBM Tivoli Monitoring Log File Agent 6.3, you can modify the `.conf` file to monitor logs from multiple remote sources. However, the user credentials might not be the same for the remote servers. You can maintain only one set of user credentials in the LFA configuration file. In this case, you create multiple subnodes with different user credentials in each. You can use this feature to monitor multiple remote sources from a single LFA node with multiple subnodes.

There is also a limitation on the naming of subnodes. For more information, see [“Character limits for subnode names”](#) on page 201.

Procedure

1. Go to the directory where the LFA is installed.
For example, if you are using the internal LFA that is delivered with Log Analysis, the directory is `<HOME>/IBM/LogAnalysis/IBM-LFA-6.30/bin/`.
2. To open LFA configuration window, run the following command:

```
bin/CandleManage
```
3. Right-click the **Tivoli Log File Agent** service and click **Configure**.
4. Click the instance that you want to configure and click **OK**. The **Configure Tivoli Log File Agent** window is displayed.
5. On the **Log File Adapter Configuration** tab, ensure that the **Conf file** and **Format File** fields are blank.
6. Click the **Log File Adapter Global Settings** tab and note the directory that is specified in the **Configuration file autodiscovery directory** field. This path is the directory where you save the subnode configuration files. The default value is `${CANDLE_HOME}/config/lo`. Click **OK**.
7. In the subsequent window, you can ignore the other changes and click **Save** to save your changes.
8. Enter the root user password when prompted to implement your changes.
9. Copy the subnode configuration files to the directory that you noted in step 6. The LFA automatically detects the changes. You do not need to restart the LFA instance.

Results

The procedure describes how to configure subnodes in the LFA UI. You can also use the command line. To use the command line to configure the subnodes:

1. Go to the directory where the LFA is installed.
2. Run the following command:

```
itmcmd config -A lo
```

3. Complete the required steps.
4. Specify the configuration file autodiscovery directory.
5. Complete the configuration.
6. Save the subnode configuration file to the directory that you specified in step 4.

Character limits for subnode names

When you name a subnode, ensure that you are aware of the character and naming limitations.

32 character limitation

The LFA uses msn to name and identify the subnode. IBM Tivoli Monitoring limits the length of this name to 32 characters. The limit includes the identifier, the dash, and the semi-colon. This leaves 28 new characters for the host name, subnode and configuration file name.

The subnode name is specified in the following format:

```
L0:<Hostname>_<Subnode>-<Conffilename>
```

where L0 is an identifier that is assigned to all subnodes. <Hostname> is the host name of the machine where the subnode is installed. <Subnode> is the name of the subnode. <Conffilename> is the name of the subnode configuration file.

For example:

```
L0:nc1234567890_WASInsightPack-1fawas
```

However, IBM Tivoli Monitoring limits the length of this name to 32 characters. The example name is 35 characters long. The limit includes the identifier, the dash, and the semi-colon, leaving 28 characters for the host name, subnode and configuration file name. To work around this limitation, IBM Tivoli Monitoring renames the subnode as:

```
L0:nc1234567890_WASInsightPack-1
```

This name is 32 characters long. The host name uses 12 characters. The subnode uses 14 characters.

This limitation can cause an issue if you use similar names for the configuration files. For example, after you name the first subnode, you create another subnode called:

```
L0:nc1234567890_WASInsightPack-1fawas2
```

IBM Tivoli Monitoring renames this subnode as:

```
L0:nc1234567890_WASInsightPack-1
```

As you can see, due to the truncation, both subnodes now have the same name, meaning that the LFA will not detect the new configuration.

Increasing the limit

The host name is used for integrations with Tivoli Endpoint Manager, where it helps you to identify subnodes. However, this is not required for Log Analysis. You remove the host name from the default naming convention so that you can use all 28 characters for the configuration file name.

To change the host name setting:

1. Stop the LFA.
2. Open the <HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo/lo_default_workload_instance.conf file.
3. Change the default value for the following property from Y (Yes) to N (No):

```
CDP_DP_USE_HOSTNAME_IN_SUBNODE_MSN='N'
```

4. Save the updated file.
5. Restart the LFA.

After you change this configuration setting, the subnode name no longer includes the host name. For example, the subnodes in the previous example are now named L0:WASInsightPack-1fawas and L0:WASInsightPack-1fawas2.

Common LFA configuration conflicts

When you create a remote LFA node and a custom data source and both use the same log path, you can create a conflict.

When you create a custom data source and use it to monitor a directory on a remote LFA subnode and you later create another data source, like a remote data source, that monitors the same directory, you can create a conflict in the LFA configuration. These conflicts may cause errors in the Log Analysis log files and reduce the performance of Log Analysis.

The following example is provided to help you to understand this situation.

To avoid these conflicts, you need to avoid monitoring the same directory with different data sources. If you want to monitor two files in the same directory, include the file name in the **Log Path** field when you create the data source.

Example

For example, you are an administrator and you want to monitor files from an LFA that is installed on a remote server as described in the Knowledge Center documentation. See [Streaming data with a remote LFA](#). In this case, the LFA is not part of the Log Analysis product.

First, you must create a custom data source called Customdatasource to load data from remote instance of the LFA. In the Data Source creation wizard, you specify the host name and the following log path:

```
/opt/WAS/WAS_logs/myLogFile.log
```

Next, you need to create the configuration and format files for the LFA sub nodes. You create two files, lfa1.conf and lfa1.fmt. In the lfa1.conf file, you specify the following data source:

```
Datasources=/WAS/WAS_logs/some_dir/*
```

Logs that are subsequently generated or appended are ingested by the Datasource1 data source.

After some time, you create another data source to load data from the same remote server. The new log file is called newLogFile.log and it is located in the same directory as the file that you created the Customdatasource data source for. You create a remote data source called Remotedatasource and specify the log path as:

```
/opt/WAS/WAS_logs/newLogFile.log
```

Finally, you push the log files into Log Analysis.

However, after you push the log file, you notice some strange behaviour in the Log Analysis log files. The GenericReceiver.log log file shows that the data is being ingested for /opt/WAS/WAS_logs/newLogFile.log. However, it also says that the /opt/WAS/WAS_logs/newLogFile.log log file is not a valid data source.

This occurs because the same log file is being monitored by both data sources. As a result, it is monitored by two different LFA sub nodes and in two different streams. The data is loaded but this can waste resources and decrease the overall performance.

To avoid this situation, you must be aware of any possible conflicts especially when you create a custom data source that monitors a directory rather than a file.

LFA example: Streaming data from a single remote host

In this example, you can learn how to configure Log Analysis to stream data from multiple sources on a single remote server with a single data source in Log Analysis.

Assumptions

This example assumes that you are streaming multiple WebSphere Application Server system logs from a remote location on a single remote server.

Create the custom data source and configuration files

1. Create a custom data source with the fields outlined in the following table:

Table 47. Fields for the Was-Aggregated-DS data source	
Field	User entry
Name	Was-Aggregated-DS
Type	Custom
Host Name	<my_remote_host>.example.com
Log path	/opt/datasources/WAS.SystemOut.log

2. Go to the <HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo directory and create two files called Was-Aggregated-DS.conf and Was-Aggregated-DS.fmt.
3. Copy the contents of the sample files to these new files. For more information, see [“Configuration template for LFA example”](#) on page 204 and [“Format file template for LFA example”](#) on page 206.

Configure the .conf file

Next, you need to configure the configuration file as specified in the following table:

Table 48. Properties for the Was - Aggregated - DS . conf file	
Property	User action
Logsources=	<p>First, you need to edit the Logsources= property. You can use this property to add multiple file paths. Use a comma to separate each file path.</p> <p>Note: Use only a comma with no spaces.</p> <p>You can specify paths to directories where the logs are stored. For example, /WAS1/logs,WAS2/logs,WAS3/logs.</p> <p>You can also specify paths to specific log files. For example, /opt/datasources/WAS/SystemOut.log,/opt/WAS/server1/SystemOut.log,/opt/WAS/server2/SystemOut.log,/opt/WAS/server3/SystemOut.log</p> <p>Alternatively, you can use a wildcard search to monitor all the files in a directory. For example, /opt/datasources/WAS/*.</p>
ServerLocation=LA_FQ_HOST_NAME	Replace the placeholder LA_FQ_HOST_NAME with the host name for Log Analysis.
#SshAuthType=PASSWORD	Remove the comment from this property.
#SshHostList=localhost	Remove the comment from this property and replace localhost with the IP address of the remote server that your data sources collect data from.
#SshPort=	Remove the comment and add 22 as the value.
#SshUserid=	Remove the comment and add the user ID that you want to use to access the remote server.

Table 48. Properties for the Was-Aggregated-DS.conf file (continued)

Property	User action
#SshPassword=	Remove the comment and add the password for the user that you already specified.
BufEvtPath=./opt/CacheFiles/ DS_NAME.cache	Create a directory that is called ./opt/ CacheFiles to store the cache files. Log Analysis creates a cache file in the format DS_NAME.cache. In this example, the cache file is WAS- Aggregated-DS.cache. Replace the value for BufEvtPath with the cache file name and full path. For example, ./opt/CacheFiles/WAS- Aggregated-DS.cache.

Review the .fmt file

The Was-Aggregated-DS.fmt file contains the following code that you need to update:

```
REGEX AllRecords
(.*?)
hostname myRemoteHost.ibm.com
-file /opt/datasources/WAS/SystemOut.log
RemoteHost ""
logpath PRINTF("%s",file)
text $1
END
```

You need to replace myRemoteHost.ibm.com with the same value that you specified in the data source. You also need to replace /opt/datasources/WAS/SystemOut.log with the log path that you specified in the data source.

This configuration ensures that the host name and log path are the same as the one that you specified in the format file for all the physical data sources.

Log Analysis loads this data as if it is sent from a single data source. This configuration effectively aggregates the data from multiple physical data sources into a single logical data source in Log Analysis.

Configuration template for LFA example

Use this template file to help you to stream data from a remote server with a single Log Analysis data source.

Copy the following text:

```
# Files to monitor. The single file /tmp/regextest.log, or any file like /tmp/foo-1.log
or /tmp/foo-a.log.
LogSources=/opt/datasources/*

# If more than one file matches the pattern /tmp/foo-?.log above, monitor only the newest one
FileComparisonMode=CompareByAllMatches

# Our EIF receiver host and port. Only needed when sending events directly to OMNIBus or TEC
via EIF.
# That is configured through either the Manage Tivoli Enterprise Monitoring Services GUI or the
# "itmcmd config -A lo" command.
ServerLocation=LA_FQ_HOST_NAME
ServerPort=5529

# Must be set to either PASSWORD or PUBLICKEY. If set to PASSWORD, the value of SshPassword is
treated as
# the password to be used for SSH authentication with all remote systems. If set to PUBLICKEY,
the value
# of SshPassword is treated as the pass phrase that controls access to the private key file. If
set to
# PUBLICKEY, SshPrivKeyfile and SshPubKeyfile must also be specified.
#SshAuthType=PASSWORD

# A comma-separated list of remote hosts to monitor. All log files that are specified in the
LogSources or
```

```

# RegexLogSources statements are monitored on each host that is listed here. If one of the host
names specified
# is localhost, the agent monitors the same set of files directly on the local system. When you
specify localhost,
# SSH is not used to access the files on the local system, the log files are read directly.
#SshHostList=localhost

# A TCP port to connect to for SSH. If not set, defaults to 22.
#SshPort=

# The user name on the remote systems which the agent uses for SSH authentication.
# SshUserId=

# When the value of SshAuthType is PASSWORD, this value is the account password of the user
that is specified in SshUserId.
# You can supply the account password in clear text, or you can supply a password that is
encrypted with the IBM Tivoli Monitoring
# itmpwdsnmp command. For more information about how to encrypt a password by using
theitmpwdsnmp command,
# see http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc\_6.3/logfile/
klo_encrypt.htm
# When the value of SshAuthType is PUBLICKEY, this value is the pass phrase that decrypts the
private key that is specified by
# the SshPrivKeyfile parameter. You can supply the pass phrase in clear text, or you can supply
a pass phrase that is encrypted
# with the IBM Tivoli Monitoring itmpwdsnmp command. For more information about how to encrypt
a password by using theitmpwdsnmp
# command, see http://pic.dhe.ibm.com/infocenter/tivihelp/v61r1/topic/com.ibm.itm.doc\_6.3/
logfile/klo_encrypt.htm
#SshPassword=

# To enable using Fully Qualified Domain information, set this flag
FQDomain=yes

# Interval, after which LFA polls to check for new log files
NewFilePollInterval=30

# If sending events through EIF, and the EIF receiver is down, cache events in a file until the
# receiver is available again.
BufferEvents=YES

# If caching EIF events, they are stored in this file until the receiver is available again.
BufEvtPath=/opt/CacheFiles/DS_NAME.cache

# If sending events through EIF, the maximum size the EIF cache file can grow to. If this is
too small,
# events will be dropped when the cache fills, and performance can suffer as well. Size is in
KB.
BufEvtMaxSize=102400

# If sending events through EIF, hold open a TCP connection to the EIF receiver all the time
rather
# than creating it and breaking it down each time. This is generally faster.
ConnectionMode=CO

# Check the monitored files for new data every 3 seconds.
PollInterval=3

# If the agent is stopped, save the last known position in the monitored logs and resume from
there
# when the agent restarts. This way events that are written while the agent is down are still
picked
# up when it resumes.
NumEventsToCatchUp=-1

# Monitor the named event logs on Microsoft Windows. The latter two require Windows 2008 or
higher,
# and the fourth one requires the Hyper-V role.
WINEVENTLOGS=System,Security,Application,Microsoft-Windows-Hyper-V-Worker-Admin,Microsoft-
Windows-TaskScheduler-Operational

# If running on Microsoft Windows 2008 or higher, use the new event log interface. This is
required to
# access the new event logs introduced in that version, such as the last two logs listed in the
# WINEVENTLOGS statement just above.
UseNewEventLogAPI=y

# For events sent to ITM (does not apply to EIF events), consider two events to be duplicates
if the
# "msg" and "CustomSlot1" attributes match across the two events.
DupDetectionKeyAttributes=msg,CustomSlot1

```

```
# Track duplicates over a 1 minute period, sending a summary event to ITM (does not apply to
EIF) at
# the end of every 1 minute. The summary event indicates how many duplicate events were
detected during
# the interval.
EventSummaryInterval=60

# When duplicate events are detected, send only the first one to ITM (does not apply to EIF)
during the
# summary interval (1 minute, specified above). At the end of the summary interval, the
summary event
# will indicate how many further duplicates were seen during the interval.
EventFloodThreshold=send_first

# On AIX only, monitor all events written to the Error Report (errpt) facility since agent
startup.
AIXErrptCmd=errpt -c -smddhmmmyy
```

Format file template for LFA example

Use this template file to help you to stream data from a remote server with a single Log Analysis data source.

Copy the following text:

```
REGEX AllRecords
(.*?)
hostname myRemoteHost.ibm.com
-file /opt/datasources/WAS/SystemOut.log
RemoteHost ""
logpath PRINTF("%s",file)
text $1
END
```

LFA example: Configuring LFA with SSH for remote streaming

Use this example to help you to use the LFA to stream data over a secure connection.

Assumptions

- You are streaming data from a remote server.
- You are using the remote installation tool to install a remote instance of the internal LFA.
- You must create one data source in Log Analysis for each log file that you want to monitor. For more information about how to aggregate data from multiple sources in a single data source in Log Analysis, see [“LFA example: Streaming data from a single remote host” on page 202.](#)

Set up Secure Shell authentication

First, you need to configure Secure Shell (SSH) authentication. For more information, see [“Setting up Secure Shell to use key-based authentication” on page 53](#)

To configure SSH, complete the following steps:

1. Generate the public and private key pair:

```
ssh-keygen -t rsa
```

2. Enter the file that you want to use to save the key pair.
3. Enter a password and reenter the password. If you do not want to use a password, enter a blank.

In this example, the identification is saved in `home/scala/.ssh/id_rsa..` The public key file is saved in the `home/scala/.ssh/id_rsa.pub`. The key fingerprint is:

```
73:55:7d:40:03:84:21:7e:9f:a3:40:e8:e2:38:20:f8 la@la132.example.com
```

The key's randomart image is:

```
+--[ RSA 2048 ]-----+
|      . .+oo=0  |
|     o ..  .o|
```

```

| . . . o . . . |
| . . . . S . . |
| + . . . . + |
| . o o . . + . . |
| E . . . . |
| . . . . |
+-----+
[la@la132 Desktop]$

```

Use `ssh-copy-id` to copy the public key to the authorized keys of the remote server:

```

[la@la132 Desktop]$ ssh-copy-id netcool@123.456.789.101
The authenticity of host '123.456.789.101 (123.456.789.101)' can't be established.
RSA key fingerprint is 69:e6:19:d7:9c:cc:68:ff:85:64:1d:d5:d8:5b:90:d2.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '123.456.789.101' (RSA) to the list of known hosts.
netcool@192.168.100.145's password:

```

Log in to `netcool@123.456.789.101` and verify that the files are in the `.ssh/authorized_keys` file.

To verify that the keys are on the local server, where Log Analysis is installed, go to the `.ssh` directory and run the following command:

```

ls
id_rsa id_rsa.pub known_hosts

```

The following response is displayed:

```

cat id_rsa.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEA0Wc0xMeFXogGr0xKJXV1YGy4pw2dmpnKHfE2lQ
YpuZpKyOhP3lSm6GPUVBGRtFATBG2WwifD5Xj1wUdheZYQ1UxeFP8A97d8PEhfVADI
gG8rch3bLfrMlURwoxHimDBb1SU4oTA/gKoJdlr5XAVbtYqYGFese1nqq0ZCR24kj0h
/o/PAQ/lzF+0C+H4aYUkOnW5l1JJPo7Xl70MlsEcgy0wRUPTapjY2QCx1V908w8yp5
kZBHNIPCMcNPp7hgZCW3k97R4mRCPqZByyRHWSE2h1ngA12Mbsm0ZiW7kU56/A0eg8+
ZcPYiqougY5wWVCnBZc/JMo0p+o2io1JQVQ== la@la132.ibm.com

```

To verify that the keys are on the remote server, log in to the remote server, go to the `.ssh` directory and enter the following command:

```

ls
authorized_keys

```

The following response is displayed:

```

cat authorized_keys
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEA0Wc0xMeFXogGr0xKJXV1YGy4pw2dmpnKHfE2lQ
YpuZpKyOhP3lSm6GPUVBGRtFATBG2WwifD5Xj1wUdheZYQ1UxeFP8A97d8PEhfVADI
gG8rch3bLfrMlURwoxHimDBb1SU4oTA/gKoJdlr5XAVbtYqYGFese1nqq0ZCR24kj0h
/o/PAQ/lzF+0C+H4aYUkOnW5l1JJPo7Xl70MlsEcgy0wRUPTapjY2QCx1V908w8yp5
kZBHNIPCMcNPp7hgZCW3k97R4mRCPqZByyRHWSE2h1ngA12Mbsm0ZiW7kU56/A0eg8+
ZcPYiqougY5wWVCnBZc/JMo0p+o2io1JQVQ== la@la132.ibm.com

```

It is important that you assign the correct permissions to `.ssh` directory and `id_rsa.pub` files. For this example, you must assign permissions as described here:

```

-rw----- 1 scala scala 1671 Oct 28 08:25 <id_rsa>
-rw-r--r-- 1 scala scala 404 Oct 28 08:25 <id_rsa>.pub
-rw-r--r-- 1 scala scala 397 Oct 28 08:28 <known_hosts>

```

where `<id_rsa>` is the identification file. `<id_rsa>.pub` is the public key file. `<known_hosts>` is the host name for the Log Analysis server.

To assign the required permissions on the remote server, assign permissions as described here:

```

-rw----- 1 netcool netcool 404 Oct 28 08:56 <authorized_keys>

```

where `<authorized_keys>` are the authorized keys.

Configure the LA remote installation tool

After you configure SSH, you need to use the remote installation tool to install the LFA on the remote server. For more information see, [“Deploying the LFA or EIF on remote servers”](#) on page 215.

Edit the following parameters in the <HOME>/IBM/LogAnalysis/remote_install_tool/config/ssh-config.properties file:

```
REMOTE_HOST=192.168.100.145
PORT=22
TIME_OUT=60000
USER=netcool
PATH_OF_PASSWORD_LESS_SSH_KEY=/home/scala/.ssh/id_rsa
```

You do not have to enter a password as in this example, as you did not set one when you set up SSH.

Installing the LFA remotely

To install the LFA, go to the <HOME>/IBM/LogAnalysis/remote_install_tool/ and run the following command:

```
./install.sh
```

The following response is displayed, you enter y when prompted:

```
Init in progress...
[/opt/IBM/LogAnalysis, -install]
+++++ IBM Log Analysis Remote Deployment Tool +++++
Ssh properties is available.
SSH Configuration :      ... IN PROGRESS
SSH Configuration :      ... SUCCESS

Enter Remote Top Level Installation
Directory absolute path: [/home/netcool/LogAnalysis]

Remote path:/home/netcool/LogAnalysis
Provide your Deployment topology inputs for the following
-----

Install EIF Receiver Instances (y|Y|n|N)      :[y]
n
Install LFA 6.3 (y|Y|n|N)                    :[y]

Proceeding with default value:y
Install logstash 1.5.3 (y|Y|n|N)              :[y]
n
Start Time:Wed Oct 28 09:16:25 PDT 2015
LogAnalysis directories:
LOG_ANALYSIS_HOME_DIR :      /home/netcool/LogAnalysis
IMAGES_DIR             :      /home/netcool/LogAnalysis/images
STORE_DIR              :      /home/netcool/LogAnalysis/store
JAVA_HOME              :      /home/netcool/LogAnalysis/ibm-java

LFA 6.3 directories:
LFA_HOME_DIR           :      /home/netcool/LogAnalysis/IBM-LFA-6.30

Copy files to IMAGES and STORE                ...BEGINS
Wait while the following remote copy command takes some time to complete...
Copy files to IMAGES and STORE                ...DONE
Wait while the following remote copy command takes some time to complete...
Copy files to IMAGES and STORE                ...DONE
Extracting IBM Java ibm-java-sdk-8.0-1.0-linux-x86_64.tgz ...
Extracted java ... :
Command execution response:
LFA Prereqs validation ...IN PROGRESS

Validation success for EXISTING PROCESS CHECK
Validation success for KSH CHECK
Validation success for PREREQ LIBRARIES
Validation success for SELINUX

LFA Prereqs validation ...DONE
Setting up LFA ...
Wait while the following remote command takes some time for LFA installation ...
```



```

Setting up LFA          ...DONE
+++++
+++++ DEPLOYMENT STATUS +++++
+++++
Response:
Response: =====
Response: COMPONENT      PID      STATUS
Response: =====
Response: Log File Agent  16446      UP
Response: =====
End Time:Wed Oct 28 09:17:59 PDT 2015
+++++

Total install duration (seconds):93

+++++ Installation Ends +++++

```

Starting the LFA

After you install the remote instance of the LFA, you must start it. To start it, go to the <HOME>/IBM/LogAnalysis/utilities directory and enter the following command:

```
./lfautil.sh -start
```

For more information, see [“lfautil.sh command” on page 108](#).

Enabling the remote LFA

After you start the remote LFA, you need to create the required data source, configuration, and format files. For more information, see [“Creating data sources for the LFA” on page 191](#).

To enable the remote LFA:

1. Create a custom data source for each file that you want to load.
2. Stop the remote LFA:

```
<HOME>/IBM/LogAnalysis/utilities/lfautil.sh -start
```

3. Create the configuration (.conf) and format (.fmt) files. You can create these files yourself or you can copy the sample files in the <HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo directory to the remote server.
4. Configure the configuration and format files. Ensure that the LogSources value matches the data sources that you created in step 1.

Regular expression support for the LFA

The LFA supports specific implementations of regular expressions.

Single-line unstructured data

If you want to use the DSV toolkit to extract and export the data in the comma-separated value (CSV) format for use with the DSV toolkit, you can use a regular expression to extract and export the data.

For example, consider the following log file record:

```

10453072 23460 E5D27197E653C548BDA744E8B407845B A0BEAI1 /EAI I H R SACP9002
BPUSRSYS/612 23460 - XGNEA108:662:000042:06:E036977:WWS00003:7000:16:1:REV=N
Proc Time=000.03

```

You can configure Log Analysis to use a regular expression to extract and export the data in the comma-separated value (CSV) format. For example, here is an example of a regular expression that is defined in the .fmt file:

```

REGEX EAILOG
^([0-9]*)(.*)SACP9002(.*) : ([0-9]*) : ([0-9]*) : ([0-9]*) : ([a-zA-Z0-9]*) :
([a-zA-Z0-9]*) : ([a-zA-Z0-9]*) :
(.*)Proc Time=([0-9]*.[0-9]*)

```

```

timestamp $1 CustomSlot1
discard $2
SACP9002 $3
bankID $4 CustomSlot3
branchID $5 CustomSlot4
discard3 $6
tellerSID $7 CustomSlot5
workstationID $8 CustomSlot6
transactionTypeID $9 CustomSlot7
discard4 $10
responseTime $11 CustomSlot8
msg PRINTF("%s,%s,%s,%s,%s,%s,%s",timestamp,bankID,branchID,tellerSID,workstationID,
transactionTypeID,responseTime)
END

```

Manipulating date time information for the Generic Annotation Insight Pack

If you use the Generic Annotation Insight Pack or the date time rule set from the Generic Annotation Insight Pack in a custom Insight Pack, you can use some limited regular expressions that you can use to parse time and date information.

The second delimiter, which is a colon (:), is not supported. The regular expression replaces the second delimiter with a period (.), which is supported. For example, to change a date from 15/12/2014 12:12:12:088 GMT to 15/12/2014 12:12:12.088 GMT, you can add the following regular expression to the .fmt file:

```

// Matches records for any Log file:
// Log Analytics Data Source chas_access.log

REGEX nongr
([0-9][0-9])/([0-9][0-9])/([0-9][0-9]) ([0-9][0-9]):([0-9][0-9])
:([0-9][0-9]):([0-9][0-9][0-9]) ([A-Z][A-Z][A-Z])
(.*Batch Status for.*)
month $1
day $2
year $3
hour $4
minute $5
second $6
ms $7
zone $8
message $9
hostname example.com
-file /opt/la/IBM/LogAnalysis/logs/GenericReceiver.log
RemoteHost ""
logpath PRINTF("%s",file)
text PRINTF("%s/%s/%s %s:%s:%s %s %s", month, day, year, hour, minute,
second, ms, zone, message)
END

```

Troubleshooting data loading

When you are using the IBM Tivoli Monitoring Log File Agent to perform data collection, monitor the UnityEIFReceiver.log and GenericReceiver.log log files located in the <HOME>/logs directory to ensure that the data ingestion has completed correctly.

This example illustrates the addition of a batch of log records. The result is indicated in the RESPONSE MESSAGE section of the log file:

```

~~~~~
2013-04-20 04:43:10,032 [pool-5-thread-1] INFO - LogEventPoster : -----
Posting Event to UNITY DATA COLLECTOR -
https://nc9118041070:9987/Unity/DataCollector
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO - LogEventPoster :
+++++++ RESPONSE MESSAGE ++++++++
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO - LogEventPoster : OK
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO - LogEventPoster :
{
  "batchSize": 2078,
  "failures": [ ], "numFailures": 0 }
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO - LogEventPoster :
+++++++
2013-04-20 04:43:24,273 [pool-5-thread-1] INFO - LogEventPoster :

```

```
EIF event delivery to Generic Receiver -- SUCCESS
```

In this log, the number of log records processed is indicated in the line:

```
{  "batchSize": 2078,  "failures": [  ],  "numFailures": 0 }
```

2078 log records were successfully ingested. The numFailures value indicates the number of failures in the ingestion of the log records. For example, a value of 5 for the numFailures value indicates that 5 log records were not ingested.

When data collection has completed, if the EIF Receiver buffer is partially filled, any remaining log records are posted to the Generic Receiver. This is recorded in the log as a TIMEOUT FLUSH event. These events are added to the log file at the end of the session of data collection:

```
~~~~~
2013-04-20 04:54:26,341 [pool-4-thread-1] INFO - LogEventService :
  TIMEOUT FLUSH for datasource:nc9118041070::
  /home/yogesh/IBM/LogAnalysis/logsources/WASInsightPack/TipTrace5.log
2013-04-20 04:54:26,359 [pool-5-thread-1] INFO - LogEventPoster : -----
Posting Event to UNITY DATA COLLECTOR -
  https://nc9118041070:9987/Unity/DataCollector
2013-04-20 04:54:38,581 [pool-5-thread-1] INFO - LogEventPoster :
  ++++++++ RESPONSE MESSAGE ++++++++
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO - LogEventPoster : OK
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO - LogEventPoster :
  {
    "batchSize": 1714,
    "failures": [  ], "numFailures": 0 }
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO - LogEventPoster :
  ++++++++
2013-04-20 04:54:38,582 [pool-5-thread-1] INFO - LogEventPoster :
  EIF event delivery to Generic Receiver -- SUCCESS
2013-04-20 04:54:38,583 [pool-4-thread-1] INFO - LogEventService :
  POST RESULT:
  {"failures": [], "batchSize":1714, "numFailures":0}
~~~~~
```

To calculate the number of events that have been processed, calculate the sum of all of the batchSize values. To calculate the number of events ingested, calculate the sum of all of the batchSize values and deduct the total sum of numFailure values.

If the ingestion fails, an error message is recorded in the UnityEIFReceiver.log:

```
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO - LogEventPoster :
  ++++++++ RESPONSE MESSAGE ++++++++
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO - LogEventPoster : Not Found
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO - LogEventPoster :
  {"BATCH_STATUS":"NONE","RESPONSE_MESSAGE":
  "CTGLA0401E : Missing data source ","RESPONSE_CODE":404}
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO - LogEventPoster :
  ++++++++
2013-05-16 02:16:11,184 [pool-7-thread-1] INFO - LogEventPoster :
  FAILURE - ResponseCode:404 ResponseMessage:Not Found
```

Additional HTTP response codes are as follows:

413

Request Entity Too Large: Displayed if a batch size is greater than the Generic Receiver default value set in the \$UNITY_HOME/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties.

500

Internal Server Error: Displayed when there is any issue with IBM Operations Analytics - Log Analysis such as a database error or any other runtime error.

404

Not Found: Displayed when a data source is not found for a hostname and log path combination in the request.

409

Conflict: Displayed if the data batch is posted for a data source that is in an inactive state or if there is a conflict between the data posted and the data expected by the server. For example, the `inputType` field in the request JSON does not match the `inputType` field in the Collection for the requested hostname and log path combination.

200

OK: Displayed when the request is processed by the server. The status of the processed batch of records is returned with the total number of records ingested, how many failed records are present and which failed.

400

Bad Request: Displayed when the request JSON does not contain the required fields expected by the Generic Receiver or where the JSON is not properly formed.

Streaming data with the IBM Performance Management OS agent

If you use IBM Performance Management 8.1.3, you can use any of the OS agents to stream data from IBM Performance Management and any servers that it monitors to Log Analysis.

The OS agents support the same functions as the IBM Tivoli Monitoring Log File Agent (LFA) that is bundled with Log Analysis. Log Analysis uses the `LogMonitoring` function in the OS agents to stream data from IBM Performance Management and any of the operating systems monitored by the agents.

For more information about the compatibility of the OS agents, see [System requirements](#).

OS agent refers to one of the three OS agents in IBM Performance Management 8.1.3, which have OS in their name:

- Monitoring Agent for Linux OS
- Monitoring Agent for UNIX OS
- Monitoring Agent for Windows OS

For more information about IBM Performance Management 8.1.3, see [IBM Knowledge Center for IBM Performance Management](#)

Configuring the IBM Performance Management OS agent to stream data

Before you can stream data from IBM Performance Management, you need to configure Log Analysis and IBM Performance Management.

Before you begin

Ensure that the following prerequisites are fulfilled:

- Use IBM Performance Management 8.1.3.
- You can install the OS agent and Log Analysis on the same server. However, ensure that they are installed in separate directories. You can also install the OS agent on a separate server.
- If you want to use the IBM Performance Management UI to distribute the configuration and format files, you must install an APM server.
- If you do not have an APM server, configure the agent in autonomous mode.
- If you use an instance of the IBM Tivoli Monitoring Log File Agent to stream data from the same server as the OS agent, shut down the LFA before you create the logical data source for the OS agent. For example, enter the following command to shut down the internal LFA:

```
itmcmd -o default_workload_instance agent stop lo
```

You cannot load data from the same physical data source with the OS agent and another data collection tool. If you are using another data collection tool, such as the LFA, to load data already from a physical data source, stop the data collection. For example, if you are using an LFA, stop the LFA.

For more information about agents, see [Agent deployment](#).

About this task

You can move existing LFA configuration and format files or you can create new ones, automatically or manually.

You can configure this integration between existing OS agents or you can install new OS agents.

You can install the OS agent and the LFA on the same server as each other and Log Analysis. However, install them in separate directories. Ensure that the LFA is stopped before you configure the OS agent.

The logs for troubleshooting are stored in the `<OS_agent_install_dir>/logs/` directory. For more information, see [Collecting monitoring agent logs for IBM support](#).

Procedure

1. Create a data source in Log Analysis.

If you create a local or remote data source, the configuration and format files are automatically generated and stored in the `<HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo` directory.

If you create a custom data source, you need to manually create these files. Note the directory where you saved the configuration and format files.

2. Move the configuration and format files from the directory where you saved them to IBM Performance Management.. For example, from the `<HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo` directory.

Ensure that you move the files out of the directory rather than copying them. Moving the files ensures that only the OS agent is monitoring the files and helps to prevent duplicated log file records

For Windows servers, move the files to the `C:\IBM\APM\localconfig\nt\log_discovery` directory on the IBM Performance Management server.

For UNIX or Linux servers, move the file to the `<OS_agent_install_dir>/localconfig/<agent_2letter_code>/log_discovery/` directory on the IBM Performance Management server. `<OS_agent_install_dir>` is the directory where the agent is installed. `<agent_2letter_code>` is the two letter code that is used to identify the agent.

Alternatively, you can use the IBM Performance Management UI to import the files. Files which are added manually to the `<OS_agent_install_dir>/localconfig/<agent_2letter_code>/log_discovery/` directory are not displayed on the IBM Performance Management UI.

3. Start the OS agent.

Results

IBM Performance Management uses the `ServerPort` and `ServerLocation` parameters that are specified in the configuration file to connect and stream data to Log Analysis.

Streaming data from multiple remote sources across a network

To facilitate dynamic data streaming that is scalable across multiple remote sources, you can configure the internal LFA that is installed with Log Analysis or the EIF Receiver to stream data from remote servers.

To enable data collection from remote hosts, you must complete the following steps:

1. Install Apache Solr on the remote machine. This step is optional but it can help optimize performance. For more information, see [“Installing Apache Solr on remote machines”](#) on page 64.
2. Set up Secure Shell (SSH) communication to use key-based authentication. For more information, see [“Setting up Secure Shell to use key-based authentication”](#) on page 53
3. Configure SSH to work with the remote installer utility. For more information, see [“Configuring secure shell \(SSH\) communication for multiple remote hosts”](#) on page 214.

4. Use the remote installer utility to install instances of the Event Integration Facility (EIF) or the IBM Tivoli Monitoring Log File Agent (LFA) on remote machines. For more information, see [“Deploying the LFA or EIF on remote servers”](#) on page 215.
5. Configure the EIF so that it is compatible with the remote instances that you create. If you use the LFA, you do not have to configure the local installation. However, you do have to manually configure the sub nodes. For more information, see [“Configuring LFA subnodes”](#) on page 199.

You can also maintain and administer these connections after you set them up.

As an alternative to streaming data, You can batch load data. For more information, see [“Loading batches of data”](#) on page 178.

Configuring secure shell (SSH) communication for multiple remote hosts

Before you can use the remote installer utility, you must configure the SSH for the remote hosts.

Before you begin

Before you configure SSH for multiple remote hosts, you must configure SSH between IBM Operations Analytics - Log Analysis and the remote hosts. For more information, see the *Setting up Secure Shell to use key-based authentication* topic in the Information Center.

About this task

By default, the SSH properties file, `ssh-config.properties`, is in the `<HOME>/IBM/LogAnalysis/remote_install_tool/config` directory. If you save the file to another location, the utility requests that the user enters values for the remote host, user, and password. In this case, the utility does not use the values specified in the file.

If you save the `ssh-config.properties` file in the `<HOME>/IBM/LogAnalysis/remote_install_tool/config` directory, the `eif_remote_install_tool` utility uses the properties specified in the file.

The SSH communication can be based on a password or public key. If you implement both options, the public key configurations are used by default.

To set up password based SSH, you specify a value for the password and comment out the private key file. The remote installer utility uses the specified password for SSH authentication when it connects to the remote host.

To set up private key file based authentication, you specify the private key file path and do not specify a value for the password. The utility uses the file to create a password-less SSH connection.

If you specify values for both the password and the private key file path, the utility uses the file to create a password-less SSH connection.

If you do not specify a value for the password or the private key file path, IBM Operations Analytics - Log Analysis cannot create a connection and instead generates an error message in the log:

```
ERROR:
example.unity.remote.SshConfigException:
Property file config/ssh-config.properties must contain at least one of:
PASSWORD, PATH_OF_PASSWORD_LESS_SSH_KEY
Correct SSH configuration OR reconfigure and retry
Installation Aborted....!
```

Procedure

1. Navigate to the `<HOME>/IBM/LogAnalysis/remote_install_tool/config` directory and open the `ssh-config.properties` file.
2. Specify values for the following properties for each remote host:
 - Remote host
 - Remote user ID

- Port
- Connection timeout in milliseconds. The default is 6000.

For example:

```
REMOTE_HOST=<REMOTE_HOST>
PORT=<PORT>
TIME_OUT=60000
USER=<REMOTE_USER>
```

3. For password-based authentication, you also need to specify the password in the configuration file.

For example:

```
PASSWORD=password1
```

4. For public key based authentication, specify the path to the directory that contains the private key file.

For example:

```
PATH_OF_PASSWORD_LESS_SSH_KEY=/home/pass/.ssh/id_rsa
```

5. If your installation of SSH requires a passphrase, specify the passphrase.

For example:

```
PASSPHRASE_OF_PASSWORD_LESS_SSH_KEY=passphrase1
```

Deploying the LFA or EIF on remote servers

To facilitate scalable data collection on multiple remote nodes, you can deploy instances of the Tivoli Event Integration Facility (EIF) Receiver or the internal IBM Tivoli Monitoring Log File Agent on remote servers.

Before you begin

Before you run the command, you must configure secure shell (SSH) communication between the local installation of IBM Operations Analytics - Log Analysis and the remote host. For more information about how to do so, see [“Configuring secure shell \(SSH\) communication for multiple remote hosts” on page 214](#).

About this task

You can use the remote installer in the following scenarios:

- If you have a high rate of data ingestion on multiple data sources. For example, if you have 100 or more events per second and 20 or more data sources.
- If you require improved throughput performance on the remote server.
- If the hardware resources on the remote server are restrained.
- If you want to optimize performance according to the conditions described on the Performance developer works page here: https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM_Log_Analytics_Beta/page/Performance_and_tuning

You can use the command to deploy up to 20 instances of the Tivoli Event Integration Facility Receiver or a single instance of the LFA on a remote node. The command deploys and configures IBM Java 1.8. The command also configures the deployed Tivoli Event Integration Facility Receiver instance to communicate with the Data Collector client.

However, this command does not configure the LFA subnode. You must configure this setting manually. Both the remote and local instance of the LFA can monitor remote data sources. For more information about configuring LFA, see [Streaming data with a remote LFA](#).

To ensure that the remote instances of the Tivoli Event Integration Facility work with the local Data Collector interface, you must create the remotely deployed Tivoli Event Integration Facility or LFA instances as part of the same installation. This is because the encryption configuration and signature

generation is done during the main installation. If you install IBM Operations Analytics - Log Analysis after you set up the remote nodes, you must install the remote Tivoli Event Integration Facility or LFA instances again. However, you can remove remote instances of the Tivoli Event Integration Facility or LFA without installing IBM Operations Analytics - Log Analysis again.

Note:

If you use the script to install the remote instance on a server that uses the SUSE Linux Enterprise Server 11 operating system, the script fails. To resolve this issue, see the *Cannot install remote EIF instance on SUSE* topic in the *Troubleshooting* guide.

Note:

The remote installer that you use to install instances of the IBM Tivoli Monitoring Log File Agent and the Tivoli Event Integration Facility does not support cross operating system integration. You must use the remote installers to install remote instances on servers that use the same operating system. For example, if you install IBM Operations Analytics - Log Analysis on Linux on System z, you must install the remote instances on Linux on System z. In this example, you cannot install remote instances on Linux on System x.

The `install.sh` command is in the `<HOME>/IBM/LogAnalysis/remote_install_tool/` directory on the local installation of IBM Operations Analytics - Log Analysis. Use the `install.sh` command to install the Tivoli Event Integration Facility Receiver or the LFA on a remote server.

Procedure

1. Navigate to the `<HOME>/IBM/LogAnalysis/remote_install_tool/` directory and run the `install.sh` command. You are prompted for a series of inputs.
2. Enter the remote installation directory. This value must be the location where the deployed artifacts are installed on the remote host.
3. If you want to deploy the Tivoli Event Integration Facility Receiver, select it. If you do, enter the Tivoli Event Integration Facility Receiver instances that you want to deploy.
4. If you want to deploy the LFA instance on the remote node, select it.
5. If you want to use the configuration or format files in the `<HOME>/IBM/LogAnalysis/IBM-LFA-6.30/config/lo` directory for your remote LFA instance, copy the files to the remote server.

Results

After you complete the procedure, you can now collect data from the remote hosts.

What to do next

After the initial setup, you will want to periodically change the configuration. IBM provides two commands to start and stop the instances so that you can update the configuration.

eifutil.sh command

To administer EIF Receiver instances, use the `eifutil.sh` command.

Syntax

The `eifutil.sh` command has the following syntax and is in the `<USER_HOME_REMOTE>/DataForwarders/EIFReceivers/utilities` where `<USER_HOME_REMOTE>` is the directory on the remote host where the EIF Receiver instances are deployed:

```
eifutil.sh -status|-start <Inst_ID>|-stop <Inst_ID>|-startAll|-stopAll|-restart
<Inst_ID>|-restartAll
```

where `<Inst_ID>` is the ID for the specific EIF instance.

Parameters

-status

Displays the status for the installed instances. For example:

COMPONENT	Instance	PID	PORT	STATUS
EIF Receiver	eif_inst_1	13983	6601	UP
EIF Receiver	eif_inst_2	14475	6602	UP
EIF Receiver	eif_inst_3	14982	6603	UP
EIF Receiver	eif_inst_4	15474	6604	UP
EIF Receiver	eif_inst_5	15966	6605	UP

-start <Inst_id>

Starts the specified instance.

-stop <Inst_id>

Stops the specified instance.

-startAll

Starts all instances.

-stopAll

Stops all instances.

-restart<Inst_id>

Restarts the specified instance.

-restartAll

Restarts all the instances.

lfautil.sh command

To administer IBM Tivoli Monitoring Log File Agent (LFA) instances, use the `lfautil.sh` command.

Syntax

The `lfautil.sh` command has the following syntax and is in the `<USER_HOME_REMOTE>/utilities/` directory on the remote host where `<USER_HOME_REMOTE>` is the directory on the remote host where the LFA instances are deployed:

```
lfautil.sh -start|-stop|-status|-restart
```

Parameters

-start

Starts all the LFA instances on the remote host.

-stop

Stops all the LFA instances on the remote host.

-status

Displays the status for the LFA instances on the remote host. For example:

COMPONENT	PID	STATUS
Log File Agent	23995	UP

-restart

Restarts the LFA instances on the remote host.

Streaming data with logstash

Installing logstash on a remote node extends IBM Operations Analytics - Log Analysis functions so it can ingest and perform metadata searches against log data that is processed by logstash.

Overview

logstash is an open source tool for managing events and logs. It can be used to collect logs, parse them, and send them to another tool such as IBM Operations Analytics - Log Analysis to store them for later use.

You can install logstash on a local host but to improve system performance, install logstash on a remote node.

The logstash agent is an event pipeline that consists of three parts:

1. Inputs
2. Filters
3. Outputs

Inputs generate events. Filters modify events. Outputs send the event somewhere. For example, events can be sent to storage for future display or search, or to the IBM Operations Analytics - Log Analysis framework. Events can have a type, which is used to trigger certain filters. Tags can be used to specify an order for event processing as well as event routing to specific filters and outputs.

logstash can be used as a "pre-processor" to analyze sources and provide a semi-structured or structured feed to IBM Operations Analytics - Log Analysis for the purposes of searching and potential usage within custom analytics applications.

Note: To install logstash on a local node, use the remote installation feature and set the SSH configuration REMOTE_HOST value to localhost or IP Address/hostname of the node. For more information about installing logstash on a remote node, see [“Installing logstash on a remote node” on page 220](#).

For more information on logstash events, see the section *the life of an event in logstash* at <https://www.elastic.co/guide/en/logstash/current/index.html>.

Versions

logstash 1.5.3 is bundled with IBM Operations Analytics - Log Analysis 1.3.3.

Fix Pack 1 logstash 2.2.1 is bundled with IBM Operations Analytics - Log Analysis 1.3.3 Fix Pack 1.

Fix Pack 1 To upgrade from logstash 1.5.3 to 2.2.1, see [“Upgrading to logstash 2.2.1” on page 219](#).

Dependencies

Supported versions of logstash and its dependencies.

Supported logstash version

The supported versions of logstash are 1.5.3 and 2.2.1.

logstash 2.2.1 is bundled with IBM Operations Analytics - Log Analysis 1.3.3 Fix Pack 1. logstash 1.5.3 is bundled with IBM Operations Analytics - Log Analysis 1.3.3.

Java version requirement

Open JDK 1.7.0_51 or higher is required to install logstash 1.5.3. To download the correct Open JDK for your system, go to <https://openjdk.java.net/install/index.html>

It is not required for logstash 2.2.1.

DSV Toolkit requirement

DSV Toolkit v1.1.0.4 or higher for generating IBM Operations Analytics - Log Analysis Insight Packs. The Insight Packs are used to index log records that have been annotated using logstash. You only require the DSV toolkit if you want to use logstash to perform ingestion, splitting, annotating or for when the data being read by logstash is in DSV format. For more information on this user scenario, see [“Configuring logstash for rapid annotation and pre-indexing”](#) on page 224.

Generic Annotation Insight Pack

Generic Annotation v1.1.0, or v1.1.1 (refresh 1) is recommended for the normalized timestamp splitter function, which recognizes a variety of timestamps.

Fix Pack 1 Removing logstash 1.5.3

To remove logstash 1.5.3, complete this procedure.

Before you begin

Ensure that SSH is configured for remote instances of logstash. For more information, see [“Secure Shell \(ssh\) configuration for remote logstash”](#) on page 221.

About this task

If you want to use logstash 2.2.1, remove logstash 1.5.3 before you install IBM Operations Analytics - Log Analysis 1.3.3 Fix Pack 1.

Procedure

1. Run the `<HOME>/IBM/LogAnalysis/remote_install_tool/uninstall.sh` command.
2. To remove the current installation of logstash, follow the instructions.

What to do next

You can now upgrade to logstash 2.2.1.

Fix Pack 1 Upgrading to logstash 2.2.1

To upgrade from logstash 1.5.3 to 2.2.1, complete the following steps.

About this task

logstash 1.5.3 is bundled with IBM Operations Analytics - Log Analysis 1.3.3.

logstash 2.2.1 is bundled with IBM Operations Analytics - Log Analysis 1.3.3 Fix Pack 1.

Procedure

1. Save the files in the `<install_dir>/Logstash/logstash-1.5.3/` directory to another directory.
2. Remove logstash 1.5.3. For more information, see [“Removing logstash 1.5.3”](#) on page 219.
3. Install IBM Operations Analytics - Log Analysis 1.3.3 Fix Pack 1.
4. Install logstash 2.2.1. For more information, see [“Installing logstash on a remote node”](#) on page 220.
5. Add the files that you saved in step 1 to the `<install_dir>/Logstash/logstash-2.2.1/` directory.

What to do next

Although logstash 2.2.1 is compatible with all Insight Packs, there are some differences in how data is processed in it as opposed to how it is processed in logstash 1.5.3. This can change how logstash behaves during run time and can affect performance.

To review the changes between both versions, see <https://www.elastic.co/guide/en/logstash/current/breaking-changes.html> and <https://www.elastic.co/guide/en/logstash/current/upgrading-logstash-2.2.html>.

Installing logstash on a remote node

You can install Logstash on a remote node to improve system performance.

Before you begin

Ensure that the SSH user has the correct permissions for installation. For more information about SSH configuration, see “Secure Shell (ssh) configuration for remote logstash” on page 221 in the *Loading and streaming data guide*.

About this task

Logstash is processor and system resource intensive. Logstash can be installed on the local host but to improve system performance, install Logstash on a remote node.

Procedure

1. To install Logstash, run the following command:

```
<HOME>/IBM/LogAnalysis/remote_install_tool/install.sh
```

2. The installation script installs Logstash, and provides options to install the EIF receivers and log file Agent. To select each option, including Logstash, select y or Y.
3. Provide the path to the installation location on the remote host.

Results

Logstash is installed in the `<install_dir>/Logstash/logstash-<logstash_version>/` directory. To confirm the installation, logon to the remote node as the configured SSH user and go to the installation location.

Note: To install logstash on a local node, use the remote installation feature and set the SSH configuration REMOTE_HOST value to localhost or IP Address/hostname of the node.

Example

Here are some example deployments for Logstash 2.2.1:

Logstash example:

```
<install-dir>/Logstash/logstash-2.2.1/
```

Output plug-in configuration path:

```
<install-dir>/Logstash/logstash-2.2.1/logstash-scala/logstash/  
config/logstash-scala.conf
```

Output plug-in jar directory

```
<install-dir>/Logstash/logstash-2.2.1/logstash-scala/logstash/outputs
```

Here are some example deployments for Logstash 1.5.3:

Logstash example:

```
<install-dir>/Logstash/logstash-1.5.3/
```

Output plug-in configuration path:

```
<install-dir>/Logstash/logstash-1.5.3/logstash-scala/logstash/
```

```
config/logstash-scala.conf
```

Output plug-in jar directory

```
<install-dir>/Logstash/logstash-1.5.3/logstash-scala/logstash/outputs
```

Secure Shell (ssh) configuration for remote logstash

Before you can use the remote installer utility, you must configure the SSH for the remote hosts.

The `ssh_config.properties` file is in the `<HOME>/IBM/LogAnalysis/remote_install_tool/config` directory. Configure the parameter values as outlined in Table 1.

The SSH communication can be based on a password or public key. If you implement both options, the public key configurations are used by default.

To set up password-based SSH, you specify a value for the password and comment out the private key file. The remote installer utility uses the specified password for SSH authentication when it connects to the remote host.

To set up private key file-based authentication, you specify the private key file path and do not specify a value for the password. The utility uses the file to create a password-less SSH connection.

To set up command-line authentication, rename the `ssh-config` properties file or move the properties file to a new location. By default the configurations are selected from the properties file. If the file is unavailable, the user is prompted for command-line input.

Table 49. <i>ssh_config</i> parameters	
Parameter	Value
REMOTE_HOST=	<REMOTE SERVER IP/FQ HOSTNAME>
PORT=	<SSH PORT> THE DEFAULT VALUE IS 22
USER=	<SSH_USER>
PASSWORD=	<SSH PASSWORD>

logstash configuration

logstash can be configured as a log file agent to ingest logs from a number of different sources.

About this task

There are two established use cases for using logstash with IBM Operations Analytics - Log Analysis, these are:

- [Configuring logstash as an alternative to ITM LFA](#)
- [Configuring logstash for rapid annotation and pre-indexing processing](#)

Both use cases are described in this section.

Configuring logstash as an alternative to ITM LFA

logstash can be used as a log file agent to ingest logs from a number of different sources. It can also support integration with numerous alternative log file agents such as Lumberjack, Minuswell, Beaver, and Syslog.

About this task

Log records are written to the IBM Operations Analytics - Log Analysis that then sends the message to the IBM Operations Analytics - Log Analysis server for annotating and indexing.

Procedure

1. Update the logstash sample configuration file with your configuration information, `<logstash_install_dir>/LogStash/<logstash_version>/logstash-scala/logstash/config/logstash-scala.conf`. Where `<logstash_install_dir>` is path to where logstash is installed using the IBM Operations Analytics - Log Analysis remote deploy tool. `<logstash_version>` is the logstash version, 2.2.1 or 1.5.3.

- a) Define the input and filter in the logstash configuration file.

For example:

```
input {
  file {
    type => "http"
    path => ["/tmp/myhttp.log"]
  }
}
filter {
  mutate {
    replace => ["host", "<hostname>"]
    replace => ["path", "/tmp/apache.log"]
  }
}
```

Note: For Windows, the logstash file plug-in requires a drive letter specification for the path, for example:

```
path => ["c:/tmp/myhttp.log"]
```

- b) Modify the logstash configuration file to add the scala output plug-in.

The scala output plug-in buffers and sends the logstash event to the IBM Operations Analytics - Log Analysis server by using the Log Analysis server ingestion REST API. The logstash configuration file can contain one or more scala output plug-ins. The output plug-ins can be configured to write to different Log Analysis servers or to the same Log Analysis server with a different set of configurations.

Every event that is sent to the scala output plug-in must contain at least the host and path fields. The values of these fields are used by the scala output plug-in to determine the target data source for the event. Any event that does not contain either of these fields is dropped by the output plug-in.

The following are the default parameters, with sample values, for the IBM Operations Analytics - Log Analysis scala output plug-in:

```
output {
  scala {
    scala_url => "https://<la_server>:<port>/Unity/DataCollector"
    scala_user => "<LA_user>"
    scala_password => "<LA_Password>"
    scala_keystore_path => ""
    batch_size => 500000
    idle_flush_time => 5
    sequential_flush => true
    num_concurrent_writers => 20
    use_structured_api => false
    disk_cache_path => "<install-dir>/Logstash/cache-dir"
    scala_fields =>
    {
      "host1@path1,host2@path2"
      => "event_field11,event_field12,...,event_field1N"
      "host3@path3"
      => "event_field21,event_field22,...,event_field2N"
    }
    date_format_string => "yyyy-MM-dd'T'HH:mm:ssX"
    log_file => "<install-dir>/Logstash/logs/scala_logstash.log"
    log_level => "info"
  }
}
```

Where:

- **scala_url** is the REST endpoint for the Log Analysis ingestion REST API.
- **scala_user** is the Log Analysis user name.
- **scala_password** is the Log Analysis user password.
- **scala_keystore_path** is blank.
- **batch_size** is the maximum number of bytes that can be buffered for a data source before transmitting to the Log Analysis server. The default is *500000* bytes.
Note: Significantly decreasing the batch size impacts on throughput. Increasing the batch size requires more heap memory.
- **idle_flush_time** is the maximum time between successive data transmissions for a data source.
- **sequential_flush** defines whether batches for each data source are sent sequentially. It is set to *true* to send the batches sequentially.
Note: Sequential sending is required when the input contains multi-line records that are combined in an Insight Pack in the Log Analysis server.
- **num_concurrent_writers** is the number of threads that concurrently transmit batches of data to the Log Analysis server.
- **use_structured_api** determines whether data is transmitted to the Log Analysis server in the JSON format. It is set to *true* to transmit data in the JSON format.
Note: The target Log Analysis data source must be associated with a source type that uses the Log Analysis structured API.
- **disk_cache_path** is the path on the file system that temporarily buffers data. The scala output plug-in writes data to this path before transmission. The available disk space under the path must be large enough to store bursts of input data that is not immediately handled by the Log Analysis server.
- **scala_fields** is the map that specifies the names of fields that must be retrieved from the incoming logstash event and transmitted to the Log Analysis server. The keys for the map are a comma-separated list of host and path names that correspond to a Log Analysis data source.
The scala plug-in extracts the host and path fields from each event before consulting the **scala_fields** map for a host and path combination entry. If there is an entry with field names, the scala plug-in extracts the corresponding field values from the event. The values are transmitted to the Log Analysis server. If the host and path entries are not in the **scala_fields** map, the scala plug-in extracts the contents of the message field from the event and transmits it to the Log Analysis server.
- **date_format_string** is the string value that all fields are transformed to before transmission to the Log Analysis server. The scala plug-in uses the **date_format_string** parameter to convert date values to the appropriate string value.
- **log_file** is the file that is used for logging information from the scala output plug-in.
- **log_level** is the level of logging information. The supported levels are *fatal*, *error*, *warn*, *info*, and *debug*.

2. Create a custom data source. For more information, see *data source creation* in the *Administering* section.

Ensure that the **Hostname** and **File Path** match the host and path that is specified in the filter mutate section of logstash configuration file, `logstash-scala.conf`.

Ensure that the **Type** matches the type of log file that is being ingested and is defined in the filter mutate section, for example **DB2Diag**.

For example, if you specified `/tmp/myhttp.log` as an input file, then create a custom data source with path set to `/tmp/myhttp.log`.

What to do next

Start logstash.

Related tasks

[“Logstash operations” on page 231](#)

You can use the `logstash-util` script to start, stop, restart, or provide the status of Logstash.

Configuring logstash for rapid annotation and pre-indexing

logstash can be used to split log records and do basic annotation. For log types not currently supported by IBM Operations Analytics - Log Analysis, this is an alternative approach to writing AQL to annotate log files.

About this task

logstash includes a broad list of filtering, manipulation, and processing capabilities, for example, the `grok` filter can be used to parse text into structured data. You can use it to match text without the need to master regular expressions. There are approximately 120 `grok` patterns that are shipped by default, though you can add more. It also includes patterns for known log file formats, such as Apache's combined access log format.

In this scenario, you use the `grok` filter with logstash used as the splitter or annotator of the log file. The `scala` output plugin sends a single log record to the IBM Operations Analytics - Log Analysis EIF Receiver, with the annotations in a delimiter separated value (DSV) format. Then, using the DSV Toolkit, the user must create and install an insight pack that matches the DSV format so that IBM Operations Analytics - Log Analysis can index the annotations.

Procedure

1. To configure your logstash installation, specify the required values in the `<logstash_install>/Logstash/logstash-<logstash_version>/logstash-scala/logstash/config/logstash-scala.conf` file.
 - a) Define the input in the logstash configuration file.

For example:

```
input {
  file {
    type => "apache"
    path => ["/tmp/logs/myapache.log"]
  }
}
```

Note: For Windows, the logstash file plugin requires a drive letter specification for the path, for example:

```
path => ["C:/tmp/logs/myapache.log"]
```

- b) Add a filter or filters to the logstash configuration file to identify the pattern of the log file format. This also creates the annotations. To trigger the filter, the type must match the input type.

For example:

```
filter {
  if [type] == "http" {
    grok {
      match => ["message", "%{IP:client} %{WORD:method} %{URIPATHPARAM:request} %{NUMBER:bytes} %{NUMBER:duration}"]
    }
    mutate{
      replace => ["host", "<hostname>"]
      replace => ["path", "/tmp/apache.log"]
    }
  }
}
```


In this example, the fields `client`, `method`, `request`, `bytes`, and `duration` are annotated by the pattern. However, only the fields `client`, `method`, and `request` are sent to IBM Operations Analytics - Log Analysis. Thus, those are the only three annotations that can be included in the index configuration. The output module sends the event text in DSV format as:

```
"client","method","request"
```

The user can also use one of the many predefined grok log format patterns such as:

```
filter {
  if [type] == "apache" {
    grok {
      match => ["message", "%{COMBINEDAPACHELOG}"]
    }
  }
  mutate{
    replace => ["host", "<hostname>"]
    replace => ["path", "/tmp/apache.log"]
  }
}
```

- c) In the output section of the configuration file, for `{COMBINEDAPACHELOG}` specify the `scala_fields` as follows:

```
scala_fields =>
{
  "<host>@<path>" => "clientip,ident,auth,timestamp,verb,
request,httpversion,rawrequest,response,bytes,referrer,agent"
}
```

Where `<host>` and `<path>` are the values that are defined in the mutate section of the grok filter.

2. Create an IBM Operations Analytics - Log Analysis DSV-generated Insight Pack in order to index the annotated data in IBM Operations Analytics - Log Analysis.

You can use the `ApacheDSV.properties` file to create an Apache-based Insight Packs.

For more information about the `ApacheDSV.properties` file, see *ApacheDSV.properties* in the *Reference* section of the *Extending* guide.

Edit this properties file to configure information about your IBM Operations Analytics - Log Analysis server:

```
[SCALA_server]
username: unityadmin
password: unityadmin
scalaHome: $HOME/IBM/LogAnalysis
```

Use the `dsvGen.py` script that is provided with the DSV Toolkit to generate and deploy the Apache Insight Pack:

```
python dsvGen.py <path>/ApacheDSV.properties -d
```

3. Create a custom data source. For more information, see *data source creation* in the *Administering* section.

Ensure that the **Hostname** and **File Path** match the host and path that is specified in the filter mutate section of logstash configuration file, `logstash-scala.conf`.

Ensure that the **Type** matches the type of log file that is being ingested and is defined in the filter mutate section, for example **DB2Diag**.

For example, if you specified `/tmp/myhttp.log` as an input file, then create a custom data source with path set to `/tmp/myhttp.log`.

What to do next

Start logstash. For more information on starting logstash, see *logstash operations*.

Related tasks

[“Logstash operations” on page 231](#)

You can use the `logstash-util` script to start, stop, restart, or provide the status of Logstash.

Related reference

[ApacheDSV.properties](#)

Example - Annotating Combined Apache log files

Using logstash to annotate Apache log files.

Procedure

1. To configure your logstash installation, specify the required values in the `<logstash_install>/Logstash/logstash-<version>/logstash-scala/logstash/config/logstash-scala.conf file.`

- a) In the input section, specify the Apache log file to be monitored.

```
input {
  file {
    type => "apache"
    path => ["/tmp/apache.log"]
  }
}
```

- b) Add the logstash grok filter with the predefined COMBINEDAPACHELOG pattern to annotate the Apache log files.

For example:

```
filter {
  if [type] == "apache" {
    grok {
      match => ["message", "%{COMBINEDAPACHELOG}"]
    }
    mutate{
      replace => ["host", "<hostname>"]
      replace => ["path", "/tmp/apache.log"]
    }
  }
}
```

The COMBINEDAPACHELOG pattern is defined as:

```
COMBINEDAPACHELOG %{IPORHOST:clientip} %{USER:ident} %{USER:auth}
\[ %{HTTPDATE:timestamp} \] "(?:%{WORD:verb} %{NOTSPACE:request}
(?: HTTP/%{NUMBER:httpversion})?| %{DATA:rawrequest})" %{NUMBER:response}
(?:%{NUMBER:bytes}|-) %{QS:referrer} %{QS:agent}
```

For more information about Apache log files, see <http://httpd.apache.org/docs/2.4/logs.html>.

The logstash event contains annotations for `clientip`, `ident`, `auth`, `timestamp`, `verb`, `request`, `httpversion`, `rawrequest`, `response`, `bytes`, `referrer`, and `agent`.

- c) Change the `<hostname>` and `path` value to the `<hostname>` of the logstash server and path to the log file.

For example

```
mutate{
  replace => ["host", "<hostname>"]
  replace => ["path", "/tmp/apache.log"]
}
```

- d) In the output section of the configuration file, specify the IBM Operations Analytics - Log Analysis output plug-in. The `scala_fields` in the output plug-in must be defined as follows:

```
scala_fields =>
{
  "<host>@<path>" => "clientip,ident,auth,timestamp,verb,
```

```
request,httpversion,rawrequest,response,bytes,referrer,agent"
}
```

Where `<host>` and `<path>` are the values defined in the mutate section of the grok filter.

2. The `ApacheDSV.properties` file can be used with the DSV Toolkit to create an Apache Insight Pack. For more information about the `ApacheDSV.properties` file, see *ApacheDSV.properties* in the *Reference* section of the *Extending* guide. Edit this properties file to configure information about your IBM Operations Analytics - Log Analysis server:

```
[SCALA_server]
username: unityadmin
password: unityadmin
scalaHome: $HOME/IBM/LogAnalysis
```

3. Use the `dsvGen.py` script that is provided with the DSV Toolkit to generate and deploy the Apache Insight Pack:

```
python dsvGen.py <path>/ApacheDSV.properties -d
```

4. In the IBM Operations Analytics - Log Analysis Administrative Settings UI, create a data source, which has the Apache source type that is created by the DSV toolkit in step 3, in your logstash configuration file.
5. To start logstash with the configuration file, and start loading Apache log files, run the following command:

```
<logstash_install>/utilities/logstash-util.sh start
```

Related reference

[ApacheDSV.properties](#)

Adding metadata fields

To add additional data or meta data to the data that is sent from Logstash to Log Analysis, add the `metadata_fields` section to your Logstash configuration.

To add metadata fields, specify the metadata fields in your Logstash configuration. If you are using an existing logical data source, clone the source type. For more information about cloning source types, see [“Cloning source types and indexing configurations”](#) on page 314.

You can use the metadata fields in any of the following scenarios:

- To specify meta data for a physical data source. The physical data source is the source log file that are defined in the logical data source in Log Analysis. You can add metadata that helps you to identify the physical data source.
- To group multi-line log file records as part of the scalable data collection architecture. For more information, see [“Configuring the Sender cluster”](#) on page 162.
- To add further annotations at the batch level without changing the code in your Insight Pack configuration.

For example, you can add a new field such as `application_name` to the log event in Logstash. You can add additional parsing in Logstash that extracts the annotations and sends them to the metadata fields. To index the field in Log Analysis, specify the field as `metadata.Application.Name` in the Logstash configuration and clone the existing source type.

The Log Analysis plug-in tracks all the distinct values that you specify in the `meta_data` fields. Combinations of these values are maintained in distinct buckets. The plug-in creates batches for each buckets and uses the REST API to send these to Log Analysis.

Syntax

Use the following syntax to add the metadata fields to your Logstash configuration:

```
metadata_fields => {
```

```

    "<DataSource_Host1>@<DataSource_Path1>" => {
      "field_names" => "resourceID"
      "field_paths" => "resourceID"
    }
    "<DataSource_Host2>@<DataSource_Path2>" => {
      "field_names" => "resourceID"
      "field_paths" => "resourceID"
    }
  }
}

```

<DataSource_Host1> is the host name as defined in the logical data source. <DataSource_Path1> is the file path as defined in the logical data source. field_name specifies the name of the field that is added to the event log. field_path is the path in the event message.

Example

For example, you can use the following code to map the application_name field to the metadata.ApplicationName field in the Insight Pack configuration:

```

metadata_fields => {
  "WAS_Server@SystemOut" => {
    "field_names" => "ApplicationName"
    "field_paths" => "application_name"
  }
}

```

Installing Logstash on Windows based servers

If you want to use Logstash to collect log data from Windows operating systems, you need to install Logstash on the server where Windows is running.

Before you begin

- If you are using Logstash 1.5.3, install Java 1.7.5 or later on your Windows server. Ensure that this version is not IBM Java.
- If you are using Logstash 2.2.1, you must use IBM Development Kit for Java.
- Copy the client.crt certificate file from the Log Analysis server to your Windows server. To find the file, go to the <HOME>/IBM/LogAnalysis directory and enter the following command:

```
find . -name client.crt
```

If you find more than one file, you can use any one of these files.

- After you install Java, import the client.crt file into the Logstash instance's Java keystore. For example, enter the following command to import the file:

```

JAVA_HOME/bin/keytool -importcert -file <Path_to_file>/client.crt
-keystore <Java_home>/jre/lib/security/cacerts
-alias LA_ClientCertificate

```

Where <Path_to_file> is the full path to the directory where the client certificate is saved. <Java_home> is the Java home variable for the Windows server that you installed in the first prerequisite. The default password is changeit.

Procedure

1. Create a directory called logstash on the Windows server. For example, C:/logstash2.2.1/logstash/.
2. Save the Logstash bundle file in this directory.
3. To install Logstash on the Windows server, extract the Logstash file.
4. Copy the <HOME>/IBM/LogAnalysis/Logstash/Logstash-2.2.1/logstash-scala.tgz file to the logstash directory on the Windows server.
5. Extract the logstash-scala.tgz file.

Extracting this file creates two directories, `../config` and `../outputs`. You must create a nested output directory in the format `/plugins/logstash/output`. For example, `C:/logstash2.2.1/logstash/plugins/logstash/output`.

6. If you want to use Logstash with the Windows OS Events Insight Pack, you must copy the `logstash-scala.conf` file from the Insight Pack to the `/plugins/logstash/config` directory on the Windows server.

For more information, see [“Integrating the Windows OS Events Insight Pack with logstash” on page 302](#).

What to do next

After you complete the installation, you can start using Logstash.

For example, assuming that you extracted the Log Analysis plugin in the `C:\logstash-plugins\logstash` folder, enter the following command to start Logstash:

```
C:\logstash-2.2.1>bin\logstash agent --verbose -f
C:\logstash-plugins\logstash\scala.conf --pluginpath
C:\logstash-plugins\logstash\plugins -l console.log
```

The log files in this example are called `console.log`. You can rename the file as you need to.

To stop Logstash, enter a CTRL+C command in the same console. You cannot use the `logstash-util.sh` script to operate Logstash on Windows servers.

Logstash configuration file reference

The `logstash-scala.conf` configuration file controls how Logstash annotates and stores log file records.

You must specify the parameters that are described in the following table.

Table 50. Required values for the <code>logstash-scala.conf</code> file	
Parameter	Description
LA_SERVER_IP	Specify the IP address or host name of the Log Analysis server.
SCALA_KEYSTORE_PATH	Specify the full path to the directory where the keystore file is saved on the Log Analysis server.
PATH_TO_DIR	Specify the path to the directory where the Windows OS Events Insight Pack stores the cache. For example, <code>C:\ProjectWork\LogAnalytics\Scala-v1.3\v1.3.2\Logstash-1-5-Integration\logstash-2.2.1\configs\WindowsOSInsightPackTests\cache</code> .
PATH_TO_FILE	Specify the path to the directory where you want to store the log files.

Example

```
input {
  eventlog {
    type => 'Win32-EventLog'
  }
}

filter{
  if [type] == "Win32-EventLog"{
    grok {
      match => ["TimeGenerated", "%{WORD:TIMEGEN_DATETIME}
.%{WORD:TIMEGEN_MICROSEC}%{ISO8601_TIMEZONE:TIMEGEN_TZONE}"]
      add_tag => ["eventlog_grokked"]
    }
  }
}
```

```

}
if "eventlog_grokked" in [tags]{
  grok{
    match => ["TIMEGEN_DATETIME","^%{YEAR:YYYY}%{MONTHNUM:MM}%{MONTHDAY:dd}%{HOUR:hh}%{MINUTE:mm}%{SECOND:ss}$"]
    add_tag => [ "eventlog_timestamp_grokked" ]
  }
}
mutate {
  # sample timestamp format that works with LA:
  # 2015-08-26 18:54:26 +0000
  # yyyy-MM-dd HH:mm:ss Z
  # Timezone info is set to UTC always

  add_field => [ "timestamp", "%{YYYY}-%{MM}-%{dd} %{hh}:%{mm}:%{ss} +0000" ]

  # Update the path - this is the data source file path on the SCALA server
  update => [ "path", "WindowsOSEventsLogStash" ]

  #####
  # These are the fields that are sent to SCALA to match the
  # Source type WindowsOSEventsLogStash The following fields are
  # ingested by SCALA
  # Hostname
  # Level
  # EventRecordNumber
  # EventSource
  # EventID
  # timestamp
  # EventLog
  # User
  # Description
  # Category
  #####

  add_field => ["scalaFields", "timestamp"]      #
  add_field => ["scalaFields", "Category"]      #
  add_field => ["scalaFields", "message"]      # Description
  add_field => ["scalaFields", "EventIdentifier"] # EventID
  add_field => ["scalaFields", "Type"]          # Level - Warning,
  Information etc
  add_field => ["scalaFields", "Logfile"]        # EventLog
  e.g. Application, System etc
  add_field => ["scalaFields", "SourceName"]     # EventSource
  add_field => ["scalaFields", "ComputerName"]  # Hostname
  add_field => ["scalaFields", "RecordNumber"]  # EventRecordNumber
  add_field => ["scalaFields", "User"]          #

  } # end mutate
  } # end if type = Win32-EventLog
} # end filter

output {
  # Uncomment out below file output if users wish to view
  # event data in a debug file. Specify the path for the file.

  #file {
  # codec => rubydebug
  # #path => "C:\yogesh\logstash-plugins\ruby-debug.log"
  # path => "PATH_TO_FILE\ruby-debug.log"
  #}

  scala {
    scala_url => "https://LA_SERVER_IP:9987/Unity/DataCollector"
    scala_user => "unityadmin"
    scala_password => "unityadmin"
    # You can either use encrypted password or use a plain text one.
    # If encrypted password is used, provide the keystore path below.
    # If plaintext password is used, just comment the below
    line for keystore path
    scala_keystore_path => "SCALA_KEYSTORE_PATH"
    batch_size => 500000
    idle_flush_time => 5
    sequential_flush => true
    num_concurrent_writers => 20
    use_structured_api => false
    # Ensure you create a cache dir. And configure the path below -
    disk_cache_path => "PATH_TO_DIR\cache"
    #disk_cache_path => "C:\ProjectWork\LogAnalytics\Scala-v1.3\v1.3.2\
    Logstash-1-5-Integration\logstash-2.2.1\configs\
    WindowsOSInsightPackTests\cache"
    scala_fields =>

```

```
{
#"host1@path1,host2@path2"
# => "timestamp,Category,...,event_field1N"
"co912212163@WindowsOSEventsLogStash"
=> "timestamp,Category,message,EventIdentifier,Type,
Logfile,SourceName,ComputerName,RecordNumber,User"
}
date_format_string => "yyyy-MM-dd'T'HH:mm:ssX"
log_file => "PATH_TO_FILE\scala-debug.log"
log_level => "debug"
```

Logstash operations

You can use the `logstash-util` script to start, stop, restart, or provide the status of Logstash.

About this task

You can use the `logstash-util` script for Logstash process lifecycle management.

Procedure

1. To start, stop, restart, or provide the status of Logstash, run the following command:

```
<install-dir>/utilities/logstash-util.sh start| stop| restart| status
```

where `<install-dir>` is the name of the Logstash installation location.

2. To confirm that Logstash is running, run the `logstash-util` script and use the status option. The status option also displays the Logstash process identifier.

logstash best practices

Best practices for logstash based on information from their user community.

For performance reasons it is recommend that logstash be installed on a different server than IBM Operations Analytics - Log Analysis. logstash is processor, memory, and disk intensive if the annotation and indexing functions are utilized.

Users who have memory constraints do not use logstash as a forwarding agent. They do not install logstash on the end client servers. They use other applications such as rsyslog to forward logs to a central server with logstash. See <https://support.shotgunsoftware.com/entries/23163863-Installing-logstash-Central-Server> for an example configuration.

Users with logstash at the end client who are concerned about performance have used applications such as Redis to forward logs to a central server with logstash. See the following for configuration of Redis <http://www.linux-magazine.com/Online/Features/Consolidating-Logs-with-logstash> .

To fine tune logstash, especially the startup time, users can tweak Java's minimum and maximum heap size with the `-Xms` and `-Xmx` flags. The `-Xms` parameter is the initial Java memory heap size when the JVM is started, and the `-Xmx` parameter is the maximum heap size.

References

Links for more information on the logstash application.

logstash website:

<http://logstash.net>

Getting Started with logstash Guide:

<https://www.elastic.co/guide/en/logstash/current/getting-started-with-logstash.html>

logstash Download:

<http://logstash.net> (Click download button)

The logstash Book:

<http://www.logstashbook.com/>

IBM Operations Analytics - Log Analysis wiki:

<http://www.ibm.com/developerworks/servicemanagement/ioa/log/downloads.html>

IBM Operations Analytics - Log Analysis wiki: Logstash Toolkit Resources:

<https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Log%20Analytics%20Beta/page/Logstash%20Toolkit%20Resources>

Known issues

Known issues when using logstash with IBM Operations Analytics - Log Analysis.

There are a number of known issues and their workarounds described in this section. To get the latest information on any issues or workarounds, please consult the IBM Operations Analytics - Log Analysis wiki: <https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Log%20Analytics%20Beta/page/Welcome>

Could not load FFI Provider

Starting logstash fails with the Ruby exception "Could not load FFI Provider".

Symptoms

The Ruby exception "Could not load FFI Provider".

Causes

The most common cause of this error is that /tmp is mounted with the **noexec** flag.

Resolving the problem

You can resolve this either by:

- Making /tmp mounted without the **noexec** flag
- Edit the startlogstash-scala script and amend the start command as follows:

```
LSCMD="$MYJAVA -jar -Djava.io.tmpdir=</some/tmp/dir> $LSJAR agent  
--pluginpath $PLUGPATH -f $CONF"
```

Where </some/tmp/dir> is a temporary directory.

Duplication of log records on the SCALA server

On occasion, when the logstash agent is re-started, and the log file being monitored is updated (for example, via a streaming log), logstash will ingest the entire file again rather than restarting from where it stopped monitoring.

Symptoms

The problem results in a duplication of log records on the SCALA server.

Causes

Several problems have been reported on the logstash forum (<https://logstash.jira.com/secure/Dashboard.jspa>) that its sincedb pointer (which tracks the last monitored position in the log file) sometimes is not updated correctly. In addition, using control-C to terminate the logstash agent does not always kill logstash. The result is a "phantom" logstash agent that is still monitoring log files. This can also result in duplicate log records.

Resolving the problem

1. A workaround to avoid duplicate log records after restarting logstash is to set the **sincedb_path** parameter in the file plugin to /dev/null, thereby telling logstash to ignore tracking the last-monitored file position, and always start monitoring from the end of the file. However, this will result in logstash ignoring any updates to the log file while the logstash agent is down. For example, in logstash-scala.conf, update:

```
input {  
  file {  
    type => "apache"  
    path => ["/tmp/logs/myapache.log"]
```



```
    }  
    sincedb_path => "/dev/null"  
  }
```

Before re-starting logstash after making these configuration changes, you may also want to clean up any sincedb databases that were already created. By default, the sincedb database is stored in the directory \$HOME, and have filenames starting with ".sincedb_".

2. When terminating the logstash agent using control-C, verify that the logstash java process was actually terminated. You can use the following command to see if logstash is still running:

```
ps -ef | grep logstash
```

Logs do not appear in the Search UI

Log records are ingested by logstash, but do not appear in the IBM Operations Analytics - Log Analysis Search UI.

Symptoms

Log records are ingested by logstash, but do not appear in the IBM Operations Analytics - Log Analysis Search UI.

Causes

Log records ingested by logstash are forwarded to the IBM Operations Analytics - Log Analysis server for splitting and annotating, and indexing. If the IBM Operations Analytics - Log Analysis server goes down during this process, it is possible to lose some log records.

Configuring the EIF Receiver

Log Analysis uses the Tivoli Event Integration Facility (EIF) Receiver to load data.

About this task

The following steps are the same for remote and local installations of the EIF unless stated otherwise.

In most cases, you modify these settings to improve the performance of data loading or to adapt the default settings to match your environment.

You can change the buffer size and timeout values, the user accounts, the number of events and the clean up interval.

Configuring receiver buffer size and timeout

When collecting data using the IBM Tivoli Monitoring Log File Agent (LFA) and Tivoli Event Integration Facility (EIF) Adapter flow, you might need to change the rate at which events are flushed to the generic receiver for indexing. Incoming events are buffered at the EIF receiver side.

About this task

To improve overall IBM Operations Analytics - Log Analysis performance, you can configure the buffer size and timeout period to match the rate of incoming events. When the event rate increases, increase the buffer size and decrease the timeout period. When the event rate decreases, decrease the buffer size and keep the timeout interval at the default value or increase it, depending on the event rate.

The following steps are the same for remote and local installations of the EIF unless stated otherwise.

Procedure

To change the buffer size and timeout parameters:

1. Stop IBM Operations Analytics - Log Analysis:

- If you use a local installation of the EIF, use the following command to stop IBM Operations Analytics - Log Analysis:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
```

- If you use a remote installation of the EIF, use the `eifutil.sh -stop` command to stop the instances. For more information, see [“eifutil.sh command” on page 107](#).

2. Open the configuration file for editing:

- If you use a local installation of the EIF, open the `unity.conf` file in the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/` directory.
- If you use a remote installation of the EIF, the `unity.conf` file is in the `<remote_deployment_location>/LogAnalysis/DataForwarders/EIFReceivers/<eif_inst_#>/config/unity.conf` directory. Where `<remote_deployment_location>` is the directory on the remote machine where you deployed the EIF instance. `<eif_inst_#>` is the folder used for the specific remote EIF instance.

3. Change the Timeout and Buffer Size parameters to suit your operating environment:

```
#Timeout in Seconds
logsource.buffer.wait.timeout=10
#Buffer Size in Bytes
logsource.max.buffer.size=250000
```

4. Save your changes.

5. Start IBM Operations Analytics - Log Analysis:

- If you use a local installation of the EIF, use the following command to start IBM Operations Analytics - Log Analysis:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -start
```

- If you use a remote installation of the EIF, use the `eifutil.sh -start` command to start the instances. For more information, see [“eifutil.sh command” on page 107](#).

Results

With higher buffer sizes, notice that it takes a longer time to fill the buffer with events and for batches to be posted to the receiver.

Configuring the EIF receiver user account

The Tivoli Event Integration Facility (EIF) receiver uses the default `unityuser` user account to access the generic receiver. You can change the user account or the default user password in the `unity.conf` configuration file.

About this task

The following steps are the same for remote and local installations of the EIF unless stated otherwise.

Procedure

To change the default EIF user or password:

1. Stop IBM Operations Analytics - Log Analysis:

- If you use a local installation of the EIF, use the following command to stop IBM Operations Analytics - Log Analysis:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
```

- If you use a remote installation of the EIF, use the `eifutil.sh -stop` command to stop the instances. For more information, see [“eifutil.sh command” on page 107](#).

2. Open the configuration file for editing:

- If you use a local installation of the EIF, open the `unity.conf` file in the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/` directory.
 - If you use a remote installation of the EIF, the `unity.conf` file is in the `<remote_deployment_location>/LogAnalysis/DataForwarders/EIFReceivers/<eif_inst_#>/config/unity.conf` directory. Where `<remote_deployment_location>` is the directory on the remote machine where you deployed the EIF instance. `<eif_inst_#>` is the folder that is used for the specific remote EIF instance.
3. Change the following `userid` and `password` parameters to suit your operating environment:

```
unity.data.collector.userid=unityuser
unity.data.collector.password=password
```

To encrypt the password, use the `unity_securityUtility.sh` command. For more information, see [“Changing the default password for the Data Collector and EIF Receiver” on page 237](#).

4. Save your changes.

5. Restart IBM Operations Analytics - Log Analysis:

- If you use a local installation of the EIF, use the following command to restart IBM Operations Analytics - Log Analysis:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
```

- If you use a remote installation of the EIF, use the `eifutil.sh -restart` command to restart the instances. For more information, see [“eifutil.sh command” on page 107](#).

Results

The EIF receiver uses the new credentials to access the generic receiver.

Configuring the number of events in the EIF Receiver

You can configure the number of events that the EIF Receiver stores for each internal queue. If you intend to ingest a large quantity of data and at a high rate, configure these values to larger values. However, increasing this value also increases the memory requirements for EIF Receiver.

About this task

Ensure that you have sufficient memory to support the number of events in the queue.

The following steps are the same for remote and local installations of the EIF unless stated otherwise.

Procedure

To change this setting:

1. Stop IBM Operations Analytics - Log Analysis:

- If you use a local installation of the EIF, use the following command to stop IBM Operations Analytics - Log Analysis:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
```

- If you use a remote installation of the EIF, use the `eifutil.sh -stop` command to stop the instances. For more information, see [“eifutil.sh command” on page 107](#).

2. Open the configuration file for editing:

- If you use a local installation of the EIF, open the `unity.conf` file in the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/` directory.
- If you use a remote installation of the EIF, the `unity.conf` file is in the `<remote_deployment_location>/LogAnalysis/DataForwarders/EIFReceivers/<eif_inst_#>/config/unity.conf` directory. Where `<remote_deployment_location>` is the

directory on the remote machine where you deployed the EIF instance. `<eif_inst_#>` is the folder used for the specific remote EIF instance.

3. Locate these lines and change the value to reflect your requirements:

```
unity.data.collector.eif.consumer.num.events=1000000
unity.data.collector.event.manager.num.events=20000
```

The following settings are applicable per data source:

```
unity.data.collector.event.service.num.events=20000
unity.data.collector.event.poster.num.events=500
```

4. Save your changes.

5. Start IBM Operations Analytics - Log Analysis:

- If you use a local installation of the EIF, use the following command to start IBM Operations Analytics - Log Analysis:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -start
```

- If you use a remote installation of the EIF, use the `eifutil.sh -start` command to stop the instances. For more information, see [“eifutil.sh command” on page 107](#).

Configuring the EIF Receiver memory clean up interval

IBM Operations Analytics - Log Analysis ensures that the memory used for data collection with the Log File Agent using a property in the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/unity.conf` file. The EIF Receiver uses this value to manage the memory usage. The configuration cycle is set to a value in minutes with a default value of 2 minutes.

About this task

The following steps are the same for remote and local installations of the EIF unless stated otherwise.

Procedure

To configure this property:

1. Stop IBM Operations Analytics - Log Analysis:

- If you use a local installation of the EIF, use the following command to stop IBM Operations Analytics - Log Analysis:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
```

- If you use a remote installation of the EIF, use the `eifutil.sh -stop` command to stop the instances. For more information, see [“eifutil.sh command” on page 107](#).

2. Open the configuration file for editing:

- If you use a local installation of the EIF, open the `unity.conf` file in the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/` directory.
- If you use a remote installation of the EIF, the `unity.conf` file is in the `<remote_deployment_location>/LogAnalysis/DataForwarders/EIFReceivers/<eif_inst_#>/config/unity.conf` directory. Where `<remote_deployment_location>` is the directory on the remote machine where you deployed the EIF instance. `<eif_inst_#>` is the folder that is used for the specific remote EIF instance.

3. Change the parameters to suit your operating environment:

```
#gc interval is in minutes
unity.data.collector.gc.interval=2
```

4. Save your changes.

5. Start IBM Operations Analytics - Log Analysis:

- If you use a local installation of the EIF, use the following command to start IBM Operations Analytics - Log Analysis:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -start
```

- If you use a remote installation of the EIF, use the `eifutil.sh -start` command to start the instances. For more information, see [“eifutil.sh command” on page 107](#).

Changing the default password for the Data Collector and EIF Receiver

If you want, you can change the default password that the Data Collector and EIF Receiver use to connect to IBM Operations Analytics. This is optional.

Changing the default EIF Receiver or Data Collector password

You can change the default password that the Data Collector and EIF Receiver use to connect to IBM Operations Analytics - Log Analysis.

About this task

After you install IBM Operations Analytics - Log Analysis, the EIF Receiver and the Data Collector are configured to use the default user name and password to connect to IBM Operations Analytics - Log Analysis. The encrypted passwords are defined in the following files:

- Data Collector client is named `<HOME>/IBM/LogAnalysis/utilities/datacollector-client/javaDatacollector.properties`.
- EIF Receiver is named `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/unity.conf`.

IBM Operations Analytics - Log Analysis uses the Advanced Encryption Standard (AES) to encrypt and decrypt passwords for your installation, in the following format:

```
password={aes}<Unique_string_of_alphanumeric_characters>
```

For example, the `javaDatacollector.properties` file uses the `unityuser` user ID to access the Data Collector server. In this example, IBM Operations Analytics - Log Analysis uses the Advanced Encryption Standard (AES) to generate the following password:

```
{aes}7DB629EC03AABEC6C4484F160FB23EE8
```

The encrypted password is replicated to the configuration files for the Data Collector and the EIF Receiver.

Procedure

1. To change the default password, use the `unity_securityUtility.sh` command.
For more information about this command, see [“unity_securityUtility.sh command” on page 109](#).
2. Update the configuration files for the Data Collector or the EIF Receiver.
3. If you want to change the password on remote instances of the EIF Receiver, complete the previous steps and copy the `unity.conf` file from the `<HOME>/IBM/LogAnalysis/UnityEIFReceiver/config/` directory on the local machine to the `<remote_deployment_location>/LogAnalysis/DataForwarders/EIFReceivers/<eif_inst_#>/config/unity.conf` directory on the remote machine. Where `<remote_deployment_location>` is the directory on the remote machine where you deployed the EIF instance. `<eif_inst_#>` is the folder that is used for the specific remote EIF instance.

Example

For example, you want to change the default password for the default user that the Data Collector and EIF Receiver use to connect to IBM Operations Analytics - Log Analysis to `myNewPassword`. Complete the following steps:

1. Go to the IBM/LogAnalysis/utilities directory.
2. Run the `unity_securityUtility.sh` command as follows:

```
[utilities]$ ./unity_securityUtility.sh encode myNewPassword
Using keystore file unity.ks
<HOME>/IBM/LogAnalysis/utilities/../wlp/usr/servers/Unity/
keystore/unity.ks
{aes}E6FF5235A9787013DD2725D302F7D08
```

3. Copy the AES encrypted password to the relevant configuration files, for example copy it to the Data Collector file. You must copy the complete, encrypted string from the command output, including the `{aes}` prefix. For example:

```
{aes}E6FF5235A9787013DD2725D302F7D088
```

unity_securityUtility.sh command

You can use the `unity_securityUtility.sh` command to change the password that the Data Collector and EIF Receiver use to connect to IBM Operations Analytics - Log Analysis.

Syntax

The `unity_securityUtility.sh` command is in the `<HOME>/IBM/LogAnalysis/utilities` directory and it has the following syntax:

```
unity_securityUtility.sh encode [textToEncode] [unity.ks]
```

Parameters

The `unity_securityUtility.sh` command has the following parameters:

encode

The encode action returns an AES encrypted version of the text that you enter as the text to encrypt.

[*textToEncode*]

Use the [*textToEncode*] parameter to enter the password that you want to encrypt. If you do not specify a password for this parameter, IBM Operations Analytics - Log Analysis prompts you for one.

[unity.ks]

The `unity.ks` file is the default keystore that is generated automatically during installation. It controls how the password is encrypted and decrypted.

The `unity.ks` file is used to encrypt and decrypt passwords for the following features:

- Java data collector client in the `<HOME>/IBM/LogAnalysis/utilities/datacollector-client/javaDatacollector.properties` file.
- EIF Receiver in the `<HOME>/IBM/LogAnalysis/utilities/UnityEIFReceiver/config/unity.conf` file.

For an example of how to use this command, see [“Changing the default EIF Receiver or Data Collector password”](#) on page 237.

Ingestion of non-English-language log files

To facilitate the ingestion of non-English-language log files, you must change the IBM Tivoli Monitoring Log File Agent locale to match the value that is used by the source application to create the log files.

You can use the IBM Tivoli Monitoring Log File Agent or the Data Collector to ingest the files.

The source application where the ingested log files are created must use UTF-8 encoding.

You must change the IBM Tivoli Monitoring Log File Agent locale to match the locale that is used in the source application. The parameter is called `LFA_LOCALE` and it is in the `locale.properties` file in the `/utilities/config` folder.

Ingesting non-English-language log files with the internal IBM Tivoli Monitoring Log File Agent

Before you can use the IBM Tivoli Monitoring Log File Agent that is installed with IBM Operations Analytics - Log Analysis to ingest non-English-language log files, you must configure the locale.

About this task

The new locale applies to all the logs that are ingested by the internal IBM Tivoli Monitoring Log File Agent, regardless of whether these logs are monitored locally or remotely. If you are monitoring logs that are remote to the agent, ensure that the locale that you specify in these steps matches the locale that is used in the source application. If you are monitoring files that are local to the agent, ensure that the locale that you specify in these steps matches the one that is used by the local agent.

Procedure

1. To stop the IBM Operations Analytics - Log Analysis server, use the `unity.sh` command:

```
./unity.sh -stop
```

2. To set the locale, go to the `<HOME>/IBM/LogAnalysis/utilities/config` folder and edit the `locale.properties` file. For example, to enable the ingestion of Chinese, set the locale to `zh_CN.UTF-8`:

```
LFA_LOCALE=zh_CN.UTF-8
```

3. To start the IBM Operations Analytics - Log Analysis server, use the `unity.sh` command:

```
./unity.sh -start
```

Results

The new locale is only applied to the IBM Tivoli Monitoring Log File Agent. It is not applied to the other services in IBM Operations Analytics - Log Analysis.

After you change the IBM Tivoli Monitoring Log File Agent locale to a value other than the default value, you must ensure that the locale for the console where you run the `unity.sh` script is the same as the changed locale.

If you used the remote installer utility to install a remote instance of the IBM Tivoli Monitoring Log File Agent, you must configure this instance separately. See [“Streaming non-English-language log files from a remote, internal IBM Tivoli Monitoring Log File Agent” on page 239](#).

Streaming non-English-language log files from a remote, internal IBM Tivoli Monitoring Log File Agent

Before you can use a remote instance of the internal IBM Tivoli Monitoring Log File Agent that is installed with IBM Operations Analytics - Log Analysis to ingest non-English-language log files, you must configure the locale.

About this task



Warning: This feature requires that you install interim fix 4 for IBM Operations Analytics - Log Analysis 1.2.

In this case, it is assumed that you used the `install.sh` command that is in the `<HOME>/eif_remote_install_tool/` to create a remote instance of the internal IBM Tivoli Monitoring Log File

Agent that is installed with IBM Operations Analytics - Log Analysis. Where <HOME> is the directory in which IBM SmartCloud Analytics - Log Analysis is installed.

The new locale applies to all the logs that are streamed by the internal IBM Tivoli Monitoring Log File Agent, regardless of whether these logs are monitored locally or remotely. If you are monitoring logs that are remote to the agent, ensure that the locale that you specify in these steps matches the locale that is used in the source application. If you are monitoring files that are local to the agent, ensure that the locale that you specify in these steps matches the one that is used by the local agent.

Procedure

1. To stop the remote instance, use the `lfautil.sh` command that is in the `<USER_HOME_REMOTE>/utilities/` directory on the remote server. `<USER_HOME_REMOTE>` is the directory on the remote server where the IBM Tivoli Monitoring Log File Agent is deployed.

```
./lfautil.sh -stop
```

2. To set the locale, go to the `<USER_HOME_REMOTE>/LogAnalysis/utilities` directory and edit the `LFA_LOCALE` parameter in the `locale.properties` file. For example, to enable the ingestion of Chinese, set the locale to `zh_CN.UTF-8`:

```
LFA_LOCALE=zh_CN.UTF-8
```

3. To start the remote instance, use the `lfautil.sh` command that is in the `<USER_HOME_REMOTE>IBM/LogAnalysis/utilities/` directory on the remote server.

```
./lfautil.sh -start
```

Results

The new locale is only applied to the IBM Tivoli Monitoring Log File Agent. It is not applied to the other services in IBM Operations Analytics - Log Analysis.

After you change the IBM Tivoli Monitoring Log File Agent locale to a value other than the default value, you must ensure that the locale for the console where you run the `lfautil.sh` script is the same as the changed locale.

Ingesting non-English-language log files from an external IBM Tivoli Monitoring Log File Agent

Before you can use an external IBM Tivoli Monitoring Log File Agent that is not installed with IBM Operations Analytics - Log Analysis to ingest non-English-language log files, you must configure the locale.

Before you begin

Ensure that UTF-8 encoding is enabled on the server where the IBM Tivoli Monitoring Log File Agent is installed.

Procedure

1. To stop the IBM Tivoli Monitoring Log File Agent instance, use the `itmcmd` command.
2. To set the locale, enter the following command:

```
LC_ALL=<Locale>  
export LC_ALL
```

For example, to set the locale for Chinese, enter the following command:

```
LC_ALL=zh_CN.UTF-8  
export LC_ALL
```

3. To list the applied changes, enter the `locale` command.

The command outputs a list of the changes. For example:

```
LANG=en_US.UTF-8
LC_CTYPE="zh_CN.UTF-8"
LC_NUMERIC="zh_CN.UTF-8"
LC_TIME="zh_CN.UTF-8"
LC_COLLATE="zh_CN.UTF-8"
LC_MONETARY="zh_CN.UTF-8"
LC_MESSAGES="zh_CN.UTF-8"
LC_PAPER="zh_CN.UTF-8"
LC_NAME="zh_CN.UTF-8"
LC_ADDRESS="zh_CN.UTF-8"
LC_TELEPHONE="zh_CN.UTF-8"
LC_MEASUREMENT="zh_CN.UTF-8"
LC_IDENTIFICATION="zh_CN.UTF-8"
LC_ALL=zh_CN.UTF-8
```

Note: You do not need to change the LANG parameter.

4. To start the IBM Tivoli Monitoring Log File Agent instance, use the `itmcmd` command.

Ingesting non-English-language log files with the Data Collector

Before you can use the Data Collector to ingest non-English-language log files, you must configure the locale.

Before you begin

Ensure that UTF-8 encoding is enabled on the server where the IBM Tivoli Monitoring Log File Agent is installed.

Procedure

1. To set the locale, enter the following command:

```
LC_ALL=<Locale>
export LC_ALL
```

Where *<Locale>* is the locale that you want to set. For example, to set the locale for Chinese, enter the following command:

```
LC_ALL=zh_CN.UTF-8
export LC_ALL
```

2. To list the applied locale changes, enter the `locale` command.

The command outputs a list of the changes. For example:

```
LANG=en_US.UTF-8
LC_CTYPE="zh_CN.UTF-8"
LC_NUMERIC="zh_CN.UTF-8"
LC_TIME="zh_CN.UTF-8"
LC_COLLATE="zh_CN.UTF-8"
LC_MONETARY="zh_CN.UTF-8"
LC_MESSAGES="zh_CN.UTF-8"
LC_PAPER="zh_CN.UTF-8"
LC_NAME="zh_CN.UTF-8"
LC_ADDRESS="zh_CN.UTF-8"
LC_TELEPHONE="zh_CN.UTF-8"
LC_MEASUREMENT="zh_CN.UTF-8"
LC_IDENTIFICATION="zh_CN.UTF-8"
LC_ALL=zh_CN.UTF-8
```

Note: You do not need to change the LANG parameter.

Results

After you set the locale, you can use the Data Collector to ingest non-English-language log files. Ensure that you run the Data Collector from the same terminal that you set the locale on.

Chapter 7. Managing Insight Packs

Use Insight Packs to help you to annotate and index-specific log file records.

Before you can use any of the Insight Packs, you must install and configure the Insight Packs. You can:

- Create your own custom Insight Packs.
- Use the Insight Packs that are available out of the box.
- Download more Insight Packs.

The Insight Pack defines:

- The type of data that is to be consumed.
- How data is annotated. The data is annotated to highlight relevant information.
- How the annotated data is indexed. The indexing process allows you to manipulate search results for better problem determination and diagnostics.
- How to render the data in a chart.

Out of the box Insight Packs

The following Insight Packs are now installed with the product:

WASInsightPack

The WebSphere Application Server Insight Pack includes support for ingesting and performing metadata searches against WebSphere Application Server V7 and V8 log files. Updates to WAS index configuration will improve indexing performance. The field `logsourceHostname` has been changed to `datasourceHostname`.

WASAppInsightPack

The Websphere Application Server (WAS) Applications Insight Pack provides troubleshooting dashboards for WebSphere Application Server Logs. A new authentication mechanism eliminates the need to specify `userid` and `password` in the application script. The field `logsourceHostname` has been changed to `datasourceHostname`.

DB2InsightPack

The DB2 Insight Pack includes support for ingesting and performing metadata searches against DB2 version 9.7 and 10.1 `db2diag.log` files. The field `logsourceHostname` has been changed to `datasourceHostname`.

DB2AppInsightPack

The DB2 Applications Insight Pack provides troubleshooting dashboards for DB2 Logs. A new authentication mechanism eliminates the need to specify `userid` and `password` in the application script. The field `logsourceHostname` has been changed to `datasourceHostname`.

Syslog Insight Pack

The Syslog Insight Pack includes support for ingesting and performing metadata searches against syslog data logging. The field `logsourceHostname` has been changed to `datasourceHostname`.

WebAccessLogInsightPack

The Web Access Logs Insight Pack provides the capability to ingest and perform metadata searches against Web Access Logs such as Apache IHS, JBoss, Apache Tomcat. The pack now includes a Web Health Check Dashboard example that provides summaries of key metrics.

WindowsOSEventsInsightPack

You can use the Windows OS Event Insight pack and the IBM Tivoli Monitoring Log File Agent to load and search Windows OS events. New support for data collection using Logstash provides an alternative to the IBM Tivoli Monitoring Log File Agent.

JavacoreInsightPack

The Java Core Insight Pack provides the capability to ingest and search metadata that originates in Java Core files in IBM Operations Analytics - Log Analysis. The field `logsourceHostname` has been changed to `datasourceHostname`.

GAInsightPack

The Generic Annotation Insight Pack is not specific to any particular log data type. It can be used to analyze log files for which a log-specific Insight Pack is not available

Additional Insight Packs

You can download more Insight Packs to further extend Log Analysis.

To view the available Insight Packs, see [Available Insight Packs](#).

For more information about Insight Packs, see [Log Analysis Insight Packs](#).

To view all of the available Log Analysis resources, including Insight Packs, see [Solutions](#)

Related concepts

[“Custom Insight Packs” on page 305](#)

You can use custom Insight Packs to implement customized indexing and annotating.

Supported software

For more information about software that is supported for a specific Insight Pack, see the documentation that is included with the Insight Pack.

Downloading an Insight Pack

You can download extra Insight Packs and upgrade existing Insight Packs.

To view the available Insight Packs, go to <https://developer.ibm.com/itom/resources/category/log-analysis/>.

If you did not use the Insight Pack previously, you can install it. If you installed an older version of the Insight Pack, you can upgrade it.

Related tasks

[“Upgrading an Insight Pack” on page 245](#)

You can upgrade an Insight Pack that you have previously installed. This topic outlines how to upgrade an existing Insight Pack.

[“Installing Insight Packs” on page 244](#)

Before you can use an Insight Pack, you need to install it.

Installing Insight Packs

Before you can use an Insight Pack, you need to install it.

About this task

This task is a generic summary of the process for installing Insight Packs. The steps can vary for specific Insight Packs. For more information, see the documentation for the Insight Pack.

Procedure

1. Upload the Insight Pack archive file, `InsightPack_<version>.zip`, to the system where IBM Operations Analytics - Log Analysis is installed.
2. Install the Insight Pack with the `pkg_mgmt.sh` command:

```
<HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -install  
<path>/InsightPack_<version>.zip
```

Where *<path>* is the path where you saved the Insight Pack.

Upgrading an Insight Pack

You can upgrade an Insight Pack that you have previously installed. This topic outlines how to upgrade an existing Insight Pack.

About this task

If the Insight Pack that you want to upgrade is not installed, you can choose to complete a full installation of the Insight Pack. In addition to upgrading existing artifacts and installing any artifacts added to the Insight Pack, this command removes unused artifacts that have been excluded from the upgraded Insight Pack.

Upgrade an Insight Pack by completing these steps:

Procedure

1. Download the Insight Pack archive and copy it to the *<HOME>/IBM/LogAnalysis/unity_content* directory on your IBM Operations Analytics - Log Analysis system.
2. Execute the command:

```
<HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -upgrade insight_pack.zip  
-U username -P password -f
```

where *insight_pack* is the path to the Insight Pack that you want to upgrade. These additional parameters are also defined:

-U

(Optional) The username for a user with administrative access rights. This parameter is not necessary if you have not changed the default *unityadmin* password.

-P

(Optional) The password for the username that you have specified. This parameter is not necessary if you have not changed the default *unityadmin* password.

-f

(Optional) This parameter can also be used to install the Insight Pack, if it is not already installed.

3. (Optional) If the Insight Pack is not installed and you have not specified the *-f* parameter, a message is displayed indicating that the Insight Pack is not installed. If you want to proceed, enter Y.

Cloning source types for Insight Packs

To help you to create your own source types and indexing configurations, you can clone them from an existing Insight Pack.

For more information about how to do so, see the *Cloning source types and index configurations* topic in the *Installation and Administration* guide.

Out of the box Insight Packs

Some Insight Packs are available as part of the product. You can install and use these Insight Packs to help you to get started.

The following Insight Packs are now installed with the product:

WASInsightPack

The WebSphere Application Server Insight Pack includes support for ingesting and performing metadata searches against WebSphere Application Server V7 and V8 log files. Updates to WAS index configuration will improve indexing performance. The field `logsourceHostname` has been changed to `datasourceHostname`.

WASAppInsightPack

The WebSphere Application Server (WAS) Applications Insight Pack provides troubleshooting dashboards for WebSphere Application Server Logs. A new authentication mechanism eliminates the need to specify `userid` and `password` in the application script. The field `logsourceHostname` has been changed to `datasourceHostname`.

DB2InsightPack

The DB2 Insight Pack includes support for ingesting and performing metadata searches against DB2 version 9.7 and 10.1 `db2diag.log` files. The field `logsourceHostname` has been changed to `datasourceHostname`.

DB2AppInsightPack

The DB2 Applications Insight Pack provides troubleshooting dashboards for DB2 Logs. A new authentication mechanism eliminates the need to specify `userid` and `password` in the application script. The field `logsourceHostname` has been changed to `datasourceHostname`.

Syslog Insight Pack

The Syslog Insight Pack includes support for ingesting and performing metadata searches against syslog data logging. The field `logsourceHostname` has been changed to `datasourceHostname`.

WebAccessLogInsightPack

The Web Access Logs Insight Pack provides the capability to ingest and perform metadata searches against Web Access Logs such as Apache IHS, JBoss, Apache Tomcat. The pack now includes a Web Health Check Dashboard example that provides summaries of key metrics.

WindowsOSEventsInsightPack

You can use the Windows OS Event Insight pack and the IBM Tivoli Monitoring Log File Agent to load and search Windows OS events. New support for data collection using Logstash provides an alternative to the IBM Tivoli Monitoring Log File Agent.

JavacoreInsightPack

The Java Core Insight Pack provides the capability to ingest and search metadata that originates in Java Core files in IBM Operations Analytics - Log Analysis. The field `logsourceHostname` has been changed to `datasourceHostname`.

GAIInsightPack

The Generic Annotation Insight Pack is not specific to any particular log data type. It can be used to analyze log files for which a log-specific Insight Pack is not available

DB2 Insight Pack

A DB2 Insight Pack is provided with IBM Operations Analytics - Log Analysis.

The Insight Pack includes support for ingesting and performing metadata searches against the DB2 version 9.7 and 10.1 `db2diag.log` files.

This document describes the version of the DB2 Insight Pack that is installed when you install IBM Operations Analytics - Log Analysis. An updated version of the DB2 Insight Pack may have been published after this version of IBM Operations Analytics - Log Analysis. To download the latest versions of this Insight Pack and updated documentation, see <http://www.ibm.com/developerworks/servicemanagement/downloads.html>.

Best practices for db2diag files

DB2 creates multiple db2diag.log files based on the database configuration. Follow these best practices to handle multiple db2diag logs for the rotating and multiple database partition scenarios.

Scenario 1: Database spread across partitions and members

If the database is spread across multiple partitions and members, then a db2diag file is created in multiple directories according to the DIAGPATH value. It can be difficult to interpret the DIAGPATH and db2nodes.cfg to find all the log files for each member and host. The best practice recommendation is to use the **db2diag** tool, which consolidates all the log records in to a single file.

For databases spread among multiple partitions and members, consolidate the records from all db2diag log files on all your database partitions, run the following command:

```
db2diag -global -output filename.log
```

Where *filename.log* is the name of the consolidated log file, such as db2diag_myglobalrecords.log.

Scenario 2: Rotating log files

If DB2 is configured for rotating log files, the files are named dynamically as db2diag.<number>.log. The best practice is to change the default log agent configuration files to recognize the rotating logs.

If you choose to consolidate the rotating logs with the db2diag tool, follow the *DB2 data loading best practice* about loading the consolidated file.

Using the Data Collector client to load the consolidated rotating logs can improve data ingestion performance. However, you must filter the logs based on timestamps to avoid ingesting duplicate logs each time you consolidate the logs.

Optional: Filtering db2diag log file records can reduce the time required to locate the records needed when troubleshooting problems. If the db2diag tool is used to filter records, only those records included in the filter result will be annotated by the IBM Operations Analytics tools - all other records are excluded. The **db2diag** tool includes other options for filtering and formatting the log records.

If you choose to consolidate the rotating logs with the db2diag tool, use the Data Collector as described in the *DB2 data loading best practice* about loading the consolidated file.

More information about the **db2diag** utility can be found at the DB2Information Center here:

For version 10.1:

http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.admin.trb.doc/doc/c0020701.html

For version 9.7:

http://www-01.ibm.com/support/knowledgecenter/SSEPGG_9.7.0/com.ibm.db2.luw.admin.trb.doc/doc/c0020701.html

Note: To find the locations of the individual logs, use the DB2 command:

```
GET DATABASE MANAGER CONFIGURATION
```

On **UNIX** systems, The DIAGPATH parameter shows the diagnostic data directory path that contains the log files. The default directory is:

```
INSTHOME/sqllib/db2dump,
```

where *INSTHOME* is the home directory of the DB2 instance.

On **Windows** systems, the default path depends on the operating system. To find the default path on Windows, use the command:

```
DB2SET DB2INSTPROF
```

Configuration artifacts

The Insight Pack supports the DB2 timestamp format YYYY-MM-DD-hh.mm.ss Z. You can customize artifacts in the index configuration file.

Note: Data sources are not predefined. A user with administrator privileges must define at least one DB2 data source before the application can be used.

The following table lists the configuration artifacts that are provided with the Insight Pack for each log file.

Table 51. Insight Pack configuration artifacts	
Artifact	Name for the db2diag.log
Splitter rule set	DB2Diag-Split
Annotator rule set	DB2Diag-Annotate
Source type	DB2Diag
Collection	DB2Diag-Collection1

Log File Agent configuration

The supported log files share IBM Tivoli Monitoring Log File Agent (LFA) configuration files. The following LFA configuration files are in the <HOME>/IBM-LFA-6.30/config/lo directory:

- DB2InsightPack-lfadb2.conf: Configuration file for the DB2 log file agent.
- DB2InsightPack-lfadb2.fmt: Matches records for the db2diag log files.

Splitting and annotation AQL modules

Splitting and annotation are handled by the following Annotation Query Language (AQL) modules.

Table 52. Insight Pack AQL modules	
AQL Module	Description
common	Common code module that is used across multiple log files (for example, to recognize timestamp formats).
commonDB2	Common annotations module that is used across splitter and annotator modules.
annotatorDB2Diag	Annotator module for db2diag log files.
splitterDB2Diag	Splitter module for db2diag log files.

Log file formats

The basic formats of the DB2 log files are described here as a reference for users.

The basic format of db2diag.log file is:

```
timestamp recordId LEVEL: level(source)
PID : pid TID : tid PROC : procName
INSTANCE: instance NODE : node DB : database
APPHDL : appHandle APPID: appID
AUTHID : authID
EDUID : eduID EDUNAME: engine dispatchable unit name
FUNCTION: prodName, compName, funcName, probe: probeNum
MESSAGE : messageID msgText
CALLED : prodName, compName, funcName OSERR: errorName (errno)
RETCODE : type=retCode errorDesc
ARG #N : typeTitle, typeName, size bytes
... argument ...
DATA #N : typeTitle, typeName, size bytes
... data ...
```


Only these fields are present in all log records.

```
timestamp
timezone
recordID
level
pid
tid
FUNCTION
```

For more information about DB2 log file formats, see the following topic on the DB2 information centers:

The format for DB2 10.1 log files is documented in:

http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.admin.trb.doc/doc/c0020815.html

The format for DB2 v9.7 log files is documented in:

http://www-01.ibm.com/support/knowledgecenter/SSEPGG_9.7.0/com.ibm.db2.luw.admin.trb.doc/doc/c0020815.html

Log file splitters

The splitters provided with the Insight Pack are described here as a reference for users.

DB2diaglog splitter

The db2diaglog splitter uses timestamps to define the beginning and end of each log record.

Log file annotations

The annotations that are defined by the log file index configurations are described here.

The following table lists the index configuration files that are included in the Insight Pack.

Table 53. Index configuration files	
Log file	Index configuration file
db2diag.log	Included in the sourcetypes.json file

The following sections describe the fields that are defined in the index configuration file. These fields, or annotations, are displayed in the IBM Operations Analytics - Log Analysis Search workspace, and can be used to filter or search the log records. Fields are extracted from the fields of a log record or collected from metadata around the log file. Each table gives the names of the fields (these names correspond to fields in the IBM Operations Analytics - Log Analysis Search workspace), descriptions of how the related annotations are made, and the index configuration attributes assigned to the fields.

Log record annotations

The following table lists the index configuration fields that relate to log record annotations. Each field corresponds to part of a db2diag log record. The fields are listed in the order in which they appear in a log record.

Table 54. Log record index configuration fields		
Field	Description	Attributes
timestamp	The timestamp of the log record, which is located at the beginning of a line.	dataType = DATE retrievable = true retrieveByDefault = true sortable = true filterable = true searchable = true

Table 54. Log record index configuration fields (continued)

Field	Description	Attributes
recordID	The record identifier of the log record that follows the timestamp. The recordID of the log files specifies the file offset at which the current message is being logged (for example, 27204 and the message length (for example, 655) for the platform where the log was created.	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
diagnosticLevel	A diagnostic level of the log record that follows the label LEVEL :. It is the diagnostic level that is associated with an error message. For example, Info, Warning, Error, Severe, or Event." Not all log records have a diagnostic level.	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = true searchable = true
processID	The process identifier in the log record that follows the PID: label. For example, 1988.	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
threadID	The thread identifier (TID) for the process in the log record that follows the TID: label. For example, 1988	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
processName	The name of the process in the log record that follows the PROC: label. For example, db2iCacheFunction.exe	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
instance	The DB2 instance that generated the message in the log record that follows the INSTANCE: label. For example, DB2.	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
node	The node identifier that generated the message for a multi-partition system. Otherwise it is 000. It follows the NODE: label. For example, 001.	dataType = TEXT retrievable = true retrieveByDefault = true sortable = true filterable = false searchable = true
databaseName	If present, the database name in the log record, that follows the DB: label. For example, DB2.	dataType = TEXT retrievable = true retrieveByDefault = true sortable = true filterable = true searchable = true

Table 54. Log record index configuration fields (continued)

Field	Description	Attributes
applicationHandle	If present, the application handle in the log record, that follows the APPHDL: label. For example, 0-2772.	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
applicationID	If present, the application identifier in the log record, that follows the APPID: label. For example, *LOCAL.DB2.130226175838.	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
authorizationID	If present the authorization user identifier in the log record, that follows the AUTHID: label. For example, adminuser1.	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
eduID	The engine dispatchable unit identifier in the log record that follows the EDUID: label. For example, 2004.	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
eduName	The name of the engine dispatchable unit in the log record that follows the EDUNAME: label. For example, db2agent (instance).	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
functionProductName	The product name that wrote the log record. It follows the FUNCTION: label, which also includes the component name, function name, and function probe point. For example, DB2.	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
functionComponentName	The name of the component that wrote the log record. It follows the product name in the FUNCTION: label, which also includes the product name, function name, and function probe point. For example, UDB	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
functionFunctionName	The name of the function that wrote the log record. It follows the component name in the FUNCTION: label, which also includes the product name, component name, and function probe point. For example, Self tuning memory manager	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true

Table 54. Log record index configuration fields (continued)		
Field	Description	Attributes
functionInfo	The information that is returned by the function in the log record. It follows the FUNCTION: label. It includes all information after the FUNCTION entry. For example, DATA #1 : unsigned integer, 8 bytes	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
functionProbePoint	The probe point within the function that wrote the log record. It follows the probe: label in the FUNCTION: label, which also includes the product name, component name, and function information. For example, probe:1008	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
message	The message that follows the MESSAGE: label. It is optional data that a function can provide in the log record. For example, New STMM log file (C:\ProgramData\IBM\DB2\DB2COPY1\DB2\stmmlog\stmm.0.log) created automatically.	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
sqlcode	The SQL code is optional data that is provided by the function in the log record. It is preceded by the text SQLCODE or sqlcode.	dataType = LONG retrievable = true retrieveByDefault = true sortable = true filterable = false searchable = true
msgClassifier	The message ID if it exists in a message (which follows MESSAGE: label). It starts with 3 or 4 letters, followed by 4 numbers, followed by I, E, or W such as ADM0506I.	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = true searchable = true
DB2Hostname	The hostname following the HOSTNAME: label where the DB2 log record was generated. If there is only one DB2 server, there is no hostname in the log record. For example, mydb2host.tiv.pok.ibm.com	dataType = TEXT retrievable = true retrieveByDefault = true sortable = true filterable = false searchable = true
start	This is the message following the label START provided by the function. It is an indication of the start of an event. For example, Starting FCM Session Manager.	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
stop	This is the message follows the label STOP provided by the function. It is an indication of the end of an event. For example, DATABASE: DTW : DEACTIVATED: NO.	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true

Metadata annotations

The following table lists the index configuration fields that relate to metadata annotations.

Table 55. Metadata index configuration fields		
Field	Description	Annotation attributes
application	The application name populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
hostname	The host name populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: true filterable: true searchable: true
logRecord	The entire log record output by the splitter.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
datasourceHostname	The host name specified in the data source.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: true searchable: true
middleware	The middleware name populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
service	The service name populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true

Searching DB2 log files for a return code example

You can search the log files for keywords. Search results are displayed in a timeline and a table format. You can use the search function to find fields that are not indexed.

Before you begin

To search a log file, you must first define a data source as the db2diag.log file.

This example shows how to search for RETCODE. For more information about other keywords you can search for in the FUNCTION information, see the DB2information center for your version here:

For version 10.1:http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.admin.trb.doc/doc/c0020815.html

For version 9.7:http://www-01.ibm.com/support/knowledgecenter/SSEPGG_9.7.0/com.ibm.db2.luw.admin.trb.doc/doc/c0020815.html

Procedure

1. From the **Search** workspace, click **Add Search** to create a tab that contains your search criteria.
2. Optional: To limit the extent of the search to specific data sources and any descendant data sources, select a leaf node from the data source tree.
3. In the **Time Filter** pane, click the **Time Filter** list and select the time period for which you want to search. Select **Custom** to specify a start time and date, and an end time and date for your search.
4. In the **Search** field, type the return code string that you want to search for in the log files.

For example, to search for return codes related to missing files include the string `FILE_DOESNT_EXIST`. The full return code for this example is:

```
RETCODE : ECF=0x9000001A=-1879048166=ECF_FILE_DOESNT_EXIST
```

To search for strings that related to missing files, type `FILE_DOESNT_EXIST` in the **Search** field.

5. Click **Search**.

For more information about searching logs, see the section *Searching log files* in the [IBM Operations Analytics - Log Analysis Knowledge Center](#)

Results

A graph that shows the distribution of matching events in the log is displayed.

DB2 data loading best practice

Best practice recommendation for DB2 Insight Pack data loading.

There are different data loading recommendations depending on whether you used the rotating or consolidated DB2 `db2diag.log` files and how many servers you need to monitor. To set up the data loading, you need to consider:

DB2 configuration

Is DB2 configured to produce a single log file for a single server, rotating log files for a single server, or multiple logs for multiple servers?

You can use the `db2diag` tool that is included with DB2 to merge and consolidate the log files. More information about the `db2diag` utility can be found in the DB2 documentation at:

For version 10.1:

http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.admin.trb.doc/doc/c0020701.html

For version 9.7:

http://www-01.ibm.com/support/knowledgecenter/SSEPGG_9.7.0/com.ibm.db2.luw.admin.trb.doc/doc/c0020701.html

Data loading

Determine if you want to use the IBM Tivoli Monitoring Log File Agent (LFA) installed on the remote DB2 server to push data or to use the LFA installed on the local IBM Operations Analytics - Log Analysis server to pull data. In some scenarios, you must use the Data Collector client as described in scenario 3.

Logfile agent configuration

Use the logfile agent configuration files to specify the log files you want to monitor. The `DB2InsightPack-lfadb2.conf` and `DB2InsightPack-lfadb2.fmt` files are located in the directory:

```
<HOME>/IBM-LFA-6.30/config/lo
```

The log file scenarios here describe the specific settings for these files.

Scenario 1 - Individual log file on one DB2 Server

For a single log file on one DB2 server follow these best practices.

DB2 Configuration

DB2 is configured for a single log file (non-rotating), db2diag.log on one server

Data Loading Method

The recommended method for loading data is to use the LFA installed on the remote DB2 server to push data or to use the LFA installed on the local IBM Operations Analytics server to pull data.

Logfile Agent Configuration - DB2InsightPack-1fadb2.conf file

In the DB2InsightPack-1fadb2.conf file, specify the following parameters to monitor the log files.

```
LogSources=<db2 log directory to monitor>/db2diag.log
#FileComparisonMode
```

The FileComparisonMode parameter should be commented out since it only applies when using wildcards in a LogSources parameter

Logfile Agent Configuration - DB2InsightPack-1fadb2.fmt file

Use the default DB2InsightPack-1fadb2.fmt file.

```
// Matches records for any Log file:
//

REGEX AllRecords
(.*)
hostname LABEL
-file FILENAME
RemoteHost DEFAULT
logpath PRINTF("%s",file)
text $1
END
```

Scenario 2 - log file rotation on one DB2 server

For rotated log files on a single DB2 server follow these best practices.

DB2 configuration

DB2 is configured for rotating log files using the DIAGSIZE configuration option. The db2diag.log files are named dynamically as db2diag.<n>.log.

Data Loading Method

The recommended method for loading data is to use the IBM Tivoli Monitoring Log File Agent (LFA) installed on the remote DB2 server to push data or to use the LFA installed on the local IBM Operations Analytics - Log Analysis server to pull data.

Logfile Agent Configuration - DB2InsightPack-1fadb2.conf file

In the DB2InsightPack-1fadb2.conf file, specify the following parameters to monitor the rotating log files:

```
LogSources=<db2 log directory to monitor>/db2diag.*.log
FileComparisonMode=CompareByAllMatches
```

Logfile Agent Configuration - DB2InsightPack-1fadb2.fmt file

Use the following DB2InsightPack-1fadb2.fmt file to specify a fixed log file name. Otherwise you must define multiple logsources in the IBM Operations Analytics - Log Analysis Administrative Settings page because the rotating log file name changes. The fmt file allows a fixed file name in the logpath.

```
// Matches records for any Log file:
//

REGEX AllRecords
(.*)
hostname LABEL
```

```
-file db2diag.log
RemoteHost DEFAULT
logpath PRINTF("%s",file)
text $1
END
```

Scenario 3 - Consolidating log files from multiple DB2 servers

If you consolidate log files from multiple DB2 servers follow these best practices.

DB2 Configuration

If the database is spread across multiple partitions and members, then a db2diag.log file is created in multiple directories according to the DIAGPATH value. It can be difficult to interpret the DIAGPATH and db2nodes.cfg to find all the log files for each member and host. The best practice recommendation is to use the db2diag tool, which will bring the information from all the members together to create a consolidated db2diag.log. The db2diag utility allows you to filter based on timestamp and this should be done to include only new log entries in the consolidated logs. Information on this filter can be found here:

For version 9.7:

http://www-01.ibm.com/support/knowledgecenter/SSEPGG_9.7.0/com.ibm.db2.luw.admin.cmd.doc/doc/r0011728.html

For version 10.1

http://www-01.ibm.com/support/knowledgecenter/SSEPGG_10.1.0/com.ibm.db2.luw.admin.cmd.doc/doc/r0011728.html

Data Loading Method

The recommended method for loading data is to use the Data Collector client. Remove the previous consolidated db2diag.log file before creating or copying a new version into the directory from which you load data.

Generic Annotation Insight Pack

A Generic Annotation Insight Pack is installed when you install IBM Operations Analytics - Log Analysis. This Insight Pack is not specific to any particular log data type. It can be used to analyze log files for which a log-specific Insight Pack is not available.

The Insight Pack facilitates data ingestion and metadata searches of logs files where a date and time stamp can be identified within the log records.

This document describes the version of the Generic Annotation Insight Pack that is installed when you install IBM Operations Analytics - Log Analysis. An updated version of the Generic Annotation Insight Pack may have been published after this version of IBM Operations Analytics - Log Analysis. To download the latest versions of this Insight Pack as well as updated documentation, see <http://www.ibm.com/developerworks/servicemanagement/downloads.html>

Generic Annotation installation

Instructions on how to install the Generic Annotation Insight Pack.

Procedure

1. Upload the Generic Annotation Insight Pack archive file, GenericAnnotationInsightPack_<version>.zip, to the system where IBM Operations Analytics - Log Analysis is installed.

Where <version> is the version of the Generic Annotation Insight Pack.

2. Install the Generic Annotation Insight Pack with the pkg_mgmt.sh command:

```
<HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -install <path>/
GenericAnnotationInsightPack_<version>.zip
```

Where <path> is the path where you saved the Generic Annotation Insight Pack.

3. Deploy the log file agent with the following command:

```
<HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -deploylfa <path>/  
GenericAnnotationInsightPack_<version>.zip
```

Related tasks

[Importing an Insight Pack](#)

Configuration artifacts

You must create at least one data source.

Data sources are not defined by default. Create at least one data source before the logs can be ingested.

The following table lists the configuration artifacts that are provided with the Insight Pack for each log file.

Table 56. Configuration Artifacts	
Insight Pack	Configuration artifact
Splitter Rule Set	dateTime-Split
Splitter Rule Set	timeOnly-Split
Annotator Rule Set	Generic-Annotate
Source Type	Generic
Collection	Generic-Col1

The following table lists the configuration artifacts that are provided with the Insight Pack for each log file. The Insight Pack supports a variety of timestamp formats. See the section [“Timestamp formats”](#) on page 261 for a complete list of the available formats.

Table 57. Configuration Artifacts. Table 2 lists the configuration artifacts that are provided with the Insight Pack.			
Splitter rule set	Annotator rule set	Source type	Collection
Generic-dateTime-Split	Generic-Annotate	Generic	Generic-Collection1
Generic-timeOnly-Split			
NormalizedMonthFirst-Split		Generic-NormalizedMonthFirst	Normalized-MonthFirst
NormalizedDayFirst-Split		Generic-NormalizedDayFirst	Normalized-DayFirst
NormalizedYearFirst-Split		Generic-NormalizedYearFirst	Normalized-YearFirst

Log file formats

The Generic Annotation Insight Pack annotates all log files irrespective of their format.

Log File Agent configuration

The supported log files share IBM Tivoli Monitoring Log File Agent (LFA) configuration files. The following LFA configuration files are in the Log_Analytics_install_dir/IBM-LFA-6.30/config/lo directory:

Log file splitter

The Generic Annotation Insight Pack contains Rule Sets that can be used to split incoming log files.

These are:

- Generic-dateTime-Split (default)
- Generic-timeOnly-Split

Each Rule Set splits a log based on each line having either a time stamp or a date and time stamp.

The Generic-dateTime-split splitter splits log records using the date and time stamp of the log file. If the log file does not have year format that the splitter can interpret in the log records, the splitter adds a year value based on the IBM Operations Analytics - Log Analysis server system time. The Index Configuration must be updated to reflect this action.

The Generic-timeOnly-split splitter splits log records using only the time stamp in the log record. Where the log file does not have a date in the log records that can be interpreted by splitter, the current date value set for the IBM Operations Analytics - Log Analysis server is used. The format MM/dd/yyyy is inserted before the format of the time. The Index Configuration must be updated to reflect this action.

The splitters provided with the Insight Pack are described here as a reference for users.

DateTime splitter

The dateTime splitter recognizes all supported timestamp formats. The timestamp must have a date and a time. If the year is missing from the date, the current year will be appended to the front of the timestamp. You must modify the index configuration with the proper timestamp format for the splitter to function properly.

TimeOnly splitter

The timeOnly splitter recognizes all supported time formats. The timestamp must have a time and must not have a date. The splitter will append the current date to the front of the timestamp in the format MM/dd/yyyy. You must modify the index configuration with the proper timestamp format for the splitter to function properly.

NormalizedMonthFirst splitter

The splitter assumes a purely numeric date (for example, 07/08/09) is in the format MM/dd/yy. The timestamp must have a time, and may have an optional date. The date may have an optional year. If the date or year is missing, the current date or year is substituted. The NormalizedMonthFirst splitter outputs the timestamp in a normalized format. As a result, the index configuration does not need to be modified with the timestamp format.

NormalizedDayFirst splitter

The splitter assumes a purely numeric date (for example, 07/08/09) is in the format dd/MM/yy. The timestamp must have a time, and may have an optional date. The date may have an optional year. If the date or year is missing, the current date or year is substituted. The NormalizedDayFirst splitter outputs the timestamp in a normalized format. As a result, the index configuration does not need to be modified with the timestamp format.

NormalizedYearFirst splitter

The splitter assumes a purely numeric date (for example, 07/08/09) is in the format yy/MM/dd. The timestamp must have a time, and may have an optional date. The date may have an optional year. If the date or year is missing, the current date or year is substituted. The NormalizedYearFirst splitter outputs the timestamp in a normalized format. As a result, the index configuration does not need to be modified with the timestamp format.

Log file annotations

The Generic annotator allows you to search and analyze log files for which a specific annotator is not available. There are two types of annotations created by the Generic annotator. Those are Concepts and

Key-Value pairs. This section outlines the purpose, scope, and use of the IBM Operations Analytics - Log Analysis Generic annotator.

Included concept tokens

A concept is a piece of text that represents a real world entity such as an IP address or a hostname. These concepts are useful for searches as they provide information that can assist you in diagnosing issues. The Generic annotator includes support for these annotation tokens:

Hostname

Names given to devices that connect to a network and that are referenced in the log record.

IP Address

Numeric labels given to devices that are connected to a network and that are referenced in the log record

Severity Level

The indicator of the severity of an event in a log record. The Generic annotator provides annotation for these severity levels:

- SUCCESS
- TRACE
- DEBUG
- INFO
- WARN
- ERROR
- FATAL
- OFF
- CRITICAL
- CRITICAL_ERROR
- SEVERE
- IGNORE
- WARNING
- CONFIG
- FINE
- FINER
- FINEST
- ALL

URL

Web addresses listed in the log record.

Identifier

Patterns intended to capture names of constants that might repeat within the log record and that signify the occurrence of some event. For example, ECH_PING_FAIL_BCKP. The Generic annotator assumes that an identifier is a sequence of alphanumeric characters in capitals that may be separated by underscores.

Excluded concept tokens

The Generic annotator assumes that these tokens are noise and they are ignored:

Date and time

For the purposes of analysis date and time are not useful.

Number

The Generic annotator ignores both whole and decimal numbers.

Hexadecimal numbers

The Generic annotator ignores hexadecimal numbers such as 7AB87F.

Stop words

A list of stop words have been defined for the Generic annotator. This is to allow the Generic annotator to ignore common words that might appear frequently, but offer no value in an analysis of the log records.

Key-value pairs

A Key-value annotation extracts data from a log record if it is in the format `<key> = <value>`. For example, `ERROR-CODE = 4499`. These Key-value pairs can be used to list the values for each key. These limitations apply to Key-value pair annotations:

Colon separator

Key-value pairs that are separated by a colon are excluded. For example, `Label : ECH_PING_FAIL_BCKP`.

Hyphen prefix

Key-value pairs where the value begin with a hyphen are excluded. For example, `ERRORCODE = -4499`.

Numbers with commas

Key-value pairs where the value includes a comma are excluded. For example, `ERRORCODE = 4,499`.

Forward and backward slash characters

Key-value pairs where the value contains a forward or backward slash are excluded. For example, `path = /opt/IBM/`.

Quotes

Key-value pairs where the value is contained within quotation marks. For example, `language = "English"`.

Delimiter characters

Some limitations exist where the value in a Key-value pair contains a delimiter. However, these depend on the whether the value contains a token that can be annotated based on the list of included tokens. For example, `Time = Thu Nov 22 06:28:48 EST 2012` is delimited by a space after Thu and therefore the Key-value pair is assumed to be `Key = Time, Value = Thu`. However, a Date and Time annotator can annotate the full value to give a value of `Key = Time, Value = Thu Nov 22 06:28:48 EST 2012`.

Key-value pairs

A Key-value annotation extracts data from a log record if it is in the format `<key>=<value>`. For example, `ERROR-CODE = 4499`. These Key-value pairs are used to list the values for each key in the Discovered Patterns section of the Search UI.

There are two categories of KVP annotations. The first is Key-value pairs that are separated by an equal sign (=). The second category is those separated by a colon (:). Each category has a different set of rules to determine what is a valid annotation.

Key-value pairs separated by an equal sign, '='

- Both the key and value must be one token
- The key can contain upper and lower case letters, dashes (-), underscores (_), and periods (.)
- The key must begin with a letter
- The value can contain upper and lower case letters, numbers, dashes (-), underscores (_), periods (.), at signs (@), and colons (:)
- The value can be surround by matching brackets [], parentheses (), angle-brackets < >, single quotes ', or double quotes “ ”
- The value must begin with a letter or a number, and may have an optional dash (-) at the beginning

- A single whitespace character may be on one or both sides of the equal sign
- The single token rule for the value is disregarded when a concept is found for the value. For example, if a multi token date is identified as the value, the whole date, not just the first token, will be annotated.
- Users may add custom regular expressions to the dictionary located at `Log_Analytics_install_dir/unity_content/GA/GAInsightPack_<version>/extractors/ruleset/GA_common/dicts/userSpecifiedStrings.dict`. Matches to these regular expressions will be used when checking if the value is part of a larger concept.

Key-value pairs separated by a colon, ':'

- Both the key and value must be between 1 and 3 tokens
- The tokens can contain any character except whitespace or colons.
- Tokens must be separated by spaces or tabs
- The colon may have one or more spaces or tabs to the left and must have at least one space or tab to the right. There may be more than one space or tab to right of the colon
- The entire string must be on a line by itself

Timestamp formats

IBM Operations Analytics - Log Analysis is capable of annotating many commonly used timestamp formats. This appendix lists the supported timestamp formats.

Timestamp annotations, also called date-time annotations, are constructed from two base formats: a date format and a time format. Date-time annotation works as follows:

1. An annotator identifies date patterns in a text and annotates them with date annotations.
2. Another annotator identifies time patterns in the text and annotates them with time annotations.
3. The timestamp annotator then identifies and annotates specific patterns in which date and time annotations occur contiguously in the text.

Date annotation formats

The date annotation formats are specified in the `Date_BI.aql` file.

View: `DateOutput`

File name: `Date_BI.aql`

The following date annotation formats are available.

Table 58. Date annotation formats		
Format name	Pattern	Examples
D1	A date interval, where: <ul style="list-style-type: none"> • the month is a supported month format • the month can precede or follow the interval • the year is optional • commas (,) are optional 	9 - 12 December 2012 December 9 – 12, 2012 9 - 12 december DEC 9 - 12
D2	A date that contains the suffixes th, st, nd, or rd, and where: <ul style="list-style-type: none"> • the word "of" is optional • the year is optional • commas (,) are optional • the month is a supported month format • the month can precede or follow the day 	3rd December 2012 4th DEC Dec, 1st 2nd of December

Table 58. Date annotation formats (continued)

Format name	Pattern	Examples
D3	<p>A date that contains the day of the week, and where:</p> <ul style="list-style-type: none"> the word "the" is optional the suffixes th, st, nd, and rd are optional the year is optional commas (,) are optional the month is a supported month format the month can precede or follow the day 	<p>Sunday, the 3rd of December, 2012</p> <p>Wed, 1st DEC</p>
D4	<p>A date that contains forward-slash characters (/), in the format Day/Month/Year, and where:</p> <ul style="list-style-type: none"> the month is a supported month format the month follows the day the year has four digits the suffixes th, st, nd, and rd are optional 	<p>3/December/2012</p> <p>1st/Dec/2012</p>
D5	<p>A date in the format Year-Month-Day or Year.Month.Day, where:</p> <ul style="list-style-type: none"> the year has four digits the month has two digits the day has two digits <p>Because this pattern comprises only digits, leading zeros (0) might be required.</p>	<p>2012-01-30</p> <p>2012.12.03</p>
D6	<p>A date in the format Day-Month-Year or Day/Month/Year, where:</p> <ul style="list-style-type: none"> the year can have two or four digits the month and the day do not require leading zeros (0), even if they are single digits 	<p>30-1-12</p> <p>3/12/2012</p>
D7	<p>A date in the format Month-Day-Year or Month/Day/Year, where:</p> <ul style="list-style-type: none"> the year can have two or four digits the month and the day do not require leading zeros (0), even if they are single digits 	<p>1-30-12</p> <p>12/3/2012</p>

Time annotation formats

The time annotation formats are specified in the `Time_BI.aql` file.

View: `TimeOutput`

File name: `Time_BI.aql`

The following time annotation formats are available.

Table 59. Time annotation formats

Format name	Pattern	Examples
T1	A time in the format Hour:Minute, where: <ul style="list-style-type: none"> the hour need not be padded with leading zeros (0) if it is a single digit seconds and milliseconds are optional the minute is two digits the second, if present, is two digits and preceded by a colon (:) or a period (.) the millisecond, if present, is three digits and preceded by a colon (:) or a period (.) 	2:05 20:30 02:05 23:11:59 23:11:59:120 23:11:59.120
T2	A time in T1 format, plus the year, where the year is four digits.	2:05 2012 23:11:59.120 2012
T3	A time in the format Hour:Minute:Seconds, where: <ul style="list-style-type: none"> milliseconds are optional the time zone is optional the minute is two digits the second is two digits and preceded by a colon (:) or a period (.) the millisecond, if present, is three digits and preceded by a colon (:) or a period (.) the time zone, if present, is: <ul style="list-style-type: none"> preceded by a plus (+) or minus (-) sign has no space preceding the plus (+) or minus (-) sign has the format Hour:Minute, where both the hour and minute are two digits contains no intervening spaces 	3:11:59+05:30 2:05:59.120-01:05
T4	A time in T1 format, plus a 12-hr clock designation.	2:05 PM 11:11:59 a.m.
T5	A time in T1 format, plus a supported time zone format.	2:05 IST 11:11:59 PST
T6	A time in T1, T4, or T5 format, plus the time zone and year, where: <ul style="list-style-type: none"> the time zone is optional the year is optional the time zone, if present: <ul style="list-style-type: none"> is in digits is preceded by a plus (+) or minus (-) sign contains an optional colon (:) between the hour and the minute the year, if present, is four digits 	11:11:59 a.m. +05:30 11:11:59 PST -0030 2012

Date-time annotation formats

The date-time (timestamp) annotation formats are specified in the `DateTime-consolidation_BI.aql` file.

View: `DateTimeOutput`

File name: `DateTime-consolidation_BI.aql`

The following date-time annotation formats are available.

Table 60. Date-time annotation formats		
Format name	Pattern	Examples
DT1	A timestamp in the format DT, where: <ul style="list-style-type: none">• D is the D1, D2, D3, D4, D5, D6, or D7 date format• T is the T1, T2, T3, T4, T5, or T6 time format	Sunday, the 3rd of December, 2012 23:11:59.120 IST 3/12/2012 2:11:59+05:30 6/10/12 2:48:28:381 MDT Thu Nov 22 06:28:48 EST 2012
DT2	A timestamp in the format D4:T3.	3/December/2012:2:00:00.000 1st/Dec/2012:23:11:59+05:30
DT3	A timestamp in the format D4:Hour:Minute:Seconds Z, where Z is an RFC 822-four-digit time zone format that conforms with the Java <code>SimpleDateFormat</code> class.	3/December/2012:02:00:00 -0030 1st/Dec/2012:23:11:59 +0530

Other formats

The view `DateTimeOutput` also supports timestamp formats in these ISO date formats:

- `yyyy-MM-ddTHH:mm:ss.SSSZ`. For example, `2013-02-27T13:57:21.836+0000`
- `yyyy.MM.ddTHH:mm:ss.SSSZ`. For example, `2013.02.27T13:57:21.836+0000`

Variations of these formats are also supported:

- `yyyy/MM/dd-HH:mm:ss.SSSZ`. For example, `2013/02/27-13:57:21.123+0000`
- `yyyy/MM/dd-HH:mm:ss.SSS`. For example, `2013/02/27-13:57:21.123`
- `yyyy-MM-dd-HH:mm:ss`. For example, `2013-02-27-13:57:21`

Date-time stamps with no year value

Some applications write logs with no year in the date/time stamp. For example, the UNIX messages log, such as shown here:

```
Apr 24 09:41:16 bluewashmachine symcfgd: subscriber 2 has left -- closed
0 remaining handles

Apr 24 09:41:20 bluewashmachine rtvscand: New virus definition file loaded.
Version: 150423c.

Apr 24 09:41:38 bluewashmachine kernel: type=1400 audit(1366792898.697:52164):
avc: denied
{ module_request } for pid=18827 comm="smtpd" kmod="net-pf-10"
scontext=system_u:system_r:postfix_smtpd_t:s0
tcontext=system_u:system_r:kernel_t:s0 tclass=system

Apr 24 09:41:38 bluewashmachine kernel: type=1400 audit(1366792898.822:52165):
avc: denied
{ module_request } for pid=18833 comm="proxymap" kmod="net-pf-10"
scontext=system_u:system_r:postfix_master_t:s0
tcontext=system_u:system_r:kernel_t:s0 tclass=system
```


In this case, the Generic-dateTime-Split splitter identifies the string Apr 24 09:38:58 as a valid date-time stamp. To meet the required date formats of IBM Operations Analytics, a valid year must be associated with the date-time string. The generic-dateTime-split splitter address this problem by placing a yyyy value at the beginning of the identified date-time stamp format. As a result, the timestamp now reads 2013 Apr 24 09:38:58.

You must update the timestamp format for files this type in the indexConfig. For example, if you want IBM Operations Analytics to ingest log records with the timestamp format MMM dd HH:mm:ss, the dateFormat must be specified as shown here.

```
"timestamp": {
  "searchable": true,
  "filterable": true,
  "retrievable": true,
  "dataType": "DATE",
  "tokenizer": "literal",
  "sortable": true,
  "source": {
    "dateFormats": [
      "yyyy MMM dd HH:mm:ss"
    ],
    "paths": [
      "metadata.timestamp"
    ],
    "combine": "FIRST"
  },
  "retrieveByDefault": true
},
```

The supported date formats without a year are:

- Apr 16 (MMM dd)
- 16 Apr (dd MMM)
- 16 April (dd MMM)
- April 16 (MMM dd)

Year end scenario: The generic-dateTime-split splitter applies the current year to any timestamp that is ingested where no year can be discerned from the log record. The exception is when the log record is ingested where current system time of the IBM Operations Analytics server identifies the month as January, but the incoming date/timestamp is December. In such situations, the year value that is applied is the current year minus 1.

Logs with no date

Some data sources that are ingested by IBM Operations Analytics do not support a date within the timestamp. For example, some lines of a log display the characteristic shown here in bold:

```
00:11:35.103 INFO [main] - Server accepting connections on rmi://9.11.222.333:1099/
09:34:33.071 INFO [main] - Server accepting connections on tcp://9.11.222.333:3035/
```

To ingest such a data source, IBM Operations Analytics provides a splitter rule set Generic-timeOnly-Split, which you can use along with the Generic-Annotator to ingest such a log. The splitter prepends a date to the identified time from each record of the log.

You must update the timestamp format for files of this type in the indexConfig. For example, in order for IBM Operations Analytics to ingest a log with records with timestamp such as 09:34:33.071, the dateFormat must be specified as here.

```
"timestamp": {
  "searchable": true,
  "filterable": true,
  "retrievable": true,
  "dataType": "DATE",
  "tokenizer": "literal",
  "sortable": true,
  "source": {
    "dateFormats": [
```

```

        "MM/dd/yyyy HH:mm:ss.SSS"
    ],
    "paths": [
        "metadata.timestamp"
    ],
    "combine": "FIRST"
  },
  "retrieveByDefault": true
},

```

Only the date portion of the dateFormat is fixed. The time portion must reflect the format found in the incoming logSource.

The Generic annotator Insight Pack defines the Annotation Query Language (AQL) that supports log files with no date in the timeOnly splitter. The timeOnly splitter is defined in the file:

```
GAInsightPack_<version>/extractors/ruleset/timeOnlySplitter
```

Supported weekday formats

The supported weekday formats are specified in the wkday.dict dictionary file.

Dictionary file: dicts/wkday.dict

The following long and short forms of the days of the week are supported. Both lower and upper cases, and upper case for first character, are supported.

Table 61. Supported weekday formats	
Long form	Short form
monday	mon
tuesday	tue tues
wednesday	wed
thursday	thu thur thurs
friday	fri
saturday	sat
sunday	sun

Supported month formats

The supported month formats are specified in the month.dict dictionary file.

Dictionary file: dicts/month.dict

The following long and short forms of the months of the year are supported. Both lower and upper cases, and upper case for first character, are supported.

Table 62. Supported month formats	
Long form	Short form
january	jan
february	feb
march	mar
april	apr

Table 62. Supported month formats (continued)

Long form	Short form
may	No short form.
june	jun
july	jul
august	aug
september	sept sep
october	oct
november	nov
december	dec

Supported time zone formats

The supported time zone formats are specified in the `timeZone.dict` dictionary file.

Dictionary file: `dicts/timeZone.dict`

The following 12-hour clock formats, in both lower and upper case, are supported:

- a.m.
- p.m.
- AM
- PM

The following page on SMC lists the supported time zone formats: [What are the supported time zones?](#)

Note: If a supplied timezone is not supported, the system behaviour is to default to reading the system timezone and normalizing to that.

Javacore Insight Pack

The Javacore Insight Pack provides the capability to ingest and perform metadata searches against javacore files in IBM Operations Analytics - Log Analysis.

Support

This document describes the version of the Javacore Insight Pack that is installed when you install IBM Operations Analytics - Log Analysis. An updated version of the Javacore Insight Pack may have been published after this version of IBM Operations Analytics - Log Analysis. To download the latest versions of this Insight Pack as well as updated documentation, see <http://www.ibm.com/developerworks/servicemanagement/downloads.html>.

The Javacore Insight Pack supports the ingestion of Linux javacore files produced by the IBM JRE versions 6.0 and 7.0.

The Javacore Insight Pack can be run on Linux.

Installing the Javacore Insight Pack

Instructions on how to install the Javacore Insight Pack

About this task

The Javacore Insight Pack is installed using the `pkg_mgmt` utility.

Procedure

1. Upload the Javacore Insight Pack archive file, JavacoreInsightPack_<version>.zip, to the system where IBM Operations Analytics - Log Analysis is installed.
2. Install the Javacore Insight Pack with the pkg_mgmt.sh command:

```
<HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -install  
<path>/JavacoreInsightPack_<version>.zip
```

Where <path> is the path where you saved the Javacore Insight Pack.

Javacore configuration artifacts

The Javacore Insight Pack configuration artifacts.

Splitter file set:

Javacore-Split

Annotator file set:

Javacore-Annotate

Source type:

Javacore

Collection:

Javacore-Collection1

The Javacore Insight Pack does not provide any Log File Agent (LFA) configuration files (.conf and .fmt). Javacore files are ingested using the Data Collector Client.

Javacore log file splitter

The Javacore Splitter uses a filter to reduce the size of the data that is grouped into each record.

Javacore files contain a single timestamp and so are processed as one log record. The Javacore splitter ensures that the contents of the javacore file are grouped into a single log record. Javacore files contain a lot of information on the state of the JVM at the time of the javacore dump. As a result javacore files can be large in size. However, not all of this data needs to be indexed and annotated. The Javacore Splitter uses a filter to reduce the size of this data. The following entries in a javacore file are filtered by the splitter:

- 1TISIGINFO
- 1TIDATETIME
- 1TIFILENAME
- 2XHOSLEVEL
- 3XHCPUARCH
- 3XHNUMCPUS
- 1CIJAVAVERSION
- 1CIVMVERSION
- 1CIJITVERSION
- 1CIGCVERSION
- 1CIRUNNINGAS
- 1CICMDLINE
- 1CIJAVAHOMEDIR

The entire thread stack trace is also filtered. All other data is blocked by the filter.

Turning off the filter

The filter can be configured to be on or off. It is on by default. To turn off the filter, follow these steps:

1. Create a file called javacore_insightpack.config

2. Add the following key/value to the file: `splitter.filter.on=false`

3. Save the file and copy it to your home directory on the IBM Operations Analytics - Log Analysis system

Note: Turning off the filter will pass the entire javacore file contents into IBM Operations Analytics - Log Analysis. This will affect the performance of searches on the IBM Operations Analytics - Log Analysis Search workspace as the entire javacore file contents will be contained in the logRecord annotation.

Javacore log annotations

The fields that are defined in the index configuration file, or annotations, are displayed in the IBM Operations Analytics - Log Analysis Search workspace, and can be used to filter or search the log records.

Fields are extracted from the fields of a log record or collected from metadata around the log file.

Table 63. Log record annotations	
Field	Attributes
timestamp	<pre>dataType = TEXT retrievable = true retrieveByDefault = true sortable = true filterable = true searchable = true</pre>
SignalInfo	<pre>dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = true searchable = true</pre>
FileName	<pre>dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = true searchable = true</pre>
OSLevel	<pre>dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = true searchable = true</pre>
CPUArchitecture	<pre>dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = true searchable = true</pre>
NumCPUs	<pre>dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true</pre>
JavaVersion	<pre>dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = true searchable = true</pre>

Table 63. Log record annotations (continued)

Field	Attributes
VMVersion	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = true searchable = true
JITVersion	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = true searchable = true
GCVersion	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = true searchable = true
RunningAs	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = true searchable = true
CommandLineArgs	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = true searchable = true
JavaHomeDir	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = true searchable = true
NumThreadsRunning	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
NumThreadsSuspended	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true

Table 63. Log record annotations (continued)

Field	Attributes
NumThreadsBlocked	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
NumThreadsParked	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
NumThreadsConditionWaiting	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
CurrentThreadStacktrace	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
AllThreadStacktraces	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true

Table 64. Metadata annotations

Field	Attributes
application	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
middleware	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true
datasourceHostname	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true

Table 64. Metadata annotations (continued)	
Field	Attributes
hostname	<pre> dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true </pre>
service	<pre> dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true </pre>

Ingesting Java core log data

About this task

To ingest a javacore file perform the following steps:

Note: This assumes that the logsource for the Javacore Insight Pack has already been created.

Procedure

1. Edit the file: <HOME>/utilities/datacollector-client/javaDatacollector.properties:
 - a) Edit the **logFile** value to correspond to the name (including path) of the javacore file to be ingested.
 - b) Edit the **logpath** value to be the same as the Javacore Logsource log path value.
2. Run the following command:


```
<HOME>/ibm-java/bin/java -jar datacollector-client.jar
```

Syslog Insight Pack

The Syslog Insight Pack extends IBM Operations Analytics - Log Analysis functionality so it can ingest and perform metadata searches against syslog data logging.

This document describes the version of the Syslog Insight Pack that is installed when you install IBM Operations Analytics - Log Analysis. An updated version of the Syslog Insight Pack might have been published after this version of IBM Operations Analytics - Log Analysis. To download the latest versions of this Insight Pack and updated documentation, see <http://www.ibm.com/developerworks/servicemanagement/downloads.html>.

The formatted log includes specific property values in a name-value pair format to aid data ingestion.

Syslog is a standard for recording events to track system activity and to diagnose problems. It separates the software that generates messages from the system that stores them and the software that reports and analyzes them. Implementations are available for many operating systems. Specific configuration permits the direction of messages to various devices (console), files (/var/log/), or remote syslog servers. rsyslog is an open source software utility that is used on UNIX and Unix-like computer systems for forwarding log messages in an IP network. It implements the basic syslog protocol.

Supported versions

The Syslog Insight Pack can be installed with IBM Operations Analytics - Log Analysis 1.1.0.0 and higher.

The following two source types are supported:

Syslog_default

The Syslog_default source type configures /var/log/messages files.

Syslog_custom

The Syslog_custom source type configures rsyslog files.

rsyslog version 3 is included as the default syslog tool for RHEL 5.2, and this is the minimum version that is supported by IBM Operations Analytics - Log Analysis. IBM Operations Analytics - Log Analysis supports rsyslog version 3, 5, 6 and 7. IBM Operations Analytics - Log Analysis supports the rsyslog list format, which is recommended by rsyslog, for version 7 and higher of rsyslog.

Note: Some applications write logs with no year in the date stamp. For example, `mmm dd hh:Mi:ss`. To meet the required date format for IBM Operations Analytics - Log Analysis, the system year value is used as the year during data ingestion

Syslog installation

Instructions on how to install the Syslog Insight Pack.

Procedure

1. Create a directory called `<HOME>/IBM/LogAnalysis//unity_content/Syslog` on the system where IBM Operations Analytics - Log Analysis is installed and upload the Syslog Insight Pack archive file, `SysLogInsightPack_<version>.zip`, to that directory.
2. Install the Syslog Insight Pack with the `pkg_mgmt.sh` command:

```
<HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -install <path>
/SysLogInsightPack_<version>.zip
```

Where `<path>` is the path where you saved the Syslog Insight Pack.

3. Deploy the log file agent with the following command:

```
<HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -deploylfa <path>
/SysLogInsightPack_<version>.zip
```

Where `<path>` is the path where you saved the Syslog Insight Pack.

Related tasks

[Installing an Insight Pack](#)

Syslog configuration

Configuration of the Insight Pack are described here as a reference for users.

rsyslog requirements

Before ingesting rsyslog log files, both rsyslog and IBM Operations Analytics - Log Analysis must be configured to ensure that rsyslog log files are output in a format that can be processed by IBM Operations Analytics - Log Analysis Syslog Insight Pack. The Syslog_custom source type configures rsyslog files.

Add the `scalaLogFormat` template to rsyslog:

- For rsyslog 7 and higher, which support the list format:

1. Open the `/etc/rsyslog.conf` for edit.
2. Add the following template:

```
template(name="scalaLogFormat" type="list") {
    property(name="timestamp" dateFormat="rfc3339")
    constant(value=" host=")
    property(name="hostname")
    constant(value=", relayHost=")
    property(name="fromhost")
    constant(value=", tag=")
    property(name="syslogtag")
    constant(value=", programName=")
    property(name="programname")
    constant(value=", procid=")
    property(name="procid")
    constant(value=", facility=")
    property(name="syslogfacility-text")
}
```

```

constant(value=", sev=")
property(name="syslogseverity-text")
constant(value=", appName=")
property(name="app-name")
constant(value=", msg=")
property(name="msg" )
constant(value="\n")
}

```

The generated log record is formatted as

```

2013-07-15T21:30:37.997295-04:00 host=co052065, relayHost=co052065,
tag=rhnsd[12171]:, programName=rhnsd, procid=12171, facility=daemon,
sev=debug, appName=rhnsd, msg= running program /usr/sbin/rhn_check

```

3. Associate the scalaLogFormat template with the log files to be ingested by IBM Operations Analytics - Log Analysis. It will log all log entries to *<filename>* in addition to any other associations in the configuration file. For example:

```

*.* /var/log/<filename>.log;scalaLogFormat

```

4. Restart the rsyslog daemon.

For more information about restarting the rsyslog daemon, see <http://rsyslog.com/doc>.

5. Ensure that the output file created for IBM Operations Analytics - Log Analysis can be read by your IBM Operations Analytics - Log Analysis user, and write that file directly to the monitored logsource directories.

For example:

```

*.* <HOME>/logsources/SyslogInsightPack
SCALA.log;scalaLogFormat

```

- For versions of rsyslog older than version 7:
 1. Open the `/etc/rsyslog.conf` for edit.
 2. Add the following template (legacy format):

```

$template scalaLogFormat,"%TIMESTAMP:::date-rfc3339% host=%HOSTNAME%,
relayHost=%FROMHOST%, tag=%syslogtag%, programName=%programname%,
procid=%PROCID%, facility=%syslogfacility-text%, sev=%syslogseverity-text%,
appName=%APP-NAME%, msg=%msg%\n"

```

The generated log record is formatted as

```

2013-07-15T21:30:37.997295-04:00 host=co052065, relayHost=co052065,
tag=rhnsd[12171]:, programName=rhnsd, procid=12171, facility=daemon,
sev=debug, appName=rhnsd, msg= running program /usr/sbin/rhn_check

```

3. Associate the scalaLogFormat template with the log files to be ingested by IBM Operations Analytics - Log Analysis. It will log all log entries to *<filename>* in addition to any other associations in the configuration file. For example:

```

*.* /var/log/<filename>.log;scalaLogFormat

```

4. Restart the rsyslog daemon.

For more information about restarting the rsyslog daemon, see <http://rsyslog.com/doc>.

5. Ensure that the output file created for IBM Operations Analytics - Log Analysis can be read by your IBM Operations Analytics - Log Analysis user, and write that file directly to the monitored logsource directories.

For example:

```

*.* <HOME>/logsources/SyslogInsightPack SCALA.log;scalaLogFormat

```

For more information about rsyslog configuration files, see: http://www.rsyslog.com/doc/rsyslog_conf.html

Configuration artifacts

The following table lists the configuration artifacts that are provided with the Insight Pack for each log file.

Table 65. Insight Pack configuration artifacts	
Artifact	Name for the syslog log
Splitter rule set	Syslog-Split
Annotator rule set	Syslog-Annotate
Source type	Syslog_custom
Collection	Syslog-Collection1

Table 66. Data sources	
Artifact	Name for the syslog log
Splitter file set	DefaultSyslogSplitter
Annotator file set	DefaultSyslogAnnotator
Source type	Syslog_default

Note: Data sources are not predefined. A user with administrator privileges must define at least one syslog data source type and collection before the application can be used.

Log File Agent configuration

The supported log files share IBM Tivoli Monitoring Log File Agent (LFA) configuration files. The following LFA configuration files are in the <HOME>/IBM-LFA-6.30/config/lo directory (where <HOME> is the installation location of IBM Operations Analytics - Log Analysis):

- SyslogInsightPack-lfasyslog.conf: Configuration file for the syslog log file agent.
- SyslogInsightPack-lfasyslog.fmt: Matches records for the syslog log files.

Splitting and annotation AQL modules

Splitting and annotation are handled by the following Annotation Query Language (AQL) modules. Syslog_custom uses the AQL-based splitters and annotators.

Table 67. Insight Pack AQL modules	
AQL Module	Description
common	Common code module that is used across multiple insight packs (for example, to recognize time stamp formats).
dateTimeSplitter newlineSplitter	Splitter modules for syslog log files.
annotatorSyslog	Annotator module for syslog log files.

Log file splitters

The splitters that are provided with the Insight Pack are described here as a reference for users.

The Insight Pack supports the ISO 8061 time stamp, yyyy-mm-ddTHH:mm:ss.SSSSSSX where X is the GMT offset. Each log record begins with an ISO-formatted time stamp and is split across time stamp boundaries. An example of the ISO-formatted time stamp generated by rsyslog is:

```
2013-06-26T12:21:29.471400-04:00
```

The IBM Operations Analytics - Log Analysis index function is limited to milliseconds in the date format. The Syslog Insight Pack annotates the time stamp and rounds up the microseconds. The sample ISO-formatted time stamp is indexed with the following format for the index configuration:

```
yyyy-mm-ddTHH:mm:ss.SSSX
```

and rendered in the IBM Operations Analytics - Log Analysis search UI as:

```
06/26/2013 16:21:29.471-04:00
```

The Syslog_default source type, which configures /var/log/messages files, supports mmm dd hh:MI:ss and yyyy-mm-ddThh:Mi:ss SSSSSS Z time formats. There is no year in the mmm dd hh:Mi:ss time format. To meet the required date format for IBM Operations Analytics - Log Analysis, the system year value is used as the year during data ingestion. The IBM Operations Analytics - Log Analysis index function is limited to milliseconds in the date format, therefore the yyyy-mm-ddThh:Mi:ss SSSSSS Z time format is reduced to yyyy-mm-ddThh:Mi:ss SSS Z.

Log file annotations

The annotations that are defined by the log file index configurations are described here.

The index configuration file is included in the Insight Pack in the sourcetypes.json file (found at <path>/SyslogInsightPack_<version>/metadata), where <path> is the path where you saved the Syslog Insight Pack.

You can customize the artifacts in the index configuration file by creating another source type and modifying a copy of the Syslog index configuration.

The following sections describe the fields that are defined in the index configuration file. These fields, or annotations, are displayed in the IBM Operations Analytics - Log Analysis Search workspace, and can be used to filter or search the log records. Fields are extracted from the fields of a log record or collected from metadata around the log file. Each table gives the names of the fields (these names correspond to fields in the IBM Operations Analytics - Log Analysis Search workspace), descriptions of how the related annotations are made, and the index configuration attributes assigned to the fields.

Log record annotations

The following table lists the index configuration fields that relate to log record annotations. Each field corresponds to part of a syslog log record.

Table 68. Log record index configuration fields		
Field	Description	Attributes
syslogHostname	The hostname from the message.	dataType = TEXT retrievable = true retrieveByDefault = true sortable = true filterable = true searchable = true

Table 68. Log record index configuration fields (continued)		
Field	Description	Attributes
syslogRelayHostname	The hostname of the system the message was received from (in a relay chain, this is the system immediately in front, and not necessarily the original sender).	dataType = TEXT retrievable = true retrieveByDefault = true sortable = true filterable = true searchable = true
tag	The TAG from the message.	dataType = TEXT retrievable = true retrieveByDefault = true sortable = true filterable = false searchable = true
programName	The static part of the tag as defined by BSD syslogd. For example, when TAG is "named[12345] ", programName is "named".	dataType = TEXT retrievable = true retrieveByDefault = true sortable = true filterable = true searchable = true
processID	The contents of the PROCID field.	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = true searchable = true
facility	The facility from the message.	dataType = TEXT retrievable = true retrieveByDefault = true sortable = true filterable = true searchable = true
severity	The severity from the message (in text form).	dataType = TEXT retrievable = true retrieveByDefault = true sortable = true filterable = true searchable = true
syslogAppName	The APP-NAME from the message.	dataType = TEXT retrievable = true retrieveByDefault = true sortable = true filterable = true searchable = true
message	The MSG (the message) in the log record.	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true

Metadata annotations

The following table lists the index configuration fields that relate to metadata annotations.

Table 69. Metadata annotation index configuration fields		
Field	Description	Annotation attributes
datasourceHostname	The host name that is specified in the data source.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: true searchable: true
timestamp	The timestamp from the log record.	dataType = DATE retrievable = true retrieveByDefault = true sortable = true filterable = true searchable = true
application	The application name that is populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
middleware	The middleware name that is populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
hostname	The host name that is populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: true filterable: true searchable: true
service	The service name that is populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
logRecord	The entire log record output by the splitter.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true

Log file considerations

This section covers log rotation and the streaming of logs to a centralized server.

Log rotation

Linux provides `logrotate` to configure rotation. The global options are specified in `/etc/logrotate.conf`. Options for specific files (which can override the global options) are in `/etc/logrotate.d` for each log file.

Note: For more information on the `logrotate` command, look up the UNIX command documentation online.

When the logs are rotated, the log file is renamed with a `.1` extension (assuming the `dateext` option is not included) and truncated to zero length. The rotation configuration also determines how often old logs are archived, that is, old `*.n` are removed or archived. The log locations are defined in `/etc/rsyslog.conf` (which is by default `/var/log`).

The recommended method for loading data is to use the IBM Tivoli Monitoring Log File Agent (LFA) installed on the remote system where `rsyslogd` is executing to push data, or use LFA installed where IBM Operations Analytics - Log Analysis is installed to pull data. Create `.conf` and `.fmt` files specific to each log file, the destination of which is specified in `/etc/rsyslog.conf`. The data source definition should specify `<filename>.1`. This ensures all log records are processed. However, they will only be sent when the rotation occurs. The user can configure rotation to occur more frequently to minimize the time lag from the current log. Alternatively, the user can monitor the `<filename>`. When you monitor `<filename>`, there is a window where log entries will not be forwarded if log entries are rotated before LFA has polled the data. If the user has configured daily or hourly rotation, they can monitor the `*.1` file name to avoid a window where log entries are not forwarded.

The best practice is to rotate the logs frequently so `<filename>.1` has recent data for IBM Operations Analytics - Log Analysis to ingest. The default log rotation is weekly. You can change the rotation for `syslog` in `/etc/logrotate.d/syslog`. To change it to daily, add the **daily** option in `/etc/logrotate.d/syslog` configuration file. If the logs are large, you can rotate them based on a size with the **size** option.

The following two sections describe configuration changes to `SyslogInsightPack-lfasyslog.conf` and `SyslogInsightPack-lfasyslog.fmt` only when dealing with rotating log files.

Logfile Agent Configuration - SyslogInsightPack-lfasyslog.conf

The following parameters should be specified to monitor the rotating files:

```
LogSources=<syslog directory to monitor>/<logfile name>.*
FileComparisonMode=CompareByAllMatches
```

Logfile Agent Configuration - SyslogInsightPack-lfasyslog.fmt

Use the following specification to avoid defining multiple data sources because of the file name changes when the log rotates. This allows a fixed file name in the log path specification.

```
// Matches records for any Log file:
//
REGEX AllRecords
(.*)
hostname LABEL
-file <logfile name>.log
RemoteHost DEFAULT
logpath PRINTF("%s",file)
text $1
END
```

Centralized logging

If you are streaming logs to a central server, the best practice is to stream to one consolidated log for ingestion by IBM Operations Analytics - Log Analysis. The same best practices are applicable to the consolidated file as for the logs in the non-server scenario. The logs should be rotated frequently so `<filename>.1` has recent data, and the Log File Agent should be used to pull or push the data to the IBM Operations Analytics - Log Analysis server.

To configure `rsyslog` to stream logs to a central server (for example, `192.168.1.1`), do the following:

1. Add the following to each client (or edge systems) `/etc/rsyslog.conf` file to stream to the central server:

```
$ModLoad imuxsock
$ModLoad imklog
```

```
# Provides UDP forwarding. The IP is the server's IP address
*. * @192.168.1.1:514
# Provides TCP forwarding. But the current server runs on UDP
# *. * @@192.168.1.1:514
```

2. On the central server (for example, with IP address 192.168.1.1) add the following to `rsyslog.conf`:

```
$ModLoad imuxsock

# provides kernel logging support (previously done by rklogd)
$ModLoad imklog

# Select the syslog reception of UDP or TCP. For TCP, load imtcp by
uncommenting $ModLoad imtcp.
#$ModLoad imudp
#$ModLoad imtcp

# Select the syslog reception port. For TCP, uncomment InputServerRun 514
#$UDPServerRun 514
#$InputTCPServerRun 514
# This FILENAME template generates the log filename dynamically.
# You can replace the specification with variables applicable to
# your site.
# The scalaLogFormat template formats the message required
# for ingestion by SCALA.
$template FILENAME, "/var/log/scala-syslog.log"
$template scalaLogFormat, "%TIMESTAMP:::date-rfc3339% host=%HOSTNAME%,
relayHost=%FROMHOST%, tag=%syslogtag%, programName=%programname%,
procid=%P ROCID%, facility=%syslogfacility-text%,
sev=%syslogseverity-text%, appName=%APP-NAM E%, msg=%msg%\n"

# Log all messages to the dynamically formed file.
*. * ?FILENAME;scalaLogFormat
```

3. Decide whether you are going to use either the UDP or TCP configuration and comment out the other. For example, to use TCP update the code section as follows:

```
# Select the syslog reception of UDP or TCP. For TCP, load imtcp
# by uncommenting $ModLoad imtcp.
#$ModLoad imudp
$ModLoad imtcp

# Select the syslog reception port. For TCP, uncomment
# InputServerRun 514
#$UDPServerRun 514
$InputTCPServerRun 514
```

Web Access Logs Insight Pack

The Web Access Logs Insight Pack provides the capability to ingest and perform metadata searches against Web Access Logs (Apache IHS, JBoss, Apache Tomcat) in IBM Operations Analytics - Log Analysis.

This document describes the version of the Web Access Logs Insight Pack that is installed when you install IBM Operations Analytics - Log Analysis. An updated version of the Web Access Logs Insight Pack may have been published after this version of IBM Operations Analytics - Log Analysis. To download the latest versions of this Insight Pack as well as updated documentation, see <http://www.ibm.com/developerworks/servicemanagement/downloads.html>.

The Web Access Log Insight Pack ingests records in the web server access log. Server access logs record information about all requests handled by a web server. This can include information about the IP address of the client making the request, userid of the person making the request (determined by the HTTP authentication), timestamp when the request was received, the request line, etc. The access log is highly configurable, and the LogFormat or pattern is used to define the contents and format of the access log.

Note: The access log is different from the web server log, `server.log`.

By using the LogFormat directive or pattern to create a delimiter-separated value (DSV) access log file, the Web Access Log Insight Pack can annotate and index access logs for ingestion, annotation, and

indexing into IBM Operations Analytics - Log Analysis. The Web Access Logs Insight Pack supports the following web servers (and any others which enable the LogFormat specification required by this insight pack):

- Apache/IBM HTTP Server 8.5.5.0
- Apache Tomcat 7.0.42, 6.0.37
- JBoss v7

The Web Access Logs Insight Pack can be installed with IBM Operations Analytics - Log Analysis 1.1.0.2 and higher.

Installing the Web Access Logs Insight Pack

Instructions on how to install the Web Access Logs Insight Pack

Before you begin

The prerequisites of the Web Access Logs Insight Pack are:

- IBM Operations Analytics - Log Analysis v1.1.0.2
- DSV Toolkit v1.1.0.1 or higher

Note: The DSV is only needed if you are generating a new insight pack to support other access log formats.

About this task

The Web Access Logs Insight Pack is installed using the `pkg_mgmt` utility.

Procedure

1. Upload the Web Access Logs Insight Pack archive file, `WebAccessLogInsightPack_<version>.zip`, to the system where IBM Operations Analytics - Log Analysis is installed.
2. Install the Web Access Logs Insight Pack using the `pkg_mgmt.sh` command:

```
<HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -install  
<path>/WebAccessLogInsightPack_<version>.zip
```

Where `<path>` is the path where you saved the Web Access Logs Insight Pack.

3. (Optional) If you are using the Log File Agent to load the data into IBM Operations Analytics - Log Analysis, deploy the log file agent configuration files with the following command:

```
<HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -deploylfa  
<path>/WebAccessLogInsightPack_<version>.zip
```

Where `<path>` is the path where you saved the Web Access Logs Insight Pack.

Configuring the Web Access Logs Insight Pack

Instructions on how to configure the Web Access Logs Insight Pack.

Procedure

1. In the IBM Operations Analytics - Log Analysis Administrative Settings workspace, create a new log source for the log file to be monitored. The source type should be `WebAccessLog`.
2. On the web server, customize the access log format to a delimiter-separated value output (DSV) that can be consumed by the Web Access Log Insight Pack and IBM Operations Analytics - Log Analysis. The syntax to customize the log format is different for each web server, but the generated log will be the same. Following is the log format directive for the supported web servers:

For Apache/IHS

- a. Edit <ServerRoot>/conf/httpd.conf file, where <ServerRoot> is the root installation path.

- 1) Add the following log format directive:

```
LogFormat "Apache/IHS,%h,%l,%u,%t,%m,\"%r\",%>s,%b,%D,\\\"%{Referer}i\\\",\\\"%{User-Agent}i\\\"" scalaAccessLog
```

- 2) Update the access log directory specification to use the LogFormat directive:

```
CustomLog logs/access_log scalaAccessLog
```

- 3) Comment out the following line by prefixing it with #:

```
CustomLog logs/access_log common
```

- b. Restart the web server.
- c. The generated access files are at <ServerRoot>/logs.

For JBoss

- a. Edit the file <JBoss_HOME>/jboss-eap-6.1/standalone/configuration/standalone.xml
- b. Find the XML element subsystem xmlns="urn:jboss:domain:web:1.4" and add the following <access_log> element:

```
<subsystem xmlns="urn:jboss:domain:web:1.4"
  default-virtual-server="default-host" native="false"
  <connector name="http" protocol="HTTP/1.1" scheme="http"
    socket-binding="http"/>
  <virtual-server name="default-host" enable-welcome-root="true">
    <alias name="localhost"/>
    <alias name="example.com"/>
    <access-log prefix="access-log." pattern="JBoss,%h,%l,%u,%t,%m,&quot;%r&quot;;%s,%b,%D,&quot;%{Referer}i&quot;;&quot;%{User-Agent}i&quot;;">
      <directory path="." relative-to="jboss.server.log.dir"/>
    </access-log>
  </virtual-server>
</subsystem>
```

- c. Restart the JBoss App Server
- d. Look for the access log file in <JBoss_HOME>/standalone/log
Where <JBoss_HOME> is the directory where you installed JBoss

For Apache Tomcat

- a. Edit the file <tomcat-dir>/conf/server.xml where <tomcat-dir> is the installation root and add the following log format:

```
<Valve className="org.apache.catalina.valves.AccessLogValve"
  directory="logs"
  prefix="localhost_access_log." suffix=".txt"
  pattern="Tomcat,%h,%l,%u,%t,%m,&quot;%r&quot;;%s,%b,%D,&quot;%{Referer}i&quot;;&quot;%{User-Agent}i&quot;;"
/>
```

- b. Restart the web server using the scripts in <tomcat-dir>/bin
- c. The log files are written on <tomcat-dir>/logs/localhost_access_log.<date>.txt
3. (Optional) Configure the Log File Agent to monitor rotated logs. This step is only required if your web server is configured to rotate log files and you are using the Log File Agent to ingest the log files.

Note: Access logs are rotated by default for Apache Tomcat and JBoss. Access Logs are not rotated by default for Apache/IHS. For instructions on how to configure log rotation for Apache/IHS, see [“Web Access Logs Best Practices”](#) on page 289.

Each web server has different syntax on how to specify rotation and the generated filename. By default, a rotated log has a timestamp or a number in the filename. Specify the log filename pattern in the `WebAccessLogInsightPack-lfadv.conf` file that is applicable to your web server.

- a. In `WebAccessLogInsightPack-lfadv.conf`, update `LogSources` to monitor all the files in the directory:

```
LogSources=<web server log directory to monitor>/
<access_log_filename_without_timestamp>*
FileComparisonMode=CompareByAllMatches
```

- b. Update `WebAccessLogInsightPack-lfadv.fmt` to specify a fixed filename so you can use the same fixed name in the path of the IBM Operations Analytics - Log Analysis logsource configuration. You only need to define one logsource with this path, and LFA will monitor all the files in the directory because you specified wildcard file naming in the `WebAccessLogInsightPack-lfadv.conf` specification.

```
// Matches records for any Log file:
// REGEX AllRecords
(.*?) hostname LABEL
-file web_server_access.log
RemoteHost DEFAULT logpath PRINTF("%s",file)
text $1
END
```

LFA will monitor all the log records in the directory (as specified by the `LogSources` value). This ensures no data will be lost as logs are rotated. However, LFA is allocating resources to monitor each file. This results in unnecessary resources since the rotated logs will not be updated again. It is a best practice to periodically archive old logs so LFA can release resources monitoring static files. For Unix, you can use tools like `logrotate` and `cron` to schedule archiving of old logs.

4. If you want to collect logs from multiple web servers, or want to ingest an archive of rotated logs, the recommended method for loading data is to use the Data Collector client.

Web Access Logs splitter rules

Splitting describes how IBM Operations Analytics - Log Analysis separates physical log file records into logical records using a logical boundary such as time stamp or a new line.

The Web Access Log Insight Pack will split log records on new line boundaries.

Web Access Logs annotation rules

After the log records are split, the logical records are sent to the annotation engine. The engine uses rules to extract important pieces of information that are sent to the indexing engine.

According to the required configuration, the format of a web access log file is:

```
<webServerType>,<clientIP>,<ident>,<auth>,<timestamp>,<verb>,
"<request>",<response>,<bytes>,<responseTime>",<referrer>",<agent>
```

For example:

```
Apache/IHS,119.63.193.107,-,-,[22/Jul/2013:18:12:37 +0200],GET,
"GET / HTTP/1.1",200,3759,324,"-",
"Baiduspider+(+http://www.baidu.jp/spider/)"
```

Where the following formatting applies:

Table 70. Web Access Logs formatting		
Field	Format String	Description
webServerType	Apache/IHS, JBoss, Tomcat	Web server that generated this log
clientIP	%h	Remote hostname or IP address

Table 70. Web Access Logs formatting (continued)		
Field	Format String	Description
ident	%l	Remote logname
auth	%u	Remote user if the request was authenticated
timestamp	%t	Time the request was received, in the common log format [dd/MMM/yyyy:HH:mm:ss Z]
verb	%m	Request method (GET, POST, etc)
request	%r	First line of request (method and request URI)
response	%s %>s (Apache/IHS)	HTTP status code of the response
bytes	%b	Bytes sent (excluding HTTP header), or "-" if 0
responseTime	%D	Time taken to process request in millis (Tomcat, JBoss) or microseconds (Apache/IHS)
referrer	%{Referer}i	Referrer on the request
agent	%{User-Agent}i	User agent on the request

If you have a mixed web server environment, and you are tracking the **responseTime** (perhaps in a Custom Search Dashboard), you may need to normalize the data. The **webServerType** can be used to know if the **responseTime** is in millisecond or microsecond.

Log File Agent configuration

You can use the IBM Tivoli Log File Agent to load data into the IBM Operations Analytics - Log Analysis.

The following Log File Agent configuration files will be installed in <HOME>/IBM-LFA-6.23/config/lo directory when **pkg_mgmt** is run with the **-deploylfa** option.

- WebAccessLogInsightPack-lfadv.conf - Configuration for the web access log file agent.
- WebAccessLogInsightPack-lfadv.fmt - Matches records for the web access log files.

Web Access Logs index configuration

To control how IBM Operations Analytics - Log Analysis indexes records from a log file, you can create indexing settings for your content Insight Pack.

The fields that are defined in the index configuration file, or annotations, are displayed in the IBM Operations Analytics - Log Analysis Search workspace, and can be used to filter or search the log records.

Fields are extracted from the fields of a log record or collected from metadata around the log file.

Table 71. Log index configuration

Field	Description	Attributes
datasourceHostname	hostname from the logsource definition	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = true searchable = true source = metadata
logRecord	complete log record text	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = false searchable = true source = annotate
webServerType	type of web server (Apache, JBoss, Tomcat)	dataType = TEXT retrievable = true retrieveByDefault = true sortable = true filterable = true searchable = true source = annotate
clientIP	remote hostname or IP address	dataType = TEXT retrievable = true retrieveByDefault = true sortable = false filterable = true searchable = true source = annotate
ident	remote logname	dataType = TEXT retrievable = true retrieveByDefault = true sortable = true filterable = true searchable = true source = annotate
auth	remote user if the request was authenticated	dataType = TEXT retrievable = true retrieveByDefault = true sortable = true filterable = true searchable = true source = annotate
timestamp	Time the request was received. The date format is: dd/MMM/yyyy:HH:mm:ss Z	dataType = DATE retrievable = true retrieveByDefault = true sortable = true filterable = true searchable = true source = annotate
verb	request method (GET, POST, etc)	dataType = TEXT retrievable = true retrieveByDefault = true sortable = true filterable = true searchable = true source = annotate

Table 71. Log index configuration (continued)

Field	Description	Attributes
request	first line of request (method and request URI)	dataType = TEXT retrievable = true retrieveByDefault = true sortable = true filterable = false searchable = true source = annotate
response	HTTP status code of the response	dataType = TEXT retrievable = true retrieveByDefault = true sortable = true filterable = true searchable = true source = annotate
bytes	bytes sent (excluding HTTP header)	dataType = LONG retrievable = true retrieveByDefault = true sortable = true filterable = false searchable = true source = annotate
responseTime	time taken to process request in milliseconds (Tomcat, JBoss) or microseconds (Apache)	dataType = LONG retrievable = true retrieveByDefault = true sortable = true filterable = true searchable = true source = annotate
referrer	referrer on the request	dataType = TEXT retrievable = true retrieveByDefault = true sortable = true filterable = false searchable = true source = annotate
agent	user agent on the request	dataType = TEXT retrievable = true retrieveByDefault = true sortable = true filterable = false searchable = true source = annotate

Web Access Logs configuration artifacts

The Web Access Logs Insight Pack configuration artifacts.

The following artifacts are created when the Insight Pack is installed:

WebAccessLog-Split

Splitter rule set.

WebAccessLog-Annotate

Annotator rule set

WebAccessLog

Source type

WebAccessLog-Collection1

Collection

Generating an alternate Web Access Log Insight Pack

Some customers may not be able to change the log format. In this situation you may need to create an insight pack that recognizes your log format, and which reuses the files provided by the Web Access Log Insight Pack.

About this task

The following steps describe how the user creates an Insight Pack re-using the files provided by the Web Access Log Insight Pack to recognize their log format. This assumes you have knowledge of the DSV toolkit (see the readme in the DSV toolkit docs directory for more information). Use the latest DSV Toolkit (at least version 1.1.0.1) available.

Procedure

1. Make a copy of the WebAccessLogModDSV.properties file, and update the copied file to contain the log fields and delimiter in your log. The file is found in <HOME>/IBM/LogAnalysis/unity_content/WebAccessLog/WebAccessLogInsightPack_v1.1.0.0/dsv
2. In the WebAccessLogModDSV.properties file, change the **aqlModule**name which is also used to generate the Insight Pack name.
For example, change **aqlModule**name to WebAccessLogMod.
3. Copy the DSV toolkit, DSVToolkit_v1.1.0.1.zip, to <HOME>/IBM/LogAnalysis/unity_content and unzip it.
4. Invoke the python script to generate a new DSV Insight Pack using the updated DSV property file:

```
python dsvGen.py
<HOME>/IBM/LogAnalysis/unity_content/WebAccessLog/WebAccessLogInsightPack_v1.1.0.0/dsv/
WebAccessLogModDSV.properties -d
```

5. Edit the generated annotation module to change the byte annotation to be the same as found in the Web Access Log annotation:

```
<HOME>/IBM/LogAnalysis/unity_content/WebAccessLog/WebAccessLogInsightPack_v1.1.0.0/dsv/
extractors/ruleset/WebAccessLog/annotations.aql
```

Replace:

```
create view bytesFinal as
  select stripQuotes(D.bytes) as bytes
  from DelimiterSplit D;
```

with the following:

```
/*-----*/
/* Tweak the AQL to annotate only bytes that appear as numeric digits */
/* Ignore bytes that are written to the log as "-" */

create view bytesBasic as
  select stripQuotes(D.bytes) as bytes
  from DelimiterSplit D;

create view bytesConsolidated as
  extract regex /(\d+)/
  on D.bytes
  return
    group 0 as match
    and group 1 as bytes
  from bytesBasic D;

create view bytesFinal as
  select GetText(D.bytes) as bytes
  from bytesConsolidated D;
/*-----*/
```

6. To restart IBM Operations Analytics - Log Analysis run the following command from the <HOME>/IBM/LogAnalysis/utilities directory:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
```

Note: Space is not a valid delimiter for the DSV Toolkit. If you use space as a delimiter, you have to update the generated annotator and splitter to recognize it.

Web Health Check Custom Search Dashboard

The Web Health Check Custom Search Dashboard visualizes data from the web access logs.

The Web Health Check Custom Search Dashboard contains the following charts:

Total Web Requests

Plots the number of web requests from the last day in a line chart.

Response Time

Plots the response times from the last day in a point chart.

Abnormal Response Code

Plots all non-200 response codes over the last day in a bubble chart.

Worst Client Response Times

Plots the maximum response times of each client IP (which is not 127.0.0.1) over the last day in a bubble chart.

Response Time SLA Violations (>100)

Plots the number of web requests where the response time is > 100 over the last day in a point chart. The query can be configured by the user to match the his SLA rules.

Note: For Apache IHS web servers, the response time is given in microseconds. For Apache Tomcat and JBoss web servers, the response times are in milliseconds.

These charts also support drill down. You can double-click on any data point in the chart to open a Search workspace that is scoped to the log records that make up that data point.

Configuring the Web Health Check Custom Search Dashboard

The available options for customizing the Web Health Check Custom Search Dashboard

About this task

By default, the Web Health Check Custom Search Dashboard displays data for the relative time interval **Last Day**. The time filter can be changed by editing the Chart Settings, and specifying the **Time Filter** in the **Query** tab.

Create a new Web Health Check Custom Search Dashboard by copying the `Web Health Check.appExmpl` in the `<HOME>/AppFramework/Apps/WebAccessLogInsightPack_v1.1.0.2` directory.

Rename the copied file as follows: `<newname>.app`.

To customize the new Custom Search Dashboard, complete following steps:

Procedure

1. The log source used by the Web Health Check Custom Search Dashboard is called `accessLog`. Change the `logsources` parameter for each chart that you want to change.

For example:

```
"logsources": [
  {
    "type": "logSource",
    "name": "/accessLog"
  },
]
```

2. The relative time interval for each chart defaults to the last day. Change the **timefilters** parameter for each chart to specify a new default relative time interval. The granularity can be specified as `second`, `minute`, `hour`, `day`, `week`, `month`, or `year`.

For example:

```
"filter":{
  "timefilters": {
    "granularity" : "day",
    "lastnum" : 1,
    "type": "relative"
  }
},
```

3. The Response Time SLA Violations charts defaults to `responseTime >100`. Change the query parameter to define a different SLA violation rule.

For example:

```
{
  "name": "ResponseTimeSLAViolations",
  "type": "FacetedSearchQuery",
  "start": 0,
  "results": 0,
  "value": {
    "filter": {
      "timefilters": {
        "granularity" : "day",
        "lastnum" : 1,
        "type": "relative"
      }
    },
    "logsources": [
      {
        "type": "logSource",
        "name": "/accessLog"
      }
    ],
    "query": "responseTime: > 100",
    "outputTimeZone": "UTC",
    "getAttributes": [
      "timestamp",
      "responseTime"
    ],
    "sortKey": [
      "-timestamp"
    ],
    "facets": {
      "timestamp": {
        "date_histogram": {
          "field": "timestamp",
          "interval": "hour",
          "outputDateFormat": "yyyy-MM-dd'T'HH:mm:ssZ",
          "outputTimeZone": "UTC"
        }
      }
    }
  }
}
```

4. After you edit the Custom Search Dashboard file, refresh the Custom Search Dashboards pane in the Search UI in order to pick up the changes.

Web Access Logs Best Practices

Best practices for integrating your web access logs with IBM Operations Analytics - Log Analysis

Note: Access logs are rotated by default for Apache Tomcat and JBoss. Access Logs are not rotated by default for Apache/IHS.

Configuring log rotation for Apache IHS

The best practice is to configure the web server so that its access logs are rotated and then monitor all rotated access logs.

rotatelog is a simple program for use in conjunction with Apache's piped logfile feature. It supports rotation based on a time interval or file size (in seconds). The filenames are `<logfile>.nnnn` or `<logfile>.<time>` dependent on the rotatelog specification. For more information on rotating logs, see <http://httpd.apache.org/docs/2.0/programs/rotatelog.html>.

Some users do not use the `rotatelog`s option because it is configured as a pipe and uses resources (that is, runs as a separate process). Another option users consider is the Unix `logrotate` tool. The filenames generated are `access.log`, `access.log.1`, `access.log.2`, and so on.

Example

An example of how to configure log rotation for Apache IHS:

1. Update the `httpd.conf` file:

a. Add the following lines to the `httpd.conf` file:

```
CustomLog "/root/IBM/HTTPServer/bin/rotatelog  
/root/IBM/HTTPServer/logs/access_log.%Y-%m-%d-%H_%M_%S 86400" scalaAccessLog
```

b. Replace `/root/IBM/HTTPServer` with whatever you are using as HTTPServer home variable

2. Update the log file agent configuration file:

a. Add the following line to the `WebAccessLogInsightPack-lfadv.conf` file:

```
LogSources=<log directory>/access_log*
```

b. If more than one file matches the pattern, add the line
`FileComparisonMode=CompareByAllMatches` so you will monitor all the files.

3. Update the `WebAccessLogInsightPack-lfadv.conf` file with the following code:

```
// Matches records for any Log file:  
//  
REGEX AllRecords  
(.*)  
hostname LABEL  
-file ihs-access-log  
RemoteHost DEFAULT  
logpath PRINTF("%s",file)  
text $1  
END
```

Web Access Logs Insight Pack References

Important references for your Web Access Logs Insight Pack.

JBoss AS v7 documentation:

<https://docs.jboss.org/author/display/AS71/Documentation>

Tomcat documentation:

<http://tomcat.apache.org/>

Apache/IHS documentation:

http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.0.0/com.ibm.websphere.ihs.doc/info/ihs/ihs/welcome_ihs.html

Note: This documentation describes IBM HTTP Server in context of the WebSphere Application Server.

WebSphere Application Server Insight Pack

A WebSphere Application Server Insight Pack is provided with IBM Operations Analytics - Log Analysis.

The Insight Pack includes support for ingesting and performing metadata searches against the following WebSphere Application Server V7 and V8 log files:

- `SystemOut.log`
- `SystemErr.log`
- `trace.log`

This document describes the version of the WebSphere Application Server Insight Pack that is installed when you install IBM Operations Analytics - Log Analysis. An updated version of the WebSphere Application Server Insight Pack may have been published after this version of IBM Operations Analytics - Log Analysis. To download the latest versions of this Insight Pack as well as updated documentation, see <http://www.ibm.com/developerworks/servicemanagement/downloads.html>.

Configuration artifacts

The Insight Pack supports the WebSphere Application Server timestamp format MM/dd/yy HH:mm:ss:SSS Z.

The timestamp format can be changed in the Data Sources workspace.

1. Open an existing WebSphere Application Server source type, then click **View Index Configuration**.
2. Select all of the text and copy it to the clipboard.
3. Create a new source type, and click **Edit Index Configuration**.
4. Paste the original index configuration into the editor, then modify the dateFormats field of the timestamp entry.
5. Fill in the remaining source type fields and click **OK**. Next, create a new collection that uses the source type you just created.

Note: Data sources are not predefined. A user with administrator privileges must define at least one WebSphere Application Server data source before the application can be used.

The following table lists the configuration artifacts that are provided with the Insight Pack for each log file.

Table 72. Insight Pack configuration artifacts			
	SystemOut.log	SystemErr.log	trace.log
Splitter	WASJavaSplitter	WASSystemErr-Split	WASTrace-Split
Annotator	WASJavaAnnotator	WASSystemErr-Annotate	WASTrace-Annotate
Source type	WASSystemOut	WASSystemErr	WASTrace
Collection	WASSystemOut-Collection1	WASSystemErr-Collection1	WASTrace-Collection1

Note: The Splitter and Annotator for SystemOut.log use Java. The Splitters and Annotators for SystemErr.log and trace.log use the Annotation Query Language (AQL).

Log File Agent configuration

The supported log files share IBM Tivoli Monitoring Log File Agent (LFA) configuration files. The following LFA configuration files are in the *Log_Analytics_install_dir/IBM-LFA-6.30/config/lo* directory:

- WASInsightPack-lfawas.conf: Configuration file for the WAS log file agent.
- WASInsightPack-lfawas.fmt: Matches records for the WAS log files.

Splitting and annotation AQL modules

Splitting and annotation are handled by the following Annotation Query Language (AQL) modules.

Table 73. Insight Pack AQL modules	
AQL Module	Description
common	Common code module that is used across most WebSphere Application Server log files (for example, to recognize timestamp formats).

Table 73. Insight Pack AQL modules (continued)	
AQL Module	Description
annotatorCommon	Common annotations module that is used for all WebSphere Application Server log files.
annotatorSystemOut	Annotator module for the trace.log file.
annotatorSystemErr	Annotator module for SystemErr.log files.
splitterWAS	Splitter module for SystemErr.log files.

Log file formats

The basic formats of the WebSphere Application Server log files are described here as a reference for users.

The basic format of SystemOut.log and SystemErr.log files is:

```
timestamp threadId shortname severity className methodName message
```

where the *className* and *methodName* fields are optional.

The basic format of trace.log files is:

```
timestamp threadId shortname severity className methodName message
parameter_1
parameter_2
parameter_n
```

where the *className*, *methodName*, and *parameter* fields are optional.

In all three log files, stack trace details are written in the following general format:

```
at packageName.exceptionClassName.exceptionMethodName(fileName:lineNumber)
```

For more information about basic and advanced WebSphere Application Server log file formats, see the following topic on the WebSphere Application Server information center:

http://www-01.ibm.com/support/knowledgecenter/SSAW57_8.5.5/com.ibm.websphere.nd.multiplatform.doc/ae/rtrb_readmsglogs.html

Log file splitter

The splitter provided with the Insight Pack is described here as a reference for users.

WebSphere Application Server splitter

The formats of the SystemOut.log, trace.log, and SystemErr.log files are similar enough that they can share a splitter. The splitter recognizes two styles of log record. Most records begin with a timestamp enclosed in brackets and the splitter uses the timestamp to define the beginning and end of each record. However, some stacktraces have a timestamp at the beginning of every line. In these cases the splitter groups the entire stack trace into one log record.

The following example shows a stacktrace with multiple timestamps. The example is split as a single log record.

```
[1/2/13 12:26:12:166 EST] 0000005d SystemErr R java.lang.NullPointerException
[1/2/13 12:26:12:166 EST] 0000005d SystemErr R at com.ibm.blah init
[1/2/13 12:26:12:166 EST] 0000005d SystemErr R at com.ibm.blah abcdefg
```

Log file annotations

The annotations that are defined by the log file index configurations are described here.

The following sections describe the fields that are defined in the index configuration file. These fields, or annotations, are displayed in the IBM Operations Analytics - Log Analysis Search workspace, and can be used to filter or search the log records. Fields are extracted from the fields of a log record or collected

from metadata around the log file. Each table gives the names of the fields (these names correspond to fields in the IBM Operations Analytics - Log Analysis Search workspace), descriptions of how the related annotations are made, and the index configuration attributes assigned to the fields.

Log record annotations

The following table lists the index configuration fields that relate to log record annotations. Each field corresponds to part of a SystemOut, SystemErr, or trace log record. The fields are listed in the order in which they appear in a log record.

Table 74. Log record index configuration fields		
Field	Description	Attributes
timestamp	The timestamp of the log record, which is located at the beginning of a line and delimited by brackets ([...]).	dataType: DATE retrievable: true retrieveByDefault: true sortable: true filterable: true searchable: true
threadID	An eight-character alphanumeric (0-9, A-F) thread identifier, which is enclosed by single white space characters, that follow a timestamp.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
shortname	A sequence of characters that represents a short name, which is enclosed by single white space characters, that follow a thread identifier.	dataType: TEXT retrievable: true retrieveByDefault: false sortable: false filterable: false searchable: false
severity	A single-character event type code or severity code (A, C, D, E, F, I, O, R, W, Z, <, >, 1, 2, 3), enclosed by single white space characters, that follow a short name.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: true filterable: true searchable: true
className	If present, a sequence of characters that represents a fully qualified class name (for example, <code>com.ibm.ws.webcontainer.servlet.ServletWrapper</code>) that follows a severity or the string "class=".	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: true searchable: true
methodName	If present, a sequence of characters that represents a method name, which is enclosed by single white space characters, that follow a class name or the string "method=".	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: false
msgclassifier	If present, a defined sequence of characters that ends with a colon (:) and that represents a message identifier (for example, WSVR0605W).	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: true searchable: true

Table 74. Log record index configuration fields (continued)		
Field	Description	Attributes
message	If present, the text of the system, error, or trace message. This field is annotated only if a msgclassifier field is present.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
javaException	If present, the Java exception names that fit the following pattern: *. *Exception	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: true searchable: true

Stack trace annotations

The following table lists the index configuration fields that relate to log record stack trace annotations.

Table 75. Stack trace index configuration fields		
Field	Description	Attributes
exceptionClassName	The class name in the top stack trace entry.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: true searchable: true
exceptionMethodName	The method name in the top stack trace entry.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
fileName	The file name in the top stack trace entry.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
lineNumber	The line number in the top stack trace entry.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
packageName	The package name in the top stack trace entry.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true

Metadata annotations

The following table lists the index configuration fields that relate to metadata annotations.

Table 76. Metadata index configuration fields		
Field	Description	Annotation attributes
application	The application name populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
hostname	The host name populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: true filterable: true searchable: true
logRecord	The entire log record output by the splitter.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
datasourceHostname	The host name specified in the data source.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: true searchable: true
middleware	The middleware name populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true
service	The service name populated by the service topology data source field.	dataType: TEXT retrievable: true retrieveByDefault: true sortable: false filterable: false searchable: true

WebSphere Application Server data loading best practice

There are different data loading recommendations based on how WebSphere Application Server logging is configured.

To set up the data loading, you need to consider:

WebSphere Application Server configuration

Is WebSphere Application Server configured to produce rolling logs or single logs?

Data loading

Determine if you want to use the IBM Tivoli Monitoring Log File Agent (LFA) installed on the remote WebSphere Application Server to push data or to use the LFA installed on the local IBM Operations Analytics - Log Analysis server to pull data. In some scenarios, you must use the Data Collector as described in scenario 2.

Logfile agent configuration

If you choose to use the LFA, use the logfile agent configuration files to specify the log files you want to monitor. The `WASInsightpack-lfawas.conf` and `WASInsightPack-lfawas.fmt` files are located in the directory:

<HOME>/IBM-LFA-6.30/config/lo

The log file scenarios here describe the specific settings for these files.

Scenario 1 - Log file rotation on one WebSphere Application Server server

WebSphere Application Server configuration

WebSphere Application Server is configured for rolling log files. For example, records are written to `SystemErr.log`. When it reaches a defined log file size it is renamed to `SystemErr_13.05.10_05.22.05.log` and a new `SystemErr.log` is created for more data. A similar flow occurs for `SystemOut.log` and `trace.log`.

Data Loading Method

The recommended method for loading data is to use the IBM Tivoli Monitoring Log File Agent (LFA) installed on the remote WebSphere Application Server server to push data or to use the LFA installed on the local IBM Operations Analytics - Log Analysis server to pull data. Create individual `.conf` and `.fmt` files that are specific to each log file. The `LogSources` definition will specify the name of the renamed file. This approach should insure that all log records are processed (no loss of log records) but with the trade off that the log records forwarded to IBM Operations Analytics will be sent only once the rollover occurs. The log forwarding will be one log file behind real-time resulting in some time lag for search results. The amount of time is dependent on the environment, that is, what the defined log size is for rollover and how frequently this occurs based on logging volumes.

Loading logs using the LFA is a CPU bound process. If your system does not meet the minimum requirements you will need to increase the `MaxEventQueueDepth`. On some systems, altering this value may produce a noticeable impact on performance. This will buffer additional LFA events while they are waiting to be processed. The required value for `MaxEventQueueDepth` may vary depending on the size of the rolled log and the number/speed of your CPU's. If you choose not to increase this value, then older events may be replaced on the event queue by newer events and not sent to the IBM Operations Analytics server.

To minimize the chance of data loss due to CPU bottlenecks, and to reduce the latency between when a log record is written to the file and when it is loaded, we recommend that the maximum size of a WebSphere Application Server log be small enough so that your system does not fall behind while processing the logs.

An alternative method is to always monitor the current log file (for example, `SystemErr.log`) and not the renamed log file. This would result in log records being forwarded immediately by the LFA. The trade off is that this configuration may result in log entries not being forwarded if those log entries were written during the LFA polling interval (sleep time) and a rollover occurs. In this case the LFA would start processing the new logfile.

Logfile Agent Configuration - `lfawas.conf` file

You must create additional `.conf` files for each log type that you monitor. For example, if you want to monitor the `SystemOut.log` and `trace.log`, then you need a `.conf` file for each log file.

Specify the following parameters to monitor the rotating `SystemErr.log` files:

```
LogSources=<was log directory to monitor>/SystemErr_*.log
FileComparisonMode=CompareByAllMatches
```

Logfile Agent Configuration - `lfawas.fmt` file

You must create additional `.fmt` files for each log type that you monitor. For example, if you want to monitor the `SystemOut.log` and `trace.log`, then you need a `.fmt` file for each log file.

Use the following `.fmt` file to specify a fixed `SystemErr.log` name and avoid the need to define multiple logsources because of the rolling log file name changes. This allows a fixed file name in the logpath.

```
// Matches records for any Log file:
//
REGEX AllRecords
(.*)
```



```
hostname LABEL
-file SystemErr.log
RemoteHost DEFAULT
logpath PRINTF("%s",file)
text $1
END
```

The same pattern can be used to define the .conf and .fmt files for the other logs:

```
<was log directory to monitor>/SystemOut_*.log OR
<was log directory to monitor>/trace_*.log</p>
```

Scenario 2 - Collecting Log Files from multiple WAS Servers

WebSphere Application Server Configuration

WebSphere Application Server is configured for single log files (non-rolling) on multiple servers and the server logs are collected for data loading on the IBM Operations Analytics - Log Analysis server.

Data Loading Method

The recommended method for loading data is to use the Data Collector client. Remove the previous log files before creating or copying new versions into the directory from which you will load data. The order of processing logs in this manner is important to handle any split records. When using the Data Collector client you need to set the `flushflag` so that split records can be merged. This is set in the `javaDatacollector.properties` file located in the `<HOME>/utilities/datacollector-client/` directory.

Configuring Insight Packs that use the LFA to stream data and Log Analysis to annotate it

To integrate the scalable data collection architecture with any Insight Packs that use the LFA to stream data and Log Analysis to annotate it, you need to adapt the configuration to make it compatible with scalable data collection.

Before you begin

Configure the scalable data collection architecture. For more information, see [“Configuring scalable data collection”](#) on page 158.

About this task

The configuration described in this topic is intended for Insight Packs that use the LFA to stream data and Log Analysis to annotate it. For example, the WebSphere Application Server Insight Pack is one such Insight Pack.

In this task, you update the Receiver cluster configuration so that data is sent from the LFA to Apache Kafka. You also update the Sender cluster configuration to pull the data from Apache Kafka and send it to Log Analysis.

Procedure

1. Create a custom data source in Log Analysis. Choose an appropriate type, for example **WASSystemOut**, and complete the other fields.

For example:

Table 77. Example data source	
Data source field	Input
Host name	PUNE_WAS
File path	SystemOut
Type	WASSystemOut
Name	PUNE_WAS_SystemOut

2. Configure the LFA.

Update the format or .fmt file to add the metadata fields for processing. For example

```
REGEX AllRecords
(.* )
hostname LABEL
-file FILENAME
RemoteHost DEFAULT
logpath PRINTF("%s",file)
type SystemOut
module WAS
site PUNE
text $1
END
```

Add the Receiver cluster or the HAProxy server and port information to the LFA's configuration or .conf file. For example:

```
ServerLocation=<HAProxy_or_receiver_cluster_server>
ServerPort=<HAProxy_or_receiver_cluster_port>
```

For more information, see [“Configuring the LFA” on page 167](#).

3. Update the Receiver cluster configuration.

You need to specify the required metadata information to facilitate the creation of topics and partitions in Apache Kafka. For more information, see [“Configuring the Receiver cluster” on page 163](#).

To update the Receiver cluster, complete these steps:

- Configure the matching pattern in the `<Logstash_install_location>/<patterns_directory>` directory where you store your patterns. This pattern matches the message and extracts the metadata fields. For example:

```
WASLFMESSAGE
<START>.*type='%{DATA:LFA_TYPE}';text='%{DATA:LFA_ORIG_MSG}'
;RemoteHost='%{DATA:LFA_REMOTE_HOST}';site='%{DATA:LFA_SITE}'
;hostname='%{DATA:LFA_HOSTNAME}';module='%{DATA:LFA_MODULE}'
;logpath='%{DATA:LFA_LOGNAME}';END
```

This pattern needs to be specified on a single line in the patterns file

- Update the Receiver cluster configuration to match the message and create the topic and partition information for Apache Kafka. For example:

```
filter {
  if [type] == "lfa" {
    grok {
      patterns_dir => "<patterns_directory>"
      match => [ "message", "%{WASLFMESSAGE}" ]
      add_tag => ["grok_lfa"]
    }
    if "grok_lfa" in [tags] {
      mutate {
        replace => [ "message", "%{LFA_ORIG_MSG}" ]
        add_field => [ "datasource", "%{LFA_SITE}_%{LFA_MODULE}_%{LFA_TYPE}" ]
        add_field => [ "resourceID", "%{LFA_HOSTNAME}_%{LFA_LOGNAME}_1" ]
      }
    }
  }
}
```

- Update the output section of the Receiver cluster configuration to send data to the Apache Kafka brokers. For example:

```
output {
  if ("grok_lfa" in [tags]) and ! ("_grokparsefailure" in [tags]) {
    kafka {
      bootstrap_servers =>
        "<Kafka_broker_server1>:<kafka_broker_port1>,... "
      topic_id => "%{datasource}"
    }
  }
}
```

```

        message_key => "%{resourceID}"
      }
    }
  }
}

```

The `datasource` field is `PUNE_WAS_SystemOut`. The `resourceID` field is composed of the host name and absolute file path, which are unique for a specific log file. The `datasource` and `resourceID` fields are mapped to topics and partitions in Apache Kafka.

4. Update the Sender cluster configuration.

- a. Update the `input` section of the Sender cluster configuration so that it can receive data that is sent from the topic in Apache Kafka. For example:

```

input {
  kafka {
    zk_connect => "<Zookeeper_host>:<Zookeeper_port>"
    group_id => "<Kafka_group_id>"
    topic_id => "<Kafka_topic_id>"
    consumer_threads => 5
    consumer_restart_on_error => true
    consumer_restart_sleep_ms => 100
  }
}

```

The `group_id` and the `topic_id` must match the values that are specified in the metadata.

- b. Update the `filter` section of the Sender configuration. Add the `host` and `path` fields to the message so that the message is mapped to the data source that is specified in Log Analysis. For example:

```

filter {
  mutate {
    add_tag => ["NO_OP"]
  }
  if "grok_lfa" in [tags] {
    mutate {
      replace => { "host" => "%{LFA_SITE}_%{LFA_MODULE}" }
      add_field => { "path" => "%{LFA_TYPE}" }
    }
  }
}

```

The `host` and `path` fields must match the **Hostname** and **File Path** that you specified when you created the custom data source in step 1.

- c. Update the output section of the Sender cluster configuration with the Log Analysis plug-in information so that it can communicate with the Log Analysis server. For more information, see [“Streaming data with logstash”](#) on page 218.

For more information, see [“Configuring the Sender cluster”](#) on page 162.

Windows OS Events Insight Pack

The Windows OS Event Insight pack allows users of IBM Operations Analytics - Log Analysis, in conjunction with the Tivoli Log File Agent or logstash, to gather and process Windows OS Events.

This document describes the version of the Windows OS Events Insight Pack that is installed when you install IBM Operations Analytics - Log Analysis. An updated version of the Windows OS Events Insight Pack may have been published after this version of IBM Operations Analytics - Log Analysis. To download the latest versions of this Insight Pack as well as updated documentation, see <http://www.ibm.com/developerworks/servicemanagement/downloads.html>.

Two separate data gathering mechanisms are supported, the Tivoli Log File Agent and logstash.

The IBM Operations Analytics - Log Analysis Windows OS Events Insight Pack is built using the IBM Operations Analytics - Log Analysis DSV Toolkit.

For Windows events gathered by the Tivoli Log File Agent (LFA) and logstash the data is configured into a comma separated format, and indexed and annotated for analysis.

The LFA is an agent that provides a configurable log file monitoring capability using regular expressions. The LFA uses the WINEVENTLOGS configuration (.conf) file option to monitor events from the Windows event log. The agent monitors a comma-separated list of event logs as shown in the following example:

```
WINEVENTLOGS=System,Security,Application
```

logstash has a supported input module named `eventlog`, <http://logstash.net/docs/1.2.2/inputs/eventlog>, which pulls events from the Windows Events Logs. The events are then forwarded using the output module available in the logstash Integration Toolkit to the IBM Operations Analytics - Log Analysis EIF Receiver.

Installing the Windows OS Events Insight Pack

If you are using IBM Operations Analytics - Log Analysis 1.2 or later, the Windows OS Events Insight Pack is installed by default, and therefore does not need to be installed separately.

About this task

The Windows OS Events Insight Pack is installed using the `pkg_mgmt` utility.

Procedure

1. Upload the Windows OS Events Insight Pack archive file, `WindowsOSEventsInsightPack_<version>.zip`, to the system where IBM Operations Analytics - Log Analysis is installed.
2. Install the Windows OS Events Insight Pack with the `pkg_mgmt.sh` command:

```
<HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -install  
<path>/WindowsOSEventsInsightPack_<version>.zip
```

Where `<path>` is the path where you saved the Windows OS Events Insight Pack.

Uninstalling the Windows OS Events Insight Pack

Instructions on how to uninstall the Windows OS Events Insight Pack.

About this task

The Windows OS Events Insight Pack is installed and uninstalled using the `pkg_mgmt` utility.

Procedure

1. Use the **`pkg_mgmt.sh`** command to determine the location of the insight pack:

```
<HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh
```

2. Uninstall the Windows OS Events Insight Pack with the **`pkg_mgmt.sh`** command:

```
<HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -uninstall  
<path>/WindowsOSEventsInsightPack_<version>
```

Where `<path>` is the path listed by the command in step 1 showing the location of Windows OS Events Insight Pack.

Performance considerations

Ensure that you consider these limitations when using the Windows OS Events Insight Pack:

Files with long log records require adjustments to the Java stack size so that they can be ingested. This adjustment is made by adding or updating the line `-Xss<stacksize>` in the `jvm.options` file located in the `<HOME>/wlp/usr/servers/Unity` directory. `<stacksize>` is the desired stack size. By default, lines of approximately 1000 characters are supported. To support lines up to 10,000 characters, the

stack size must be set to 6144 kb. To support lines up to 9,000 characters, the stack size must be set to 5120 kb. An example line is:

```
-Xss6144k
```

If you add or update the value of this parameter within the `jvm.options` file, you must restart the IBM Operations Analytics - Log Analysis system. For more information on how to restart IBM Operations Analytics - Log Analysis, see the *unity command* topic in the documentation.

Integrating the Windows OS Events Insight Pack with the Log File Agent

Configuring a Log File Agent instance on Windows allows Windows OS events to be forwarded to IBM Operations Analytics - Log Analysis.

Before you begin

Ensure that the Tivoli Log File Agent (LFA) is installed on the Windows server that is being monitored. For more information on installing the Tivoli LFA, see the "Tivoli Log File Agent User's Guide" in the [IBM Tivoli Monitoring Knowledge Center](#).

Ensure that the Windows Server can communicate with the IBM Operations Analytics - Log Analysis server. Communication is directed to the EIF receiver port on the IBM Operations Analytics - Log Analysis server (default 5529). Ensure that any firewall restrictions are lifted.

About this task

The steps in this task outline how to use the LFA to gather and push Windows OS events to IBM Operations Analytics - Log Analysis server. The LFA can be configured to send Windows OS Events to the EIF Receiver that is deployed with IBM Operations Analytics - Log Analysis. For more details on configuring the EIF Receiver on IBM Operations Analytics - Log Analysis, see section "Configuring the EIF Receiver" in the IBM Operations Analytics - Log Analysis [Knowledge Center](#).

Procedure

1. On the IBM Operations Analytics - Log Analysis server, copy the LFA `.conf` and `.fmt` files to the target Windows Server.

The `.conf` and `.fmt` files are in the directory that Windows OS Events Insight Pack is installed in.

The location of the Windows OS Events Insight Pack can be determined by using the `pkg_mgmt.sh` command:

```
<HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -list
```

2. On the target Windows Server place, both files in a directory accessible to the installation of the Tivoli LFA.
3. Edit the `lfaWinEvt.conf` file.
 - a) Update the **ServerLocation** to the host name or IP address of the IBM Operations Analytics - Log Analysis server
 - b) Update the **ServerPort** to the configured value on the IBM Operations Analytics - Log Analysis server.

The default port is 5529.

```
# Our EIF receiver host and port.
# Only needed when sending events directly to OMNIBus or TEC via EIF.
# That is configured through either the Manage Tivoli Enterprise Monitoring
# Services GUI or the
# "itmcmd config -A lo" command.
ServerLocation=unityserver.ibm.com
ServerPort=5529
```

For more information on configuring the EIF Receiver on IBM Operations Analytics - Log Analysis, see section "Configuring the EIF Receiver" in the IBM Operations Analytics - Log Analysis Knowledge Center.

The `lfaWinEvt.fmt` file formats the Windows OS events that are read by the Tivoli LFA into a CSV format for ingestion by the Windows OS Events Insight Pack.

4. The only value within this `.fmt` file you are recommended to edit is **logpath**. This string must match that of the configured data source on the IBM Operations Analytics - Log Analysis server.

By default, the value of the host name is the value that is returned by executing the DOS command **hostname** from the command line. This string must be used as the host name value when configuring the data source on the IBM Operations Analytics - Log Analysis server.

5. Launch the **Manage Tivoli Enterprise Monitoring service** application on the Windows Server.
6. Select the **Tivoli Log File Agent** template and select **Actions > Configure** using defaults.
7. Enter a unique instance name when prompted.

Note: There is a limit on the length of the instance names. The internal identification of an LFA instance by ITM libraries restricts the length to 32 chars in total.

8. In the **Log File Adapter Configuration** tab, enter the location of the `.conf` and `.fmt` files, and set the **Send ITM Event** option to **No**.

The LFA instance will now be configured and can be started from the **Manage Tivoli Enterprise Monitoring service**.

Once started, it is possible to troubleshoot the LFA instance by:

- a. Select and right-click the LFA instance in the **Manage Tivoli Enterprise Monitoring service** dialog.
- b. Click **Advanced > View Trace File**.

The `$UNITY_HOME/logs/UnityEifReceiver.log` file on IBM Operations Analytics - Log Analysis server can now be used to observe events being received from the LFA by IBM Operations Analytics - Log Analysis.

For more information on logging the `UnityEifReceiver`, see section "Enabling console logging and changing the log level for the EIF receiver" in the IBM Operations Analytics - Log Analysis Knowledge Center.

Note: When configuring the LFA, ensure that the **No TEMS** option is selected. For more details on configuring this option, see the known issue "Log File Agent fails to post events" in the IBM Operations Analytics - Log Analysis Knowledge Center.

Integrating the Windows OS Events Insight Pack with logstash

You can configure logstash on Windows to send Windows OS events to Log Analysis.

Before you begin

Ensure that the logstash is deployed on the Windows Server being monitored. For more information, see [Installing logstash on Windows based servers](#).

Ensure that the Windows Server can communicate with the Log Analysis server. Communication is directed to the REST interface port on the Log Analysis server (default 9987). Ensure that any firewall restrictions are lifted.

The `logstash-scala.conf` file is in the directory that Windows OS Events Insight Pack is installed in. The location of the Windows OS Events Insight Pack can be determined by using the `pkg_mgmt.sh` command:

```
<HOME>/IBM/LogAnalysis/utilities/pkg_mgmt.sh -list
```

About this task

The steps in this task outline how to configure logstash to send Windows OS Events to the REST interface that is part of Log Analysis.

Procedure

1. On the target Windows Server, stop logstash .
2. Make a backup of the <logstash Location>\logstash\config\logstash-scala.conf file.
3. Edit the logstash-scala.conf file. You must add the required values. For more information, see [logstash configuration file reference](#) .
4. On the IBM Operations Analytics - Log Analysis server, copy the logstash-scala.conf file to the target Windows Server.
5. On the Windows Server, place the logstash-scala.conf file in the <logstash_install>\logstash\config directory.
This overwrites the existing version.
6. On the Windows server, ensure that logstash REST output module is configured to send data to the IBM Operations Analytics - Log Analysis server.
7. On the Windows server, check that the values of the output module in the new logstash-scala.conf file match that of the backed up copy. This check is needed if you specify a non-standard location for the REST interface output module.
8. Start logstash on the Windows server.

Windows OS event format generated by the Tivoli Log File Agent

Windows OS events are formatted by the Tivoli Log File Agent into a csv format.

The value of the log source's logpath must match that specified for the logpath in the .fmt file deployed on the LFA on the Windows server.

The Windows OS Events Insight pack has been built using the IBM Operations Analytics - Log Analysis DSV toolkit. Events are formatted by the Tivoli Log File Agent into a csv format with the following columns.

Table 78. Log file format		
Number	Column Name	Description
1	EventCategory	Describes the subsystem of event, for example, EventLog:Application or EventLog:Security
2	Timetsamp	Time of event
3	Level	Information, Warning, Error etc
4	User	If a user name is associated with the event
5	EventSource	Source of event
6	Keywords	Events may have keywords associated upon generation.
7	EventID	Event ID
8	Description	Text description of event

Windows OS event format generated by logstash

The basic format of the Windows Event Log generated by logstash is described here as a reference for users.

The Windows OS Events Insight pack has been built using the IBM Operations Analytics - Log Analysis DSV toolkit. Events are formatted by logstash into a csv format with the following columns.

Table 79. Log file format		
Number	Column Name	Description
1	EventLog	Describes the subsystem of event, for example Application or Security
2	Timetsamp	Time of event
3	Level	Information, Warning, Error etc
4	User	If a user name is associated with the event
5	EventSource	Source of event
6	EventID	Event ID
7	Description	Text description of event
8	Hostname	Hostname of the Windows machine
9	EventRecordNumber	Unique event ID
10	Category	Numeric category

Windows OS Events Insight Pack App

A single sample app is provided with the Windows OS Events Insight Pack. This app queries the data gathered by the application and generates four charts.

The four charts generated by the application are:

- Event Level counts per hour over the past 24 hours
- Event Log counts per hour over the past 24 hours
- Event Source Counts per hour over the past 24 hours
- Event Level per Event Source over the past 24 hours

By default, the app will query a logsource named **WindowsOEventsLFA**.

If you want to run the chart on another logsource based on the **WindowsOEventsLFA** source type:

1. Open the file: <HOME>/AppFramework/Apps/WindowsOEventsInsightPack_<version>/WindowsEvents.app.
2. Update the value of the logsource name from **WindowsOEventsLFA** to whatever logsource name is required.

Note: All configured logsource names can be seen on the IBM Operations Analytics - Log Analysis Administrative settings UI under the **Log Sources** tab.

Limiting the flow of events to the Windows OS Event Log Insight Pack

An optional set of steps the purpose of which is to limit the events flowing to the Windows OS Event Log Insight Pack.

About this task

Windows OS generates a large number Information level logs that users may not wish to track. The .fmt file can be edited to limit what is monitored.

Procedure

1. On the Windows server edit the fmt file (See steps above for configuring the tivoli LFA) as follows.

For more information about how to configure the IBM Tivoli Monitoring Log File Agent, see [“Integrating the Windows OS Events Insight Pack with the Log File Agent” on page 301.](#)

Update the .fmt file from:

```
// Matches records for any Log file and convert to csv format:
//
REGEX AllRecords
^([A-Z][a-z]{2} [0-9]{1,2} [0-9]{1,2}:[0-9]{2}:[0-9]{2} [0-9]{4})
[0-9] (\S+) (\S+) (\S+) (\S+) ([0-9]+) (.*)
hostname LABEL
-file FILENAME
RemoteHost DEFAULT
logpath "WindowsOSEventsLFA"
text PRINTF("%s,%s,%s,%s,%s,%s,%s,%s",file,$2,$3,$4,$5,$6,$7,$8)
END
```

To:

```
// Matches records for any Log file and convert to csv format:
//
REGEX AllRecords
^([A-Z][a-z]{2} [0-9]{1,2} [0-9]{1,2}:[0-9]{2}:[0-9]{2} [0-9]{4})
[0-9] (Warning|Error|Critical) (\S+) (\S+) (\S+) ([0-9]+) (.*)
hostname LABEL
-file FILENAME
RemoteHost DEFAULT
logpath "WindowsOSEventsLFA"
text PRINTF("%s,%s,%s,%s,%s,%s,%s,%s",file,$2,$3,$4,$5,$6,$7,$8)
END
```

This will limit the events being sent to IBM Operations Analytics - Log Analysis to those of type Warning or Error or Critical. No 'Information' events will be sent to IBM Operations Analytics - Log Analysis.

2. Restart the LFA instance using the **Manage Tivoli Enterprise Monitoring service** application

Custom Insight Packs

You can use custom Insight Packs to implement customized indexing and annotating.

The process for creating custom Insight Packs is described in the Extension Guide and it is intended for advanced users.

For more information, see the *Extending* guide.

Related concepts

[Creating custom Insight Packs](#)

Chapter 8. Administrating

Read this section to understand how to use Log Analysis to administrate the data model and other aspects of Log Analysis.

Getting started with Log Analysis

Complete these tasks to help you to get started with Log Analysis.

Logging in to IBM Operations Analytics - Log Analysis

This topic outlines how to log in to IBM Operations Analytics - Log Analysis and how to change the default username and password.

By default, the `unityuser` user is assigned the `UnityUser` role. Log in as this user to access the Search workspace. The `UnityAdmin` role allows you to access the Search workspace and also the administrative workspaces. This role is assigned to the `unityadmin` user. The default passwords for each of these users is the same as the username.

You can change the default passwords for each of these users by changing the value in the basic user registry. For more information, see the *Creating roles, groups, and users in the file-based user registry* section of the documentation.

To log in to the IBM Operations Analytics - Log Analysis administrative workspaces:

1. In a web browser, type the address: `http://ip_address:9988/Unity/Admin`. If you use SSL communication, the address is `https://ip_address:9987/Unity`.
2. When prompted, type the username and password for a user with administrator access permissions and click **Go**.

To log in to the IBM Operations Analytics - Log Analysis Search workspace:

1. In a web browser, type the address: `http://ip_address:9988/Unity`. If you use SSL communication, the address is `https://ip_address:9987/Unity`.
2. When prompted, type the username and password for a user with administrator or user access permissions and click **Go**.

Note: If the 9988 or 9987 port is blocked by a firewall, IBM Operations Analytics - Log Analysis might not display. Check the firewall status and unblock the port where necessary.

Note: If you are running IBM Operations Analytics - Log Analysis over a slow network, a warning might be displayed indicating that a script is unresponsive script. To proceed, click **Continue**.

To log in to IBM Operations Analytics - Log Analysis on a Dynamic Host Configuration Protocol (DHCP) server, use the Fully Qualified Domain Name (FQDN) or the host name to log in. You cannot use the IP address as you would for non-DHCP servers. For more information, see the *Cannot display Custom Search Dashboards on Dynamic Host Configuration Protocol (DHCP) server* topic in the *Troubleshooting IBM Operations Analytics - Log Analysis* guide.

Installing sample files

This topic outlines how to load sample artifacts and data so that you can get started using IBM Operations Analytics - Log Analysis quickly and to allow you to understand how you can apply IBM Operations Analytics - Log Analysis to your own environment.

About this task

When you load the sample data, sample IBM Operations Analytics - Log Analysis artifacts, sample data, and sample Custom Search Dashboards are loaded. In addition, a number of sample Custom Search Dashboards are loaded:

- **sample-Web-App:** This Custom Search Dashboard displays a sample dashboard of a typical web application built using web servers, an application server, and a database.
- **sample-WAS-Troubleshooting:** This Custom Search Dashboard displays a sample dashboard for WebSphere Application Server SystemOut logs from multiple servers.
- **sample-events-hotspots:** This Custom Search Dashboard displays a sample event analysis dashboard built for sample IBM Tivoli Netcool/OMNIBus events.

Note: If you enable Lightweight Directory Access Protocol (LDAP) authentication and you want to load the sample data, you must create a user called unityadmin that is assigned to a group called unityadmin. If you do not, you cannot load the sample data.

Procedure

1. Log into the Search workspace: `https://<ipaddress>:<port>/Unity` where *<port>* is the port specified during installation for use by the web console. The default value is 9987. The default administrative username and password are unityadmin and unityadmin respectively.
2. On the **Getting Started** page, click **Install Sample Data > Start Now**. The sample data loads.

Enabling the GUI search history

You can enable history for the search on the IBM Operations Analytics - Log Analysis UI.

About this task

IBM Operations Analytics - Log Analysis uses a cookie that is saved in the temporary directory of the browser to remember the last 10 terms that are entered in the search field.

To view the last 10 search results, select the search list. To add a term to the list of searches, enter the term in the search field and click **Search**.

The cookie that saves the search terms expires every 30 days by default.

Procedure

To enable search history for the IBM Operations Analytics - Log Analysis UI, you must ensure that your browser settings are configured to save cookies.

To clear your search history, delete the saved cookies for your browser.

Defining a default search

If you want to define a default initial search that is displayed when you log into the Search workspace, complete the steps in this topic.

About this task

You can define the search query. However, any additional parameters such as the Data Source or Time filter cannot be defined for your default search. To define your search:

Procedure

1. Open the `unitysetup.properties` file located in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF` directory.
2. Locate and, if necessary, remove the comment notation from the line:

```
SEARCH_QUERY_FOR_DEFAULT_SEARCH=*
```

3. Add the text that you want to define as your default query: For example:

```
SEARCH_QUERY_FOR_DEFAULT_SEARCH=TUNE9001W
```

4. Save the file.

5. Use the following command to restart IBM Operations Analytics - Log Analysis:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
```

Results

If there are search results in the last 15 minutes for the specified query, the results of the query are displayed. If you have defined a value for the search and no results are found, a message is displayed indicating that no matches were found.

Creating and updating the data model

Before you load data into Log Analysis, you must create the data sources, source types, index configuration, and physical source IDs required to model your data ingestion, indexing, annotation, and grouping of data.

You also need to add and delete objects from your model.

Data Sources workspace

The Data Sources workspace allows you to define your data sources. This information is used to process the content of the log file and facilitate search by converting the file contents into a structured format.

Data Source creation

You create data sources to ingest data from a specific source.

Before you can search a log file, you must use the data source creation wizard to create a data source. To start the wizard, open the Admin UI, click **Add > Data Sources**. The wizard contains text to help you to complete the task.

You can create three types of data source. Select one of the following types:

- To stream data from a file that is stored locally on the IBM Operations Analytics - Log Analysis server, create a local data source. The IBM Tivoli Monitoring Log File Agent on the same server is automatically configured to retrieve local data.
- To stream data from a file that is stored on a remote server, create a remote data source. Enter a host name, user name, and password. The IBM Tivoli Monitoring Log File Agent on the same server is automatically configured to pull data from a remote server.

Note: If the remote server uses the Windows operating system, this setting requires that the ssh daemon is configured on the remote server.

- To stream data from a file that is stored on a remote IBM Operations Analytics - Log Analysis server where data delivery is manually configured, or where data ingestion is configured for Logstash or the Data Collector client, create a custom data source. Enter a host name. There is no automatic configuration or verification.

If you select local or remote as the location, IBM Operations Analytics - Log Analysis creates a new collection automatically. If you select custom as the location, you can choose to associate it with an existing collection or you can create a new collection.

The log file specified for the **File path** field is automatically ingested into IBM Operations Analytics - Log Analysis for data sources with **Local file** or **Remote file** configuration options.

If you want to ingest rolling files, click the **Rolling file pattern** check box and enter the rolling file pattern. For example, for a WebSphere Application Server log file the base line file is `SystemOut.log`. The rolling file pattern is `SystemOut_*.log`. If you use this setting, the data latency depends on how frequently the files are rolled in the source application.

To complete the process, enter a name for the data source.

You can also edit an existing data source.

Logical and physical data sources

A logical data source is a data source that you create in Log Analysis. A physical data source is source of data in your environment. For example a directory where log files are stored.

You can use Log Analysis to create a logical data source which you can use to stream data from multiple, physical sources of data. For example, if you have multiple servers where you store log files, you can use a data source in Log Analysis to stream data from the servers.

For an example of how to use a logical data source to stream data from multiple physical data sources, see [“LFA example: Streaming data from a single remote host” on page 202.](#)

Editing service topology information in Group JSON

If your services are provided using a web application that is deployed across a range of servers such as web servers, application servers, and database servers, you might want to define the scope and structure of your application or service. To represent your application or service hierarchy, you can create a group or edit the default groups in the Log Analysis Group JSON file.

About this task

After you create a service topology, you can then associate your data source with the appropriate node in the service topology. The hierarchy of data sources, reflecting your service topology, is shown in the **Data Sources** lists in the Data Sources workspace and the Search workspace. When you select a data source to narrow your search criteria, that data source, and any data sources in the service topology tree are searched.

To create a service topology that meets your requirements, edit the existing default Group JSON file. The JSON file has structure with each node in the service topology that is defined by a type, a name, and a value.

Before you can view a data source in the **Data Sources** filter on the **Search** UI, you must assign the data source to a group in JSON file. If you do not assign any data sources to a group, the group is not displayed UI. If you only assign a data source to a specific leaf node of a group, only that leaf node is displayed for the group.

You cannot delete groups. Log Analysis retains the association between the group and the data source in case you want to search it. You can remove old groups from the service topology file. However, the removed groups are still visible on the UI.

Note: The `unityServiceTopology.json` file is renamed as the `Group.json` file.

A sample of the JSON file is displayed:

```
[
  {
    "type": "Service",
    "name": "Day Trader",
    "value": [
      {
        "type": "Application",
        "name": "Trading Application",
        "value": [
          {
            "type": "Middleware",
            "name": "WAS",
            "value": [
              {
                "type": "Hostname",
                "name": "nc9118041001",
                "value": ""
              },
              {
                "type": "Hostname",
                "name": "nc9118041002",
                "value": ""
              },
              {
                "type": "Hostname",
                "name": "nc9118041003",
                "value": ""
              }
            ]
          }
        ]
      }
    ]
  }
]
```

```

    "value": []
  },
  {
    "type": "Hostname",
    "name": "nc9118041005",
    "value": []
  }
],
{
}

```

To edit the JSON file:

Procedure

1. Create a backup copy of the Group.json file that is in the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/com/ibm/tivoli/loganalytics/framework directory and copy it to a secure location.
2. Using a JSON editor, open and edit the original file to reflect your service topology.
3. Save the file.
4. To ensure that your updates are reflected on the Admin UI, clear the browser cache before you log in.

Results

After you define the Group JSON file, you can then assign the group to the data source when you configure the data source. For more information about creating data sources, see [“Data Source creation”](#) on page 309.

If you assign a data source to a node in the service topology, the assignment is maintained regardless of any updates you make to service topology. Changes to the Group JSON are not reflected in data sources that were made before the update.

Data Types workspace

The Data Types workspace allows you to configure the type of log information that is consumed.

Collections

Collections allow you to group together log data from different data sources that have the same source type. For example, you might want to assign all the data sources for a WebSphere Application Server cluster into a single Collection so that you can search them as a group. This section outlines how to manage your Collections.

Adding a Collection

This topic outlines the steps you must follow to add a Collection to IBM Operations Analytics - Log Analysis.

Procedure

To add a Collection:

1. In the Data Types workspace, click **Add > Collection**. The **Add Collection** tab is displayed
2. Enter the details for the collection that you want to add:

Name

Provide a unique name for the Collection.

Source Type

From the list, select the type of log file data that you want the Collection to contain.

3. Click **OK**. A message is displayed requesting that you confirm that you want to create the Collection and indicating that you cannot edit the Source Type property after the Collection is saved. If you are satisfied that you have added the correct values to these properties, click **OK**.

Editing a Collection

After a Collection has been created, you cannot edit the properties associated with the Collection. Use the **Edit** button to review the existing properties for a Collection.

Procedure

To view a collection:

1. In the Data Types workspace, expand the **Collections** list.
2. Select the Collection that you want to view and click **Edit**. The Collection is opened in a new tab.
3. To close the Collection, click **Close**.

Deleting a Collection

You can delete an existing Collection.

Procedure

To delete a Collection:

1. In the Data Types workspace, expand the **Collections** list.
2. Select the Collection that you want to delete and click **Delete**. A message is displayed listing the data sources that are associated with the Collection. These data sources are deleted when the Collection is deleted. All data associated with the Collection is deleted when the Collection is deleted.
3. Click **Delete**.

Source Types

A Source Type defines how data of a particular type is processed by IBM Operations Analytics - Log Analysis. This determines how data is indexed, split, and annotated.

Index configuration

You can use the index configuration of a Source Type to provide important information about how data of that Source Type is indexed in IBM Operations Analytics - Log Analysis.

Index configuration is specified using JSON configuration notation. When creating a source type, you can specify a file containing JSON configuration notation, or you can enter the configuration directly to the **Index Config** field.

To index data appropriately, different types of data require different index configuration.

Editing an index configuration

Use **Edit Index Configuration** editor to edit an index configuration. You can add new fields and change the order of existing fields.

Procedure

1. Open the **Admin UI** workspace, click on a source type and click **Index Configuration**.
2. To edit the order of an existing field, click on the field and click **Edit Field Order**. When you are finished, click **Save Field Order**.
3. To add a new field, click **Add New Field**.
4. You can also specify the data type for each field. To change the data type, click on the **Data type** column and click the data type. The following data types are available:

<i>Table 80. Edit Index Configuration UI data types</i>	
Data type	Use
TEXT	Use this type if the field contains text.
DOUBLE	Use this type if the field is a double.
LONG	Use this type if the field contains a long string.

Table 80. <i>Edit Index Configuration UI data types (continued)</i>	
Data type	Use
DATE	Use this type if the field contains a date.

5. You can also specify that date and time format that you want to use. To change the format, click on the **Data type** column and click **Date**. The top ten most frequently used date formats are displayed.

If you enter a custom time format, the time format is validated against the Java Simple Date format. If you enter an invalid format, an error message is displayed.

6. To configure the indexing settings for the specified index configuration, click the relevant check box to configure the option. For example, to make the search results filterable, click the **Filterable** check box. The following options are available:

Table 81. <i>Edit Index Configuration UI check boxes</i>	
Check box	Use
Retrievable	Specify whether you want the index data to be retrieved by Log Analysis.
Retrieve by default	Specify that you want Log Analysis to retrieve all available data by default.
Sortable	Specify whether you want the Log Analysis users to be able to sort search results.
Filterable	Specify whether you want the Log Analysis users to be able to filter search results.
Searchable	Specify whether you want the Log Analysis users to be able to search results.
Combine	Specify whether Log Analysis combines only the first or all index entries.
Paths	Specify the path that is associated with the specified field.

7. To save your changes, click **Apply**.

Related concepts

[“Edit Index Configuration UI” on page 313](#)

Use the **Edit Index Configuration** to create and edit index configurations.

Edit Index Configuration UI

Use the **Edit Index Configuration** to create and edit index configurations.

The following table summarizes the buttons that are available as part of the **Edit Index Configuration UI**:



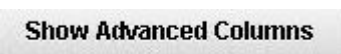

Table 82. <i>Edit Index Configuration UI buttons</i>		
Icon	Name	Use
	Add New Field	Use this button to create a new field in the index configuration.
	Edit Field Order	Use this button to edit the order of a field in the index configuration.
	Show Advanced Columns	Click this button to display the Combine and Path columns

Table 82. Edit Index Configuration UI buttons (continued)		
Icon	Name	Use
	Delete selected field	Click a field and click this button to delete the field. You cannot delete the timestamp field.

The following table summarizes the data types that are available as part of the **Edit Index Configuration** UI:

Table 83. Edit Index Configuration UI data types	
Data type	Use
TEXT	Use this type if the field contains text.
DOUBLE	Use this type if the field is a double.
LONG	Use this type if the field contains a long string.
DATE	Use this type if the field contains a date.

The following table summarizes the check boxes that are available as part of the **Edit Index Configuration** UI:

Table 84. Edit Index Configuration UI check boxes	
Check box	Use
Retrievable	Specify whether you want the index data to be retrieved by Log Analysis.
Retrieve by default	Specify that you want Log Analysis to retrieve all available data by default.
Sortable	Specify whether you want the Log Analysis users to be able to sort search results.
Filterable	Specify whether you want the Log Analysis users to be able to filter search results.
Searchable	Specify whether you want the Log Analysis users to be able to search results.
Combine	Specify whether Log Analysis combines only the first or all index entries.
Paths	Specify the path that is associated with the specified field.

Cloning source types and indexing configurations

If you want to modify an index configuration without losing the original configuration, you can clone the source type and edit the cloned index configuration

Procedure

1. Open the **Admin UI** workspace, click the source type that you want to clone and click **Edit Index Configuration**.
2. To copy the source type, click the **Clone** icon. The copied source type is displayed.
3. To edit the copied source type and index configuration, click **Edit Index Configuration**.
4. Save your changes.

Related tasks

[“Editing an index configuration” on page 312](#)

Use **Edit Index Configuration** editor to edit an index configuration. You can add new fields and change the order of existing fields.

Index configuration example: PMI data

This sample JSON index configuration can be used to index PMI data:

```
{
  "indexConfigMeta": {
    "name": "PMIConfig",
    "description": "IndexMappingforPMIdata",
    "version": "0.1",
    "lastModified": "9/10/2012"
  },
  "fields": {
    "hostname": {
      "dataType": "TEXT",
      "retrievable": true,
      "retrieveByDefault": false,
      "sortable": true,
      "filterable": true,
      "searchable": true,
      "tokenizer": "literal",
      "source": {
        "paths": ["metadata.hostname"]
      }
    },
    "sourceip": {
      "dataType": "TEXT",
      "retrievable": true,
      "retrieveByDefault": true,
      "sortable": true,
      "filterable": true,
      "searchable": true,
      "tokenizer": "literal",
      "source": {
        "paths": ["metadata.sourceip"]
      }
    },
    "logpath": {
      "dataType": "TEXT",
      "retrievable": true,
      "retrieveByDefault": false,
      "sortable": false,
      "filterable": false,
      "searchable": false,
      "tokenizer": "literal",
      "source": {
        "paths": ["metadata.logpath"]
      }
    },
    "logsource": {
      "dataType": "TEXT",
      "retrievable": true,
      "retrieveByDefault": false,
      "sortable": true,
      "filterable": true,
      "searchable": true,
      "tokenizer": "literal",
      "source": {
        "paths": ["metadata.logsource"]
      }
    },
    "timeformat": {
      "dataType": "TEXT",
      "retrievable": true,
      "retrieveByDefault": false,
      "sortable": true,
      "filterable": true,
      "searchable": true,
      "tokenizer": "literal",
      "source": {
        "paths": ["metadata.timeformat"]
      }
    },
    "description": {
      "dataType": "TEXT",
      "retrievable": true,

```

```

        "retrieveByDefault": false,
        "sortable": false,
        "filterable": false,
        "searchable": false,
        "tokenizer": "literal",
        "source": {
            "paths": ["metadata.description"]
        }
    },
    "PoolSize": {
        "dataType": "TEXT",
        "retrievable": true,
        "retrieveByDefault": true,
        "sortable": true,
        "filterable": true,
        "searchable": true,
        "tokenizer": "literal",
        "source": {
            "paths": ["metadata.poolsize"]
        }
    },
    "FreePoolSize": {
        "dataType": "TEXT",
        "retrievable": true,
        "retrieveByDefault": true,
        "sortable": true,
        "filterable": true,
        "searchable": true,
        "tokenizer": "literal",
        "source": {
            "paths": ["metadata.FreePoolSize"]
        }
    },
    "WaitingThreadCount": {
        "dataType": "TEXT",
        "retrievable": true,
        "retrieveByDefault": true,
        "sortable": true,
        "filterable": true,
        "searchable": true,
        "tokenizer": "literal",
        "source": {
            "paths": ["metadata.WaitingThreadCount"]
        }
    },
    "PercentUsed": {
        "dataType": "TEXT",
        "retrievable": true,
        "retrieveByDefault": true,
        "sortable": true,
        "filterable": true,
        "searchable": true,
        "tokenizer": "literal",
        "source": {
            "paths": ["metadata.PercentUsed"]
        }
    },
    "UseTime": {
        "dataType": "TEXT",
        "retrievable": true,
        "retrieveByDefault": true,
        "sortable": true,
        "filterable": true,
        "searchable": true,
        "tokenizer": "literal",
        "source": {
            "paths": ["metadata.UseTime"]
        }
    },
    "WaitTime": {
        "dataType": "TEXT",
        "retrievable": true,
        "retrieveByDefault": true,
        "sortable": true,
        "filterable": false,
        "searchable": false,
        "tokenizer": "literal",
        "source": {
            "paths": ["metadata.WaitTime"]
        }
    },
    "HeapSize": {

```

```

        "dataType": "TEXT",
        "retrievable": true,
        "retrieveByDefault": true,
        "sortable": true,
        "filterable": false,
        "searchable": false,
        "tokenizer": "literal",
        "source": {
            "paths": ["metadata.HeapSize"]
        }
    },
    "UsedMemory": {
        "dataType": "TEXT",
        "retrievable": true,
        "retrieveByDefault": true,
        "sortable": true,
        "filterable": false,
        "searchable": false,
        "tokenizer": "literal",
        "source": {
            "paths": ["metadata.UsedMemory"]
        }
    },
    "UpTime": {
        "dataType": "TEXT",
        "retrievable": true,
        "retrieveByDefault": true,
        "sortable": true,
        "filterable": false,
        "searchable": false,
        "tokenizer": "literal",
        "source": {
            "paths": ["metadata.UpTime"]
        }
    },
    "ProcessCpuUsage": {
        "dataType": "TEXT",
        "retrievable": true,
        "retrieveByDefault": true,
        "sortable": true,
        "filterable": true,
        "searchable": true,
        "tokenizer": "literal",
        "source": {
            "paths": ["metadata.ProcessCpuUsage"]
        }
    },
    "CPUUsageSinceLastMeasurement": {
        "dataType": "TEXT",
        "retrievable": true,
        "retrieveByDefault": true,
        "sortable": true,
        "filterable": true,
        "searchable": true,
        "tokenizer": "literal",
        "source": {
            "paths": ["metadata.CPUUsageSinceLastMeasurement"]
        }
    },
    "webcontainerthreads": {
        "dataType": "TEXT",
        "retrievable": true,
        "retrieveByDefault": true,
        "sortable": true,
        "filterable": true,
        "searchable": true,
        "tokenizer": "literal",
        "source": {
            "paths": ["metadata.webcontainerthreads"]
        }
    },
    "timestamp": {
        "dataType": "DATE",
        "retrievable": true,
        "retrieveByDefault": true,
        "sortable": true,
        "filterable": true,
        "searchable": true,
        "tokenizer": "literal",
        "source": {
            "paths": ["metadata.timestamp"],
            "dateFormats": ["dd/MM/yyyy HH:mm:ss.SSS"]
        }
    }
}

```

```

    }
  }
}

```

Index configuration example: Expert advice data

This sample JSON index configuration can be used to index expert advice data:

```

{
  "sourceTypeClass": 0,
  "name": "Test",
  "splitRuleSet": null,
  "extractRuleSet": null,
  "indexConfigMeta": {
    "name": "Test",
    "description": "Testing UVI Integration",
    "version": "0.1",
    "lastModified": "17/08/2012"
  },
  "fields": {
    "hostname": {
      "dataType": "TEXT",
      "retrievable": true,
      "retrieveByDefault": true,
      "sortable": true,
      "filterable": true,
      "searchable": true,
      "tokenizer": "literal",
      "source": {
        "paths": ["metadata.hostname"]
      }
    },
    "timestamp": {
      "dataType": "DATE",
      "dateFormat": "dd/MM/yyyy HH:mm:ss.SSS",
      "retrievable": true,
      "retrieveByDefault": true,
      "sortable": true,
      "filterable": true,
      "searchable": true,
      "tokenizer": "literal",
      "source": {
        "paths": ["metadata.timestamp"],
        "dateFormats": ["dd/MM/yyyy HH:mm:ss.SSS"],
        "combine": "FIRST"
      }
    },
    "logpath": {
      "dataType": "TEXT",
      "retrievable": true,
      "retrieveByDefault": false,
      "sortable": false,
      "filterable": false,
      "searchable": false,
      "tokenizer": "literal",
      "source": {
        "paths": ["metadata.logpath"]
      }
    },
    "regex_class": {
      "dataType": "TEXT",
      "retrievable": true,
      "retrieveByDefault": false,
      "sortable": false,
      "filterable": false,
      "searchable": false,
      "tokenizer": "literal",
      "source": {
        "paths": ["metadata.regex_class"]
      }
    },
    "Hostname": {
      "dataType": "TEXT",
      "retrievable": true,
      "retrieveByDefault": true,
      "sortable": true,
      "filterable": true,
      "searchable": true,
      "tokenizer": "literal",
      "source": {

```

```

        "paths": ["metadata.Hostname"]
      },
    },
    "service": {
      "dataType": "TEXT",
      "retrievable": true,
      "retrieveByDefault": true,
      "sortable": true,
      "filterable": true,
      "searchable": true,
      "tokenizer": "literal",
      "source": {
        "paths": ["metadata.Service"]
      }
    },
    "logsource": {
      "dataType": "TEXT",
      "retrievable": true,
      "retrieveByDefault": true,
      "sortable": true,
      "filterable": true,
      "searchable": true,
      "tokenizer": "literal",
      "source": {
        "paths": ["metadata.logsource"]
      }
    },
    "logRecord": {
      "dataType": "TEXT",
      "retrievable": true,
      "retrieveByDefault": true,
      "sortable": true,
      "filterable": true,
      "searchable": true,
      "tokenizer": "literal",
      "source": {
        "paths": ["metadata.text"]
      }
    },
    "className": {
      "dataType": "TEXT",
      "retrievable": true,
      "retrieveByDefault": false,
      "sortable": false,
      "filterable": true,
      "searchable": false,
      "tokenizer": "literal",
      "source": {
        "paths": ["annotations.ClassnameWSOutput.text"]
      }
    }
  }
}

```

Log annotator

You can use annotations to extract and identify pieces of information from unstructured text, for example, IP address, host name, and time stamp. Doing this allows you to search for specific types of information, rather than being limited to a simple text search of the log files.

You can search annotated information more efficiently. For example, you can search annotated information for log entries where the severity is E. If this information had not been annotated, you could perform only a simple text search for instances of the letter E, which might return irrelevant matches.

To identify annotations in a log file or configuration file, use rules specified in Annotation Query Language (AQL). Annotation can be configured for each data source type.

These annotator types are available:

Default

When using the default annotator type, you must specify a rule for splitting the text into different fields and a rule for extracting values from the identified fields.

Java

When using the Java annotator type, annotations are made to data sources of this type based on the annotator Java class you specify. The `CustomAnnotatorImpl.class` class in `annotator.jar` is provided for this purpose.

Script

When using the Script annotator type, annotations are made to data sources of this type based on the annotator script you specify. The `DefaultPythonAnnotator.py` script is provided for this purpose.

None

No annotations are made to data sources of this Source Type.

The Java and Script annotator types are provided as customization options for the default annotator type.

Adding a Source Type

This topic outlines the steps that you must complete to add a Source Type to IBM Operations Analytics - Log Analysis.

Procedure

To add a Source Type:

1. In the Data Types workspace, click **Add > Source Type**. The **Add Source Type** tab is displayed
2. Provide values for each of the required fields to create an appropriate Source Type to meet your requirements:

Name

Provide a unique name for the source type.

Input type

Provide the type of data that is processed.

Enable splitter:

Click to enable this option and select the Rule Set or File Set that is used to split the log records in the log files processed:

Rule Set: Choose from the options that are provided or add a Rule Set before selecting it.

File Set: Select the File Set option if you want to use a set of custom rules.

Enable annotator

Click to enable this option and select the Rule Set or File Set that is used to annotate the log records in the log files processed:

Rule Set: Choose from the options that are provided or add an additional Rule Set before selecting it.

File Set: Select the File Set option if you want to use a set of custom rules.

Deliver data on annotator execution failure: Select this option if you want records that fail during annotation to be added.

Index Config

Click **Edit Index Configuration** and specify the index configuration that is used to determine how annotated data is indexed.

Add physical source ID

If you want to view statistics for physical sources of data on the Data Ingestion dashboard in the Analytics Console, you need to specify a Physical Data Source ID. For more information, see the *Physical Data Source IDs* topic in the *Analytics Console* section of the IBM Operations Analytics - Log Analysis documentation.

3. Click **OK**. A message is displayed requesting that you confirm that you want to create the Source Type and indicating that you cannot edit some properties after the Source Type is saved. If you are satisfied that you have added the correct values to these properties, click **OK**.

Editing a Source Type

After a Source Type has been created, you cannot edit the properties associated with the Source Type. Use the **Edit** button to review the existing properties for a Source Type.

Procedure

To edit a Source Type:

1. In the Data Types workspace, expand the **Source Types** list.
2. Select the Source Type that you want to view and click **Edit**. The Source Type is opened in a new tab.
3. To close the Source Type, click **Close**.

Deleting a Source Type

You can delete an existing Source Type. However, a Source Type can only be deleted if it is not referenced by a Collection. Artifacts must be deleted in a specific order. The Collection must be deleted first, then the Source Type, and finally any Rule Set or File Set associated with the Source Type.

Procedure

To delete a Source Type:

1. In the Data Types workspace, expand the **Source Types** list.
2. Select the Source Type that you want to delete and click **Delete**.
3. Click **OK**.

Rule Set

A Rule Set is a file containing rules specified in Annotation Query Language (AQL). There are two available types of Rule Set:

Split

Used to split unstructured or semi-structured data into different fields.

Annotate

Used to annotate the fields already identified in a file containing unstructured or semi-structured data.

Adding a Rule Set

This topic outlines the steps that you must complete to create a Rule Set.

Procedure

To add a Rule Set:

1. In the Data Types workspace, click **Add > Rule Set**. The **Add Rule Set** tab is displayed
2. Enter the required information in each of the fields. Place your cursor on each of the fields for information about the requirements of that field.
3. Enter the following details for the Rule Set that you want to add:

Name

Provide a unique name for the rule set.

Type

Specify the type of rule set you want to create. The types available are Split and Annotate.

Rule File Directory

Specify the list of paths to the module source directories.

4. Click **OK**. A message is displayed requesting that you confirm that you want to create the Rule Set and indicating the properties that cannot be changed after the Rule Set is saved. If you are satisfied that you have added the correct values to these properties, click **OK**.

Editing a Rule Set

After a Rule Set has been created, you cannot edit the name or type properties associated with the Rule Set.

About this task

You cannot edit the **Name** or **Type** associated with the Rule Set. Use the **Edit** button to review the existing properties for a Rule Set.

Procedure

To view the properties of a Rule Set:

1. In the Data Types workspace, expand the **Rule Sets** list.
2. Select the Rule Set that you want to view and click **Edit**. The Rule Set is opened in a new tab.
3. To close the Rule Set, click **Close**.

Deleting a Rule Set

You can delete an existing Rule Set. However, a Rule Set can only be deleted if it is not referenced by a Source Type. Artifacts must be deleted in a specific order. Delete the Collection first, next delete the Source Type, and finally delete any Rule Set or File Set.

Procedure

To delete a Rule Set:

1. In the Data Types workspace, expand the **Rule Sets** list.
2. Select the Rule Set that you want to delete and click **Delete**.
3. Click **OK**.

File Sets

A File Set allows you to define a set of custom rules that are used to split or annotate a type of log data.

Adding a File Set

This topic outlines the steps that you must complete to add a File Set.

Procedure

To add a File Set:

1. In the Data Types workspace, click **Add > File Sets**. The **Add File Set** tab is displayed
2. Enter the required information in each of the fields. Place your cursor on each of the fields for information about the requirements of that field.
3. Click **OK**. A message is displayed requesting that you confirm that you want to create the File Set and indicating the properties that cannot be changed after the File Set is saved. If you are satisfied that you have added the correct values to these properties, click **OK**.

Editing a File Set

After a File Set has been created, you cannot edit the properties associated with the File Set.

About this task

You cannot edit the properties associated with an existing File Set. Use the **Edit** button to review the existing properties for a File Set.

Procedure

To view the properties of a File Set:

1. In the Data Types workspace, expand the **File Sets** list.
2. Select the File Set that you want to view and click **Edit**. The File Set is opened in a new tab.
3. To close the File Set, click **Close**.

Deleting a File Set

You can delete an existing File Set. However, a File Set can only be deleted if it is not referenced by a Source Type. Artifacts must be deleted in a specific order. Delete the Collection first, next delete the Source Type, and finally delete any Rule Set or File Set.

Procedure

To delete a File Set:

1. In the Data Types workspace, expand the **File Sets** list.
2. Select the File Set that you want to delete and click **Delete**.
3. Click **OK**.

Administrative tasks

You can change the refresh rate for dashboards, modify the service topology file and configure the time settings as part of the administrative tasks which may need to complete from time to time.

Configuring automatic refreshes for new dashboards

You can use the auto-refresh feature to regularly refresh a dashboard at scheduled intervals.

About this task

You can configure automatic refreshes only for a dashboard if all the charts in the dashboard use a relative time filter.

The default maximum number of charts that can be refreshed simultaneously across all dashboards is 20. For example, you can refresh 10 dashboards simultaneously if each dashboard has two charts. To edit the maximum number of charts that can be refreshed simultaneously, edit the `MAX_AUTO_REFRESH_CHARTS=<20>` property at the following location: `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties`, where `<20>` is the number of charts that can be refreshed simultaneously.

Auto-refresh supports intervals of 0, 1, 5, 15, 30, and 60 minutes.

Procedure

1. Open the dashboard.
2. To set the auto-refresh interval, click **Actions**, then **Auto-Refresh** and select the interval that you want to specify.
For example, to refresh the dashboard every 5 minutes, click **5 minutes**.

Results

The dashboard and the associated charts are automatically refreshed with the most current data at the specified interval.

A check mark beside the specified interval indicates that an auto-refresh interval was already specified.

Configuring automatic refreshes for existing dashboards

You can use the auto-refresh feature to regularly refresh an existing dashboard at scheduled intervals.

About this task

You can configure automatic refreshes only for a dashboard if all the charts in the dashboard use a relative time filter.

The default maximum number of charts that can be refreshed simultaneously across all dashboards is 20. For example, you can refresh 10 dashboards simultaneously if each dashboard has two charts. To edit the

maximum number of charts that can be refreshed simultaneously, edit the `MAX_AUTO_REFRESH_CHARTS=<20>` property at the following location: `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/apps/Unity.war/WEB-INF/unitysetup.properties`, where `<20>` is the number of charts that can be refreshed simultaneously.

Auto-refresh supports intervals of 0, 1, 5, 15, 30, and 60 minutes.

Procedure

Using a JSON editor, open and edit the original file with the following properties:

autoRefreshInterval:1

where `<1>` is the auto-refresh interval.

searchType: relative

where `<relative>` is the time filter.

Results

The dashboard and the associated charts are automatically refreshed with the most current data at the specified interval.

A check mark beside the specified interval indicates that an auto-refresh interval was already specified.

Configuring the timeout for the Log Analysis server

You can extend the timeout value for the Log Analysis server beyond the default of 120 minutes. For example, if you want to use Log Analysis as a monitoring console you can extend the timeout value.

About this task

To change the default value, modify the `server.xml` file that is used by the Log Analysis server.

Procedure

1. To stop the Log Analysis, enter the following command:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -stop
```

2. Open the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/server.xml` file
3. Add the following line to the file. The time is measured in minutes.

For example, to set the time out value to 8 hours, add the following line after the end of the last logical tag:

```
<ltpa>
<ltpa expiration="480"/>
</ltpa>
```

For example, you can insert it after the `httpEndpoint` tag.

4. Save the file.
5. To start the Log Analysis again, enter the following command:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -start
```

Results

The timeout value is changed.

Note: The session can time out in less time than is specified in the `server.xml` file. This issue can occur for a number of reasons outside of Log Analysis. For example, it can occur if the browser's timeout value is less than the value specified in the `server.xml` file.

Adding or removing IBM Tivoli Monitoring Log File Agent from an existing installation

After you install IBM Operations Analytics - Log Analysis, you can add or remove the IBM Tivoli Monitoring Log File Agent from an existing installation.

Before you begin

IBM Operations Analytics - Log Analysis is installed.

Procedure

1. Navigate to the <HOME>/IBM/InstallationManager/eclipse/ directory and run the command:

```
./launcher
```

Note: If you are accessing the installation environment remotely, ensure that your virtual desktop software is configured to allow you to view the graphical user interface for the IBM Installation Manager.

2. Click **Modify** and add or remove the IBM Tivoli Monitoring Log File Agent.

Changing the IP address for Log Analysis

You must change the IP address if you move the installation of Log Analysis to another server.

Before you begin

Ensure that the new IP address that you specify matches the one that is specified in the <HOME>etc/hosts directory.

About this task

You can change the IP address only. If you want to change the host name, contact IBM Support.

Procedure

1. Stop Log Analysis:

```
unity.sh -stop
```

2. Change the current IP address to the new IP address in the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/server.xml file.
3. Start Log Analysis:

```
unity.sh -start
```

Monitoring and configuring data ingestion statistics

You can monitor your data ingestion statistics and export these statistics to help you to meet the terms of your licence agreement. If you use Log Analysis as part of a bundle such as Netcool Operations Insight, you need to configure the free data ingestion limit.

Data loading entitlements

If you use the Entry Edition of Log Analysis, you can only load 2 Gigabytes (GB) of data per day. If you exceed this limit, Log Analysis warns you and then shuts down.

If you use the stand alone version of the Standard Edition of Log Analysis, you can load the amount of data that is specified in your licence agreement.

If you use Log Analysis as part of a bundle such as Netcool Operations Insight, you can load up to 2 Gigabytes of data for free. You can also define non-billable data that is not included in the total.

Configuring the data ingestion limit for bundle users

If you use Log Analysis as part of a bundle such as Netcool Operations Insight, you must configure the data ingestion limit after you install Log Analysis to ensure that you meet the terms of your licence agreement. This configuration is also required for auditability and compliance.

Bundle users are entitled to load up to 2 Gigabytes (GBs) of data for free. Bundle customers are also entitled to load non-billable data for free. Non-billable data is defined separately depending on the bundle and licence agreement. For example, for Netcool Operations Insight users can load events data for free. Application Performance Management users can load Application Performance Management log files for free. For more information about your entitlements, see your licence.

To enable the data ingestion limit and configure your non-billable data, you need to create a file called `seed.txt` and create non-billable data sources. For more information, see [“Configuring the data limit for bundled versions”](#) on page 326.

Monitoring data ingestion statistics

You can monitor data ingestion on the **Server Statistics** UI. For more information, see [“Server Statistics workspace”](#) on page 327.

You may also want to export your data ingestion statistics. You can use the command to do this. For more information, see [“export_statistics command”](#) on page 327.

Configuring the data limit for bundled versions

After you install Log Analysis as part of a bundle, you must configure the free data ingestion limit and segregate your data into billable and non-billable data sources.

About this task

This configuration is only required if you use Log Analysis as part of a bundle. Users of the stand alone version do not need to complete this task.

If you use Log Analysis as part of a bundle such as Netcool Operations Insight, you must configure the data ingestion limit after you install Log Analysis to ensure that you meet the terms of your licence agreement. This configuration is also required for auditability and compliance.

Bundle users are entitled to load up to 2 Gigabytes (GBs) of data for free. Bundle customers are also entitled to load non-billable data for free. Non-billable data is defined separately depending on the bundle and licence agreement. For example, for Netcool Operations Insight users can load events data for free. Application Performance Management users can load Application Performance Management log files for free. For more information about your entitlements, see your licence.

Procedure

1. To enable the free data ingestion limit, create a file that is called `seed.txt` in the `<HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity` directory.

Use the first line to specify the free ingestion limit in Megabytes (MBs). The free ingestion limit is up to 2GB average data per day of the 30 day rolling period. Read your license agreement to understand the free data ingestion limit. Use the next lines to specify the paths or directories where the related applications are installed. The log files are, in most cases, stored in the installation directories. These locations must match the log path that is defined when the data source is created. In some integrations, a data source can have an arbitrary string for a log path. In these cases, the strings in the `seed.txt` file must match the string that is specified in the data source.

The following sample shows a `seed.txt` file. Lines that start with a hashtag (#) are commented out and are ignored.

```
#FREE INGESTION LIMIT (MB)
2048
#Data Source locations to be ignored by statistics tracker
```

```
/home/LAuser1/LogAnalysis/logs  
/home/LAuser1/my/log/path
```

2. To restart Log Analysis, enter the following command:

```
<HOME>/IBM/LogAnalysis/utilities/unity.sh -restart
```

3. Create the required data sources for the non-billable data.

If the log path that you specify when you create the data source matches any of the file paths that you specified in the seed.txt file, Log Analysis does not count the data from these data sources in the billing statistics.

For more information about how to create a data source, see [“Data Source creation” on page 309](#).

Results

The free data ingestion limit which, is 2GBs average data for the 30 day rolling period, is displayed as a horizontal red line on the **Server Statistics** UI.

Server Statistics workspace

Use the Server Statistics workspace to display the rolling 30 day average and the peak average for data ingestion.

To calculate the 30 day rolling average, IBM Operations Analytics - Log Analysis measures the amount of data that is ingested over the previous 30 days, including the current day and divides by 30. Data that is ingested on the day of the upgrade is counted as part of the average.

The workspace displays details in the following fields:

30 day ingestion average

Displays the rolling 30 day average for the current day.

Peak 30 day ingestion average

Displays the highest rolling 30 day average.

Date of peak 30 day ingestion average

Displays the date when the peak day occurred.

The **Daily ingestion and Rolling 30 Day Average** graph displays the daily ingestion total and the rolling 30 day average for the specified date range. The graph is automatically updated when you enter a date.

If you want to refresh the chart without changing the date, click the **Refresh** button.

export_statistics command

Use the export_statistics command in the <HOME>/IBM/LogAnalysis/utilities directory to export statistics data to a table or comma-separated values (CSV) file line.

Syntax

This command has the syntax:

```
export_statistics <base_uri> [-u -p] | -f | -o | -t | -s | -h
```

where <base_uri> is the location of the IBM Operations Analytics - Log Analysis for which you want to export data. If you do not specify a value for the <base_uri> parameter, the \$UNITY_BASE_URI environmental variable is used.

Parameters

These parameters are also available:

-u

The user name for a user with administrative privileges. The format for this parameter is --u=username. Where username is a user with administrative privileges. If you do not specify a value for the -u parameter, the \$UNITY_USER environmental variable is used.

-p

The password for the user that you specified. The format for this parameter is --p=password. Where password is the password for the user name that you specified. If you do not specify a value for the -p parameter, the \$UNITY_PASS environmental variable is used.

-f

The format in which you want to export the data. The format for this parameter is --f format. Where format is table or CSV. The default value is Table.

-o

Outputs the statistics to a file. The format for this parameter is --o path. Where path is the file name and path for the export file. The default value is stdout.

-t

Specifies the type of data that you want to export. The format for this parameter is --t type. Where type is summary, daily, or thirtydayavg. The results are presented in table format with each column having a name. For example, Data Source. The default value is daily.

-s

Use this parameter to separate records when outputting to a CSV file. The format for this parameter is --s separator. Where separator is the separator that you want to use in the CSV file. The default value is a comma (,).

-h

(Optional) Displays help for this command. The <base_uri> parameter is not required for this parameter.

Note: A discrepancy might occur between the file size of the file that is ingested and the value that is returned by the export_statistics command. The value returned is less than or equal to the file size of the ingested file. This difference occurs because incomplete records in the log file are discarded by the Splitter reducing the file size.

The command outputs different information, depending on the type parameter that you specify, summary, daily, or thirtydayavg. The column headings for daily data are:

Data Source

The Data Source with which the ingested data is associated.

Collection

The Collection with which the Data Source containing the data is associated.

Date

The date on which the data was ingested.

Ingested Bytes

The volume of data, in bytes, that has been ingested.

Billable

The volume of data that is used for billing purposes.

Log Path

The path to the data source.

Hostname

The host name that indicates the source of the log file.

The column headings for the 30 day, rolling average are as follows:

Current Thirty Day Rolling Avg

The current 30 day, rolling average.

Thirty Day Rolling Avg High-Water Mark

Shows the average peak amount of data that is used over the last 30 days.

Date of Thirty Day Rolling Avg High-Water Mark

Shows the date when the amount of data ingested peaked.

The column headings for the 30 day average are as follows:

Date

Shows the date for each average.

Thirty Day Average

Shows the average amount of data that is used over 30 days.

Example

This example downloads the daily data source statistics from `http://unityserver.example.com:9988/Unity` using the user name `unityadmin` and the password `secretpassword`. The statistics are output to a CSV file: `~/Desktop/LogStats.csv`.

```
export_statistics http://unityserver.example.com:9988/Unity
--username=unityadmin --password=secretpassword --f csv --o ~/Desktop/LogStats.csv
```

To export the daily statistics, use the following command:

```
./export_statistics -u <user_name> -p <password> -t daily
```

The command outputs the following information:

Data Source	Collection	Date	Ingested Bytes	Billable	Log Path
localtest	localtest	2013-11-14	22640	True	/alogtest/SystemOut.log
localtest	localtest	2013-11-13	396200	True	/alogtest/SystemOut.log

Hostname
<hostname1>.example.com
<hostname2>..example.com

To export a summary, enter the following command:

```
./export_statistics -u <user_name> -p <password> -t daily
```

This command outputs the following information:

Current Thirty Day Rolling Avg	Thirty Day Rolling Avg High-Water Mark
13961	13961

Date of Thirty Day Rolling Avg High-Water Mark
2013-11-14 00:00:00 -0500

To export the 30 day average data, enter the following command:

```
./export_statistics -u <user_name> -p <password> -t thirtydayavg
```

This command outputs the following information:

Date	Thirty Day Average
2013-11-14	13961
2013-11-13	13206

Deleting data

To delete data from IBM Operations Analytics - Log Analysis, use the deletion tool.

The tool is available in the `<HOME>/utilites/deleteUtility` directory.

Use the tool to:

- Delete all of the data from a single data source or delete a specific subset of data from the specified data source.
- Delete all of the data from a single Collection or delete a specific subset of data from the specified Collection.
- Delete data from all the IBM Operations Analytics - Log Analysis collections for a time period that you specify.
- Delete data that is older than the specified retention period at regular intervals.

The tool is configured to run use case 1 by default. If you do not change the default value for the retention period, the default value of 24 hours is used.

Limitations and prerequisites for data deletion

Before you use the data deletion features, read the prerequisites and limitations.

The tool has one limitation. Do not run this utility for use case 1 or 2 when data ingestion is in progress for the specified data source or collection. For example, if you are loading data into a data source, do not delete data from the same data source until ingestion is complete.

The tool is configured to run use case 1 by default.

The query parameter is optional and can only be used with use cases 1 and 2. You cannot use this parameter to delete data that is stored in Hadoop.

For more information about the `delete.properties` parameters and values, see [“delete.properties” on page 334](#).

Deleting data

To remove data from Log Analysis, use the deletion tool.

About this task

Read the prerequisites and limitations. For more information, see [“Limitations and prerequisites for data deletion” on page 330](#)

Log Analysis stores Indexing Engine data in Apache Solr.

Standard Standard edition users can also store log file record data in Hadoop.

Use this utility to specify which types of data that you want to delete.

Procedure

1. Open the `<HOME>/IBM/LogAnalysis/utilities/deleteUtility/delete.properties` file.
2. Locate the lines and specify the use case that you want to run:

```
[useCase]
useCaseNumber = <usecase>
```

where `<usecase>` is one of these use cases.

useCase_1

Delete all of the data from a single data source or delete a specific subset of data from the specified data source. If you run this use case, locate this variable in the `delete.properties` file and specify an appropriate value:

- `dataSourceName`: Specify the name of the data source for which you want to delete data.
- `query`: Enter a query to delete a specific subset of data. For example, if you enter `severity:E`, all the log records in the data source that contain this severity status are deleted. You must use the Indexing Engine query syntax to specify the query.

You cannot use the query to delete data from Hadoop. You must use the data source name parameter to delete this data.

useCase_2

Delete all of the data from a single Collection or delete a specific subset of data for the specified collection. If you run this use case, locate this variable in the `delete.properties` file and specify an appropriate value:

- `collectionName`: Specify the name of the Collection for which you want to delete data.
- `query`: Enter a query to delete a specific subset of data. For example, if you enter `severity:E OR severity:W`, all the log records in the collection that contain either or both of these severity statuses are deleted. You must use the Indexing Engine query syntax to specify the query.

You cannot use the query to delete data from Hadoop. You must use the data source name parameter to delete this data.

useCase_3

Use this use case to delete data from all the IBM Operations Analytics - Log Analysis collections for a time period that you specify. You must specify a start and end time in the same time zone as the time specified in the `unitysetup.properties` file.

Note:

The utility deletes data for whole multiples of the time window value that occur during the specified time only. The time window value is defined in `unitysetup.properties` file. For example, if the time window value is one day and you enter the following start and end times, the utility deletes three complete days worth of data from all IBM Operations Analytics - Log Analysis collections.

```
startTime = <11/21/2013 14:00:00>
endTime = <11/25/2013 09:00:00>
```

The deleted days are 22, 23, and 24 November. Data from 14:00 to 00:00 on 21 November is not deleted nor is data from 00:00 to 09:00 on 25 November.

Note:

If you specify an `endTime` that is in the future, IBM Operations Analytics - Log Analysis displays a message that warns you that this can delete data from the current, active collection. Data is only removed from the current collection if the future date exceeds the value for the collection window. For example, if the collection window value is one day and the `endTime` is specified as `23:00:00` on the same day as the utility is run, data is not deleted from the active collection. However, if the collection window value is one day and the `endTime` is specified as `00:30:00` on the day after the current day, data is deleted from the active collection.

useCase_4

Use this use case to delete data that is stored for longer than the specified retention period at regular intervals. The retention period is specified by the `retentionPeriod=1d` parameter in the `delete.properties` file. The time is measured in days and one day is the default value. You must specify the retention period as a multiple of one day.

3. Save the `delete.properties` file.

4. Run the tool in stand-alone mode:

- To run the utility in stand-alone mode, run the command:

```
<Path to Python> deleteUtility.py <password> <target>
```

where `<Path to Python>` is the location to which python is installed and the `<password>` and the associated user name are defined in the `delete.properties` file.

`<target>` is the target data that you want to delete. Use it to specify the type of data that you want to delete, `-solr` or `-hadoop`. The default value is `-solr`

To delete Hadoop data, specify `-hadoop`. To delete Apache Solr data, specify `-solr`. To delete both, specify `-all`.

- To run the utility in cron mode:
 - a. Update the `<HOME>/IBM/LogAnalysis/utilities/deleteUtility/callDeleteUtility.sh` file. Ensure that the password specified here is correct.
 - b. Enter the following command to run the utility:

```
sh ./createCron.sh
```

- c. After the script runs, an entry is created in the cron file. To view the delete cron job that you created, enter the following command:

```
crontab -l
```

Results

When you delete data with this tool, a log file that is called `DeleteApplication.log` is created and stored in the `<HOME>/logs` directory.

Administering reference

Read to get reference information about the utilities and tools that you can use when you are administering Log Analysis.

export_statistics command

Use the `export_statistics` command in the `<HOME>/IBM/LogAnalysis/utilities` directory to export statistics data to a table or comma-separated values (CSV) file line.

Syntax

This command has the syntax:

```
export_statistics <base_uri> [-u -p] | -f | -o | -t | -s | -h
```

where `<base_uri>` is the location of the IBM Operations Analytics - Log Analysis for which you want to export data. If you do not specify a value for the `<base_uri>` parameter, the `$UNITY_BASE_URI` environmental variable is used.

Parameters

These parameters are also available:

-u

The user name for a user with administrative privileges. The format for this parameter is `--u=username`. Where `username` is a user with administrative privileges. If you do not specify a value for the `-u` parameter, the `$UNITY_USER` environmental variable is used.

-p

The password for the user that you specified. The format for this parameter is `--p=password`. Where `password` is the password for the user name that you specified. If you do not specify a value for the `-p` parameter, the `$UNITY_PASS` environmental variable is used.

-f

The format in which you want to export the data. The format for this parameter is `--f format`. Where `format` is `table` or `CSV`. The default value is `Table`.

-o

Outputs the statistics to a file. The format for this parameter is `--o path`. Where `path` is the file name and path for the export file. The default value is `stdout`.

-t

Specifies the type of data that you want to export. The format for this parameter is `--t type`. Where type is summary, daily, or thirtydayavg. The results are presented in table format with each column named. For example, Data Source. The default value is daily.

-s

Use this parameter to separate records when outputting to a CSV file. The format for this parameter is `--s separator`. Where separator is the separator that you want to use in the CSV file. The default value is a comma (,).

-h

(Optional) Displays help for this command. The `<base_uri>` parameter is not required for this parameter.

Note: A discrepancy might occur between the file size of the file that is ingested and the value that is returned by the `export_statistics` command. The value returned is less than or equal to the file size of the ingested file. This difference occurs because incomplete records in the log file are discarded by the Splitter reducing the file size.

The command outputs different information, depending on the type parameter that you specify, summary, daily, or thirtydayavg. The column headings for daily data are:

Data Source

The Data Source with which the ingested data is associated.

Collection

The Collection with which the Data Source containing the data is associated.

Date

The date on which the data was ingested.

Ingested Bytes

The volume of data that is ingested, in bytes.

Billable

The volume of data that is used for billing purposes.

Log Path

The path to the data source.

Hostname

The host name that indicates the source of the log file.

The column headings for the 30 day rolling average are as follows:

Current Thirty Day Rolling Avg

The current 30 day, rolling average.

Thirty Day Rolling Avg High-Water Mark

Shows the average peak amount of data that is used over the last 30 days.

Date of Thirty Day Rolling Avg High-Water Mark

Shows the date when the amount of data ingested peaked.

The column headings for the 30 day average are as follows:

Date

Shows the date for each average.

Thirty Day Average

Shows the average amount of data that is used over 30 days.

Example

This example downloads the daily data source statistics from `http://unityserver.example.com:9988/Unity` using the user name `unityadmin` and the password `secretpassword`. The statistics are output to a CSV file: `~/Desktop/LogStats.csv`.

```
export_statistics http://unityserver.example.com:9988/Unity
--username=unityadmin --password=secretpassword --f csv --o ~/Desktop/LogStats.csv
```

To export the daily statistics, use the following command:

```
./export_statistics -u <user_name> -p <password> -t daily
```

The command outputs the following information:

Data Source	Collection	Date	Ingested Bytes	Billable	Log Path				
localtest	localtest	2013-11-14	22640	True	/alogtest/SystemOut.log				
localtest	localtest	2013-11-13	396200	True	/alogtest/SystemOut.log				

Hostname									

<hostname1>.example.com									
<hostname2>..example.com									

To export a summary, enter the following command:

```
./export_statistics -u <user_name> -p <password> -t daily
```

This command outputs the following information:

Current Thirty Day Rolling Avg	Thirty Day Rolling Avg High-Water Mark
13961	13961

Date of Thirty Day Rolling Avg High-Water Mark	

2013-11-14 00:00:00 -0500	

To export the 30 day average data, enter the following command:

```
./export_statistics -u <user_name> -p <password> -t thirtydayavg
```

This command outputs the following information:

Date	Thirty Day Average
2013-11-14	13961
2013-11-13	13206

delete.properties

To remove data from IBM Operations Analytics - Log Analysis, use the deletion tool.

The delete.properties file is in the <HOME>/IBM/LogAnalysis/utilities/deleteUtility directory.

The delete utility supports four use cases. The use cases require general information, and information specific to each use case.

The hostName, port, userName, and delayDuration parameters are required.

The query parameter is optional and can only be used with use cases 1 and 2.

Table 85. General parameters	
Parameter	Value
useCase	The use case number that you want to run.
hostName	The host name that corresponds to the data source defined.
port	The https port. The default port is 9987.
userName	The user name for a user with administrative access rights.

Table 85. General parameters (continued)	
Parameter	Value
delayDuration	The delay interval (in milliseconds) between two consecutive rest (post) calls.
query	If you want to delete a specific type of data in a specified data source or collection, use this parameter to specify a query in the Indexing Engine syntax. For example, entering Severity:E deletes all the log records for the specified severity.
delete -solr -hadoop -all	Use this option to delete specific data. To delete data from your Indexing Engine instances, enter delete -solr. To delete data that is stored in Hadoop, enter delete -hadoop. To delete both, enter delete -all.

Use Case 1

dataSourceName: Specify the name of the data source for which you want to delete data.

query: Enter a query to delete a specific type of data. For example severity:E

Use Case 2

collectionName: Specify the name of the Collection for which you want to delete data.

query: Enter a query to delete a specific type of data. For example severity:E or severity:W

Use Case 3

```
startTime = <11/21/2013 14:00:00>
endTime = <11/25/2013 09:00:00>
```

Use Case 4

retentionPeriod=1d: Specify the retention period time. The time is measured in days, and one day is the default.

API guide

Use this guide to help you to integrate Log Analysis with other applications.

Manually configuring authentication for the REST API

Before you can use the REST API, you need to configure the authentication for communications with Log Analysis.

About this task

To enable authentication, you add the cross-site request forgery (CSRF) token to your search queries. The CSRF token that is generated by the Liberty component of Log Analysis. The token is valid only for a couple of hours. After this time, a new token is generated. When the token is updated, you need to update your REST API queries.

Procedure

1. To authenticate Log Analysis, do a POST call that includes the Log Analysis administrator's user and password:

```
https://<IOALA_hostname>:9987/Unity/j_security_check?j_
username=<IOALA_admin_user_name>
&j_password=<IOALA_admin_password>&action=Go
```

For example:

```
https://eg12345:9987/Unity/j_security_check?j_username=unityadmin
&j_password=unityadmin&action=Go
```

The return contains a cryptic response with Java Script code.

2. You must extract the CSRF token. Log Analysis returns a CSRF token, which, can be reused in subsequent queries. To find the token, you need to search the response string. For example:

```
sclaidtoken = "B7EFED2C60D9843729ADCEAF81B22527"
```

The sclaidtoken token is the CSRF token that is generated every time you log in to Log Analysis. You need to pass this value as a query parameter in all of your REST invocations.

3. Finally, you need to create a query that includes the CSRF token. Ensure that the Content Type is set to application or JSON. For example Content-Type=application/json.

For example, create a query that uses a POST method:

```
Method:
POST
URL:
https://IOALA_HOST:9987/Unity/Async/Search?
searchMode=async&CSRFToken=B7EFED2C60D9843729ADCEAF81B22527

Body:
Body:
{"start":0,"results":100,"outputTimeZone":"UTC",
"filter":{"range":{"timestamp":{"from":"07/05/2014 12:02:47.000 +0530,
12:02 PM","to":"07/05/2015 12:02:47.083 +0530, 12:02 PM",
"dateFormat":"dd/MM/yyyy HH:mm:ss.SSS Z"}}},
"logsources":[{"name":"/automation app","type":"tag"}],
"query":"*", "highlight":true,
"sortKey":["-timestamp"],
"facets":{"histogram_facet":{"date_histogram":{"field":"timestamp",
"interval":"week","outputDateFormat":"yyyy-MM-dd'T'HH:mm:ssZ"}}},
"allTermFacets":true,
"allTermFacetsSize":25,
"genericConcepts":100,
"genericKeys":100,
"locale":{"language":"en","country":"US","variant":""}}
```

The CSRFToken is generated by the Liberty component of Log Analysis. The token is valid only for a couple of hours. After this time, a new token is generated. When this happens, you must update your queries.

Using the Data Collector client to authenticate and invoke the REST API

Before you can use the REST API, you need to authenticate communications with Log Analysis. You can use the Data Collector client which, is available out of the box, to help you to do this. You can also use it to invoke the REST API and leverage public APIs.

Before you begin

- Ensure that you have at least a basic understanding of Java .
- Ensure that you are using Java 8.0 or later.
- Ensure that the Data Collector client that you use is from the same version of Log Analysis that you use for authentication. For example a Data Collector from 1.2 may not work with Log Analysis 1.3.

About this task

If you run your REST client on the same server where you installed Log Analysis, you do not need to complete steps 3 and 4 because the client certificate is already imported into the Java client that is installed with Log Analysis.

Procedure

1. Log in to the server where you installed Log Analysis.
2. Copy the <HOME>/IBM/LogAnalysis/utilities/datacollector-client/datacollector-client.jar file to the server where you want to run the client side code.

You must ensure that the datacollector-client.jar file is part of the classpath that is specified in your Java client.
3. Copy the <HOME>/IBM/LogAnalysis/wlp/usr/servers/Unity/resources/security/client.crt file from the Log Analysis server to a directory on the server where your REST call invoker client is.
For example, home/IBM/myCertificates/client.crt.
4. Import the certificate that you created in the previous step into your Java certificate store.
For example:

```
/home/user1/java/bin/keytool -import
-file /home/user1/IBM/myCertificates/client.crt
-keystore /home/user1/java/jre/lib/security/cacerts
-alias SCALA_HOST -storepass changeit
```

What to do next

For example, imagine that you use a Java program called MyClient.java with 'main' method that is under a package called "com.org.tools". You saved the Data Collector JAR file in the same directory as the program. Go to the directory and run the following command:

```
java -cp datacollector-client.jar:. com.org.tools.MyClient
```

The program runs and invokes the REST APIs that are exposed in the datacollectorclient.jar file.

To invoke the REST API, declare and define the class member variable for the REST Client APIs:

```
RestClient restClient = new RestClient(userID, password);
```

You can also use the following public APIs. You can use the endpoint addresses that you define in the Log Analysis REST API as the restAddr value:

```
public RequestResult postToServer(String restAddr, JSONObject json)
throws UnityException
public RequestResult putOnServer(String restAddr, JSONObject json)
public RequestResult getFromServer(String restAddr)
public RequestResult deleteOnServer(String deleteAddr)
```

To load a batch of data into the Log Analysis server hosted at exmp123.example.com, use the following code:

```
RestClient restClient = new RestClient("unityadmin","unityadmin");

JSONObject batch = .....

String restAddr = "https://exmp123.example.com:9987/Unity/DataCollector";

try {
    restClient.postToServer(restAddr, batch);
}catch(UnityException ue){
    System.err.println(ue.getMessage());
}
```

Search REST API overview

The Search REST interface can be used to execute search queries. Search can be executed through a HTTP POST request on `https:<host>:<port>;/Unity /Search`.

The Search REST interface is the primary search method that is used to execute search queries.

Searching over data sources

The input search request and the response back from USR are both modeled as JSON objects. In the following section, the structure of these input and output JSON objects are described.

JSON Format for Search Request

The input JSON for a search request is a single JSON object.

The input JSON object has the structure:

```
{
  "logSources": ["logsource1", "logsource2", ...],
  "query": "the search query string entered by the user",
  "filter": "//filter query (see section 4.1 below)",
  "queryLang": "the language of the search query",
  "start": 0,
  "results": 10,
  "getAttributes": ["attribute1", "attribute2", ...],
  "sortKey": ["key1", "key2", ...],
  "facets": {
    "facet1D1": "// facet request (see section 4.2 below)",
    "facet1D2": "// facet request (see section 4.2 below)",
    .....
  }
  "advanced": {
    "rankingHint": "a query expression that provides a ranking hint for Gumshoe",
    "explain": false,
    "interpretations": true,
    "catchallOnly": false,
    "rules": {
      "allRules": true,
      "categoryRules": true,
      "rewriteRules": true,
      "ignoreCategories": ["cat1", "cat2", ...]
    }
  }
  "grouping": ["groupingType1", "groupingType2", ..],
  "highlighting": true,
}
```

1,000 or more results and Custom Search Dashboard: When a query in a Custom Search Dashboard returns more than 1000 records, you get only 1000 results back. The search result returned includes a field `totalResults` which shows total number of matching results. Another field `numResults` gives the number of records returned. You can check these values in the Custom Search Dashboard script and handle the results accordingly.

The following table lists the semantics of the remaining attributes

Table 86. Search request input parameters

Name	Description	Default Value	Comments
logsources	logsources against which the search request must be performed	Required	<p>A list of logsources. This should be a JSON array. Each entry can be a logsource name or tag name. If a tag name is specified, all logsources under the tag will be included in the search</p> <pre>"logsources": [{"type": "tag", "name": "/ipo"}, {"type": "logSource", "name": "/DayTraderLogSource"}]</pre>
query	Search query	Required	Typically, the value of this parameter is whatever is entered by the end-user in a search box. However, any valid query string as per the Velocity query syntax (excluding range queries) is permitted.
filter	Application filter	No filter	Valid JSON object as described in section 4.1. This parameter is intended for applications to pass in filters in addition to the user search query. Conceptually, the overall query processed by USR is “query AND filter”. The separation into two parts is to allow for additional query manipulation of the “query” part when we implement advanced search capabilities.
start	Offset of the first result to return (Integer)	0	If specified value is negative, the value will be defaulted to 0; if the specified value is greater than the number of results, no results will be returned.
results	Number of results desired (Integer)	10	Min value is 1 (values <= 0 default to 1); maximum value is 1000.
getAttributes	Attributes to be returned for each result entry	Not required	<p>When this parameter is not specified in the request, the engine will return only the set of attributes marked as <code>retrievebyDefault</code> in the indexing configurations associated with the logsources in question.</p> <p>If this parameter is specified and is a non-empty array, then the attributes listed in the array are fetched.</p> <p>Finally, if the parameter is specified but is an empty array, then ALL retrievable attributes across all logsources will be returned for each result entry.</p>

Table 86. Search request input parameters (continued)

Name	Description	Default Value	Comments
sortKey	One or more fields on which to sort the result	Relevance order	<p>A valid value for this parameter is a comma-separated list of field names, with each field name prefixed by “+” or “-”. Each field name appearing in the list must have been declared to be a “sortable” field at index build time. The first field in the list is treated as the primary sort key, the second field (if present) as the secondary sort key, and so on. The “+” (resp. “-”) prefix is an instruction to sort the corresponding field values in ascending (resp. descending) order.</p> <p>For queries involving multiple logsources, sort keys must exist in all logsources involved in the query, and sort keys must have the same type across logsources.</p>
outputTimeZone	Time zone in which DATE field results should be formatted	<p>Collection time zone (for single logsource queries and multi-logsource queries where logsources have identical time zones).</p> <p>Server time zone (for multi-logsource queries where logsources have different time zones)</p>	
outputDateFormat	SimpleDateFormat string that specifies how DATE field results should be formatted	<p>UNITY_DATE_DISPLAY_FORMAT</p> <p>Read from unitysetup.properties</p>	

Filter query

A filter query is a Boolean query specified as a nested JSON record. In its simplest form a Boolean query consists of a basic query. A basic query can be a term query, wildcard query, phrase query or range query. Basic queries can also be combined using arbitrarily nested conjunctions (*AND* queries), disjunctions (*OR* queries) and negations (*NOT* queries) to form complex Boolean queries.

4.1.1 Basic filter queries

Term query

A term query specifies a field name and a term. It matches all documents for which the field contains the term. A term query is specified as follows:

```
{
  "term":
  {
    "myField": "termValue"
  }
}
```

Wildcard query

A wildcard query specifies a field name and a wildcard expression. It matches all documents for which the field matches the wildcard expression. A wildcard query is specified as follows:

```
{
  "wildcard":
  {
    "myField": "wildcardExpression"
  }
}
```

Phrase query

A phrase query specifies a field name and a phrase. It matches all documents for which the field contains the phrase. A phrase query is specified as follows:

```
{
  "phrase":
  {
    "myField": "phraseValue"
  }
}
```

Range query

A range query specifies a field name along with a lower bound (inclusive) and an upper bound (exclusive) for the field value. A range query is applicable only to numeric and date fields. For date fields, an additional date format must be provided. A range query is specified as follows:

```
{
  "range":
  {
    "myField":
    {
      "from": "lower-bound", // value will be included in the search
      "to": "upper-bound", // value will be excluded in the search
      "dateFormat": "date-format" // only for date fields
    }
  }
}
```

4.1.2 Complex filter queries

Complex filter queries can be constructed by combining one or more queries using AND, OR or NOT queries.

AND query

An AND query consists of two or more sub-queries. Sub-queries can be either a basic query or another complex query. A document satisfies an AND query only if it satisfies all of its sub-queries. An AND query is specified as follows:

```
{
  "and": [
    {
      "query1": ...
    },
    {
      "query2": ...
    },
    ...
    {
      "queryN": ...
    }
  ]
}
```

OR query

An OR query consists of two or more sub-queries. Sub-queries can be either a basic query or another complex query. A document satisfies an OR query if it satisfies at least one of its sub-queries. An OR query is specified as follows:

```
{ "or": [
  { "query1": ... },
  { "query2": ... },
  ...
  { "queryN": ... }
]
```

NOT query

A NOT query consists of a single sub-query. The sub-query can be either a basic query or a complex query. A document satisfies a NOT query if it does not satisfy the contained sub-query. A NOT query is specified as follows:

```
{ "not": { "query": ... } }
```

Facet requests

Different types of facet requests are supported by USR, along with the JSON format used to specify each type of facet request.

Each facet request is specified as a JSON key-value pair with the key being the facetID and the value being a JSON record. The type of facet being computed determines the structure of this JSON record. The supported facet types and their corresponding JSON request format are described here.

Term Facets

```
"myTermFacet": {
  "terms": {
    "field": "myField",
    "size": N
  }
}
```

Facet counting is performed on the field *myField* and the top-N most frequent facet values (for some positive integer N) is returned.

The next two facets (histogram and statistical) apply only to numeric fields. In other words, the field on which these facets are being computed must have been configured with a `dataType=LONG` or `dataType=DOUBLE` in the indexing specification associated with the IBM Operations Analytics collection(s) over which the facet request is being processed.

Histogram Facets

```
"myHistoFacet": {
  "histogram": {
    "field": "myField",
    "interval": 100
  }
}
```

Performs facet counting with buckets determined based on the *interval* value.

Statistical Facets

```
"myStatFacet": {
  "statistical": {
    "field": "myField",
    "stats": [ "min", "max", "sum", "avg", "count", "missing",
              "sumOfSquares", "stddev", "percentile" ]
  }
}
```

Performs simple aggregate statistics on a facet field. The "stats" attribute specifies which of these statistics should be computed for the given facet request.

- **Entry** If you are using the Entry Edition, you can use the **sum**, **min**, **max**, **avg**, and **count** functions.
- **Standard** If you use the Standard Edition, you can use the **missing**, **sumOfSquares**, **stddev**, and **percentile** functions.

Percentile operations allow you to run queries for a defined percentile. You can also run the operation for multiple percentiles. For example, to query the maximum and minimum results for the 50th, 95th, and 99th percentiles, you enter "stats": ["min", "max", "percentile,50,95,99"]. The facet response is:

```
{
  "min": 10,
  "max": 1000,
  "percentile": {
    "50": 100,
    "95": 200,
    "99": 225
  }
}
```

Date Histogram Facets

```
"myDateHistoFacet": {
  "date_histogram": {
    "field": "myField",
    "interval": "day",
    "outputDateFormat": "yyyy-MM-dd
    'T' HH:mm:ssZ"
  }
}
```

A version of the histogram facet specialized for date fields. The value of the *interval* attribute can be one of the string constants *year*, *month*, *week*, *day*, *hour*, or *minute*. The value of the *outputDateFormat* is any valid date format string as per the Java SimpleDateFormat class. This format string is used to represent the histogram boundaries in the response JSON coming out of USR.

For single collection data histogram facets, boundaries are based on the collection time zone (either from the index configuration, or from `unitysetup.properties` if missing in the index configuration). For multi-collection facets, boundaries are based on collection time zone if the time zone of all collections is identical. For multi-collection facets where collection time zones differ, boundaries are based on the server time zone.

Note: The results returned from the date histogram facet are not sorted. If you are plotting the resulting time intervals in a chart, you need to sort the JSON returned by the date histogram facet. For example, in python, if your search request is the following:

```
request = {
  "start": 0,
  "results": 1,
  "filter": {
    "range": {
      "timestamp": {
        "from": "01/01/2013 00:00:00.000 EST",
        "to": "01/01/2014 00:00:00.000 EST",
        "dateFormat": "MM/dd/yyyy HH:mm:ss.SSS Z"
      }
    }
  },
  "logsources": [{"type": "logSource", "name": "MyTest" }],
  "query": "*",
  "sortKey": ["-timestamp"],
  "getAttributes": ["timestamp", "perfMsgId"],
  "facets": {
    "dateFacet": {
      "date_histogram": {
        "field": "timestamp",
        "interval": "hour",
        "outputDateFormat": "MM-dd HH:mm",
        "nested_facet": {
          "dlFacet": {
```

```

        "terms":{
          "field":"perfMsgId",
          "size":20
        }
      }
    }
  }
}

```

First, retrieve the dateFacet from the JSON returned by the http request and then call the dateSort() function .

```

response = connection.post(
  '/Search', json.dumps(request),
  content_type='application/json; charset=UTF-8');
content = get_response_content(response)

#convert the response data to JSON
data = json.loads(content)

if 'facetResults' in data:
    # get the facet results
    facetResults = data['facetResults']

    if 'dateFacet' in facetResults:
        # get the dateFacet rows
        dateFacet = facetResults['dateFacet']

        # the results of the dateFacet are not sorted,
        # so call dateSort()
        dateSort(dateFacet)

```

where dateSort() is defined as follows:

```

#-----
# dateSort()
#-----
def dateSort(dateFacet):
    # This function parses the UTC label found in the dateFacet in
    # the format "mm-hh-DDD-yyyy UTC"
    # and returns an array in the form [yyyy, DD, hh, mm]
    def parseDate(dateLabel):
        aDate = map(int, dateLabel.split(" ")[0].split("-"))
        aDate.reverse()
        return aDate

    # call an in-place List sort, using an anonymous function
    # lambda as the sort function
    dateFacet.sort(
        lambda facet1, facet2: cmp(parseDate(facet1['label']),
        parseDate(facet2['label'])))
    return dateFacet

```

Nested Facets

A facet request can be nested inside another facet request, by specifying a nested_facet key. You can nest facets to any number of levels.

The following is a valid nested facet query, with a termsfacet query nested inside a date_histogram facet query:

```

"facets":{
  "dateFacet":{
    "date_histogram":{
      "field": "timestamp", "interval": "hour",
      "outputDateFormat": "MM-dd HH:mm",
      "nested_facet":{
        "severityFacet":{
          "terms":{
            "field": "severity",
            "size":10
          }
        }
      }
    }
  }
}

```



```

    }
  }
}

```

Percentile statistical functions

You can use the statistical function to return values for specified percentiles.

You can use the Search REST API or the user interface to create percentile statistical functions.

For example, to query the maximum and minimum results for the 50th, 95th, and 99th percentiles with the Search REST API, you enter "stats": ["min", "max", "percentile,50,95,99"]. The facet response is:

```

{
  "min": 10,
  "max": 1000,
  "percentile": {
    "50": 100,
    "95": 200,
    "99": 225
  }
}

```

Percentile queries are not calculated incrementally, unlike the other queries that are used in Log Analysis. This fact means that the query needs to run over the entire time range before it can return any results. Log Analysis limits the number of asynchronous windows that can run simultaneously for this function. This property is set in the MAX_NON_INCREMENTAL_WINDOWS property in the `unitysetup.properties`. The default value is 2.

For example, if you specify a percentile query based on a time range from August 1 2015 to August 10 2015 and MAX_NON_INCREMENTAL_WINDOWS=5 and COLLECTION_ASYNC_WINDOW=1d, only the most recent 5 days of data that is returned by the query are considered for percentile evaluation.

JSON Format for Search Response

The search results from USR are also packaged as a single JSON object .

The JSON object has the structure:

```

{
  "searchRequest": // copy of the entire input JSON object that
                  // generated this response
  "totalResults": // integer value representing total number of
                  // results for the query
  "numResults": // number of top-level results sent back within
                // this JSON ("top-level"
                // because a grouped/clustered result is counted as 1
  "executionInfo": {
    "processingTime": // time (in ms) measured from the receipt of the search
                     // request by USR to point when USR begins to construct the result
    "interpretations": // for advanced search post
    "rules": // for advanced search post
  }
  "searchResults": [
    // Array of JSON objects one per top-level result entry.
    // The size of this array will be the value of the "numResults"
    // attribute
  ]
  "facetResults": {
    "facetID1": { // Object with facet information for facetID1 }
    "facetID2": { // Object with facet information for facetID2 }
    .....
  }
}

```

Each element of the "searchResults" array will have the following structure:

```
{
  "resultIndex": // a number that denotes the position of this result in the
                // overall result set for this query
  "attributes": {
    "field1": "value1",
    "field2": "value2",
    ....
    // one key-value pair for each field of the result entry;
    // the set of fields
    // will be determined by the semantics of the getAttributes
    // parameter
  }
}
```

The JSON structure for the facet results depends on the specific type of facet request.

Term Facets

```
"facetID": {
  "total": // total number of distinct terms in the field
           // used for generating this facet
  "counts": [
    { "term": "term1", "count": 10},
    { "term": "term2", "count": 5},
    ...
  ]
}
```

Histogram Facets

```
"facetID": [
  { "low": 50, "high": 150, "count": 10},
  { "low": 150, "high": 250, "count": 25},
  ...
]
```

Statistical Facets

```
"facetID": {
  "max": // max value
  "min": //min value
  "sum": // sum value
  "avg": // avg value
  "count": // count value
  "missing": // missing values
  "sumOfSquares": // sumOfSquares value
  "stddev": // stddev value
}
```

In general, all three aggregates do not have to be present. Only the aggregates listed in the "stats" attribute of the corresponding facet request will be included.

Date histogram Facets

Identical to the output of the histogram facets, except that the "low" and "high" attributes will be represented according to the format string specified in the input date histogram facet request. For example, the output may be something that looks like the following:

```
"facetID": [
  { "low": "2012-01-01 10:00:00", "high":
    "2012-01-01 11:00:00", "count": 10},
  { "low": "2012-01-01 11:00:00", "high":
    "2012-01-02 12:00:00", "count": 10
    "label": "9-22-188-2012 UTC" },
  ...
]
```

Nested Facets

If the outermost facet is a term facet, the response will be as follows:

```
"total": // total number of distinct terms in the field used for
          // generating this facet
"counts": [
```

```

{ "term": "term1", "count": 10, "nested_facet":
  {nested facet result...}},
{ "term": "term2", "count": 5, "nested_facet":
  { nested facet result...}}]
...
]

```

Search query API

The /query API works like the /Search API, with the exception of the structure of the response JSON. This API returns the data in tabular format instead of hierarchical format.

Search Request

Search request structure is the same as the /Search API.

The only extra field is name, which is an optional field. The name is used as the ID of the data set in the results. If the name is not specified,

```
searchResults
```

is used as the ID.

```

{
  "name": "AllErrors",
  "start": 0,
  "results": 10,
  "filter": { },
  "logsources": [ ... ],
  "query": "*",
  "getAttributes": [ ... ],
  "facets": { "facetId1" :{...}, ...}
}

```

Other search parameters, such as **search filters**, **facets**, **nested facets**, **logsources**, **query**, **getAttributes**, **start**, and **results** are described in section 4.

Search Results

The search results are in a tabular format, which can be ingested by Custom Search Dashboards. The key 'data' in results points to an array of data sets. Each data set has ID, fields, and rows.

Results include one data set for search results and one for each facet that is specified in the request.

Search results data set uses the 'name' specified in the request as the ID. If not specified **searchResults** is used as ID.

Facet results use the facet ID used in the request as the ID for the data set. In case of term, histogram and date-histogram facets, 'count' is added to the fields along with the specified fields.

For statistical facets (max, min, sum, avg, count, missing, sumOfSquares, and stddev), field ID is generated by combining field name and the function name. For example, for 'min' it is `fieldname-min` where `fieldname` is the field included in the statistical facet. Similarly, for max it is `fieldname-max`.

```

{
  "data": [
    {
      "id": "AllErrors",
      "fields": [
        { "label": "fieldLabel1", "type": "TEXT", "id": "fieldId1" },
        { "label": "fieldLabel2", "type": "LONG", "id": "fieldId2" }
      ],
      "rows": [
        {
          "fieldId1": "value1",
          "fieldId2": "value2"
        }
      ]
    },
    {
      "id": "facetId1",
      "rows": [

```

```

{
  "fieldId1": "value1",
  "fieldId2": "value2",
  "count": "value3"
},
{
  "fieldId1": "value1",
  "fieldId2": "value2",
  "count": "value3"
},
{
  "fields": [
    {
      "label": "fieldLabel1", "type": "LONG", "id": "fieldId1" },
    {
      "label": "fieldLabel2", "type": "LONG", "id": "fieldId2"},
    {
      "label": "Count", "type": "LONG", "id": "count" } ]
  }
}

```

Search request and results

This example shows a search request and response with sample data.

Search request

```

{
  "start": 0,
  "results": 100,
  "name": "AllErrors",
  "logsources": [
    {
      "type": "tag",
      "name": "*"
    }
  ],
  "query": "*",
  "facets": {
    "termFacet01": {
      "terms": {
        "field": "msgclassifier",
        "size": 419
      }
    }
  }
}

```

Search results

```

{
  "data": [
    {
      "fields": [
        {
          "label": "msgclassifier",
          "type": "TEXT",
          "id": "msgclassifier"
        },
        {
          "label": "className",
          "type": "TEXT",
          "id": "className"
        },
        {
          "label": "logsource",
          "type": "TEXT",
          "id": "logsource"
        }
      ],
      "rows": [
        {
          "msgclassifier": "SRVE0250I",
          "className": "com.ibm.ws.wswebcontainer.VirtualHost",
          "logsource": "WASLogSource"
        },
        {
          "msgclassifier": "SECJ0136I",
          "className": "com.ibm.ws.wswebcontainer.VirtualHost",
          "logsource": "WASLogSource"
        }
      ]
    }
  ]
}

```

```

    },
    "id": "AllErrors"
  },
  {
    "rows": [
      {
        "msgclassifier": "SRVE0242I",
        "count": 132
      },
      {
        "msgclassifier": "CWPKEI0003I",
        "count": 3
      }
    ],
    "fields": [
      {
        "label": "msgclassifier",
        "type": "TEXT",
        "id": "msgclassifier"
      },
      {
        "label": "count",
        "type": "LONG",
        "id": "count"
      }
    ],
    "id": "termFacet01"
  }
]
}

```

Using the REST API to administer the Log Analysis data model

You can use HTTP methods and the REST API to load, create, update, and delete Log Analysis artifacts and batches of artifacts.

To load, create, update, and delete a batch of artifacts such as rule sets, file sets, source types, collections, and log sources, use a GET, POST, PUT, or DELETE method and the following URL:

```
https://<server>:<port>/Unity/<Artifact>
```

where *<server>* is the machine that IBM Operations Analytics - Log Analysis is installed on. *<port>* is the port that you want to use for REST API requests on the same machine. *<Artifact>* is the name of the artifact that you want to process, for example rule sets.

To load, create, update and delete a specific artifact, use a GET, POST, PUT, or DELETE method and the following URL:

```
https://<server>:<port>/Unity/<Artifact>?id=<ruleset_id>
```

Read the following documentation to get more information about specific input parameters and returned values.

Rule sets

You can use various HTTP methods to load, create, update, and delete rule sets.

Load (GET method)

To return all the rule sets, use a GET method and the following URL:

```
https://<server>:<port>/Unity/RuleSets
```

To return a specific rule set, use a GET method and the following URL:

```
https://<server>:<port>/Unity/RuleSets?id=<ruleset_id>
```

where *<ruleset_id>* is the ID of the rule set that you want to retrieve.

The operation returns the following values from the requested rule sets in the simple JSON format:

Table 87. Returned value for rule sets	
Returned value	Description
rulesFileDirectory	Full path to the file that contains the rules that govern how the splitting and annotating is done. The rules must be written in the Annotated Query Language (AQL) format.
type	This parameter can have a value of 1 or 0. 0 means that the rule set is used for splitting. 1 means that the rule set is used for annotation.
name	The name of the rule set.
id	The rule set identifier.

The operation returns the following value for a single rule set:

```
{
  "rulesFileDirectory": "/home/....;",      (- AQL Path)
  "type": 1,                               (- 0 for Split 1 for Annotate)
  "name": "windowsOSEventsLS-Annotate",
  "id": 3
}
```

Create (POST method)

To create a rule set, use a POST method and the following URL:

```
https://<server>:<port>/Unity/RuleSets?
```

To specify the values for the rule set, define them in the HTTP message body in the JSON format. The input values are listed in the table.

Table 88. Input values for POST method	
Input parameter	Description
rulesFileDirectory	Enter the AQL path.
type	This parameter can have a value of 1 or 0. 0 means that the rule set is used for splitting. 1 means that the rule set is used for annotation.
name	The name of the rule set.

The specification for the new rule set is defined in the input JSON. For example:

```
{
  "name": "Test",
  "type": 0,                               (- 0 for Split 1 for Annotate)
  "rulesFileDirectory": "/home/....;"      (- AQL Path)
}
```

Update (PUT method)

To update a rule set, use a PUT method and the following URL:

```
https://<server>:<port>/Unity/RuleSets
```

To specify the values for the rule set, define them in the HTTP message body in the JSON format. The input values are the same as those values listed in table 2.

The input JSON is the same as that which is used for the POST method.

Delete (DELETE method)

To delete a rule set, use a DELETE method and the following URL:

```
https://<server>:<port>/Unity/RuleSets
```

File sets

You can use various HTTP methods to load, create, update, and delete file sets.

Return file sets (GET method)

To return all the file sets, use a GET method and the following URL:

```
https://<server>:<port>/Unity/FileSets
```

To return a specific file set, use a GET method and the following request:

```
https://<server>:<port>/Unity/FileSets?id=<fileset_id>
```

where <fileset_id> is the ID of the file set that you want to retrieve.

The operation returns the following values from the requested file sets in the simple JSON format:

Table 89. Returned values for file sets	
Returned value	Description
className	The name of the Java class that is used by the file set.
fileType	This parameter value can be either 0 or 1. 0 means that the file set is Java-based. 1 indicates that the file set is Python-based.
fileName	The name of the file that contains the code that does the splitting or annotating.
type	This parameter value can be either 0 or 1. 0 means that the file is used to split log files. 1 indicates that the file set is used to annotate log files.
name	The name of the file set.
id	The file set identifier.

For example, the operation returns the following values in the JSON format for a file set:

```
{
  "className": "com.ibm.tivoli..annotator.JavacoreAnnotator",
  "fileType": 0, (- 0 for Java 1 for Script)
  "fileName": "JavacoreExtractor_v1.1.0.1.jar",
  "type": 1, (- 0 for Split 1 for Annotate)
  "name": "Javacore-Annotate",
  "id": 2
}
```

Create (POST method)

To create a file set, use a POST method and the following URL:

```
https://<server>:<port>/Unity/FileSets
```

You define the parameter values in the JSON format in the HTTP message body. For example:

```
{
  "name": "Test",
  "type": 0, (- 0 for Split 1 for Annotate)
  "fileType": 0, (- 0 Java 1 for Script)
  "fileName": "db2-content.jar", (- Jar file name
  "className": "TestClass" (- Java Class name)
}
```

The input values are the same as the ones that are specified in table 1 except for id. This value is generated when the file set is created.

Update (PUT method)

To update a file set, use a PUT method and the following URL:

```
https://<server>:<port>/Unity/FileSets
```

You define the parameter values in the JSON format in the HTTP message body. The input values are the same as the ones that are specified in table 1 except for id. This value is generated when the file set is created.

Delete (DELETE method)

To delete a file set, use a DELETE method and the following URL:

```
https://<server>:<port>/Unity/FileSets?id=<fileset_id>
```

where *<fileset_id>* is the ID of the file set that you want to delete.

Source types

You can use various HTTP methods to load, create, update, and delete source types.

Load (GET method)

To load a source type, use a GET method and the following URL:

```
https://<server>:<port>/Unity/SourceTypes
```

To load a single source type, use a GET method and the following URL:

```
https://<server>:<port>/Unity/SourceTypes?id=<sourcetype_id>
```

where *<sourcetype_id>* is the identifier of the source type that you want to load.

The possible input values are described in the table:

Table 90. GET method parameters	
Parameter	Description
indexingConfig	Enter the valid index configuration for the JSON file.
inputType	Specify the type of file that is loaded. 0, denoting log file, is the only possible value.
splitter:fileSet	Specify the file set that the splitter uses to split log files.

Table 90. GET method parameters (continued)	
Parameter	Description
splitter:ruleSet	Specify the rule set that the splitter uses to split log files.
splitter:type	Specify the type of splitter that is used. 0 means that the file is used for splitting log files. 1 means that the file is used for annotating log files.
name	Specify the name of the source type that you want to create.
id	Specify the identifier of the source type that you want to load.
annotator:fileSet	Specify the file set that the annotator uses to annotate log files.
annotator:ruleSet	Specify the rule set that the annotator uses to annotate log files.
annotator:type	Specify the type of annotator that is used. 0 means that the file is used for splitting log files. 1 means that the file is used for annotating log files.
annotator:postOnFailure	Specify whether you want to annotator to post results if the annotation process fails. The default value is false.

You define the parameter values in the JSON format in the HTTP message body. For example:

```
{
  "indexingConfig": {},
  "inputType": 0,
  "splitter": {
    "fileSet": <fileset_id>,
    "ruleSet": <ruleset_id>,
    "type": 1
  },
  "name": "Javacore",
  "id": <id>,
  "annotator": {
    "fileSet": <fileset_id>,
    "ruleSet": <ruleset_id>,
    "type": 1,
    "postOnFailure": false
  }
}
```

- Valid index configuration JSON
- 0 for log file
- 0 for Split 1 for Annotate
- 0 for Split 1 for Annotate

Create (POST method)

To create a source type, use a POST method and the following URL:

```
https://<server>:<port>/Unity/SourceTypes
```

You define the parameter values in the JSON format in the HTTP message body. The values are the same as the ones that are listed in table 1. For example:

```
{
  "name": "Test",
  "indexingConfig": {},
  "inputType": 0,
  "splitter": {
    "ruleSet": <ruleset_id>,
    "fileSet": <fileset_id>
  },
  "annotator": {
    "ruleSet": <ruleset_id>,

```

- Valid index configuration JSON
- 0 for log file

```

    "fileSet": <fileset_id>,
    "postOnFailure": true
  }}

```

Update (PUT method)

To update a source type, use a PUT method and the following URL:

```
https://<server>:<port>/Unity/SourceTypes?id=<sourcetype_id>
```

where <sourcetype_id> is the identifier of the source type that you want to update.

You define the parameter values in the JSON format in the HTTP message body. The values are the same as the ones that are listed in table 1. The input JSON is the same as that described for POST method.

Delete (DELETE method)

To delete a source type, uses a DELETE method and the following URL:

```
https://<server>:<port>/Unity/SourceTypes?id=<sourcetype_id>
```

where <sourcetype_id> is the identifier of the source type that you want to delete.

Collections

You can use various HTTP methods to load, create, update, and delete collections.

Load (GET methods)

To load a single collection, use a GET method and the following URL:

```
https://<server>:<port>/Unity/Collections
```

To return a specific collection, use a GET method and the following URL:

```
https://<server>:<port>/Unity/Collections?id=<collection_id>
```

where <collection_id> is the ID of the collection that you want to retrieve.

The method returns the information in the JSON format. The returned values are listed in the table.

Table 91. Parameters for GET method	
Parameter	Description
indexingConfig	The valid index configuration value for the JSON file.
sourceType	The source type ID of the returned collection.
name	The name of the returned collection.
id	The identifier of the returned collection.
annotator	This parameter value can be either 0 or 1. 0 means that the source type is used to split log files. 1 indicates that the source type is used to annotate log files.

For example:

```

{
  "indexingConfig": null,
  "sourceType": 15,
  "name": "Javacore-Collection1",
  "id": 1,

```

```
{
  "annotator": 0
}
```

Create (POST method)

To create a collection, use the POST method and the following URL:

```
https://<server>:<port>/Unity/Collections
```

You define the parameter values in the JSON format in the HTTP message body. The parameter values are:

Name

Specify the name for the collection.

sourceType

Specify the number that represents the new source type.

For example:

```
{
  "name": "Test",
  "sourceType": 6
}
```

Update (PUT method)

To update a collection, use a PUT method and the following URL:

```
https://<server>:<port>/Unity/Collections?id=<collection_id>
```

where *<collection_id>* is the identifier of the collection that you want to update.

You define the updated parameter values in the JSON format in the HTTP message body. The parameter values and input JSON are the same as those that are used by the POST method.

Delete (DELETE method)

To delete a collection, use a DELETE method and the following URL:

```
https://<server>:<port>/Unity/Collections?id=<collection_id>
```

where *<collection_id>* is the identifier of the collection that you want to delete.

Data sources

You can use various HTTP methods to load, create, update, and delete data sources.

Data sources are called log sources in the REST API files.

Load (GET method)

To load all the data sources and associated tags, use a GET method and the following URL:

```
https://<server>:<port>/Unity/Logsources?pattern=*
```

The data sources are returned in a JSON array. The returned values for each data source are listed in the table.

Table 92. Returned values for GET method	
Parameter	Description
logsource_pk	The ID of the log source.
name	The name of the tag or log source.

Table 92. Returned values for GET method (continued)	
Parameter	Description
description	Description for the data source.
tag_id	The ID for any tags that are associated with the data source.
service_topology	The name of the group hierarchy that is associated with the data source.

For example:

```
[
  {
    "logsource_pk": 1,
    "name": "_alerts",
    "description": "Built-in datasource for indexing alerts",
    "tag_id": 4,
    "service_topology": ""
  },
  {
    "logsource_pk": 2,
    "name": "access1.log",
    "description": "",
    "tag_id": 7,
    "service_topology": "service:cloud provider application
> middleware:webserver > hostname:cldegd58"
  },
  ....
]
```

To load the details for a specified data source, use a GET method and the following URL:

```
https://<server>:<port>/Unity/DataSourceConfig?
datasourceconfigid=<logsource_pk>
```

where *<logsource_pk>* is the ID of the log source that you want to load.

The data sources are returned in a JSON array. The returned values are described in the following table:

Table 93. Returned values	
Returned value	Description
description	Description for the data source.
configType	The type of data source. The values are custom, local, and remote.
rollingfile	If you collect rolling log files with this data source, this value is true. If you do not, it is false.
hostname	The host name for the server where the data source resides.
password	The password for the user name that is associated with the data source.
sourceType	The number which represents the source type that is associated with the data source.
datasourcepath	The full path and log file name for the log file records that the data source loads into Log Analysis.
logsource_pk	The ID for the log source.
name	The name of the data source.

Table 93. Returned values (continued)	
Returned value	Description
filepattern	The file pattern that is associated with the data source.
collectionid	The identifier for the collection that is associated with the data source.
username	The ID for the user name who created the data sources.
group	The name of the group hierarchy that is associated with the data source, if any.

For example:

```
{
  "description": "",
  "configType": "custom",
  "rollingfile": false,
  "hostname": "cldegd60",
  "password": null,
  "sourceType": 3,
  "datasourcepath":
    "DemoWinOSEvents/WinOSEvents.txt",
  "logsource_pk": 11,
  "name": "sample-WinOS-events",
  "filepattern": null,
  "collectionId": 24,
  "username": null,
  "group": ""
}
```

Create (POST method)

To create a new data source, use a POST method and the following URL:

```
https://<server>:<port>/Unity/DataSourceConfig
```

You define the parameter values in the JSON format in the HTTP message body. The input parameters are listed in the table.

Table 94. Input parameters for POST method	
Parameter	Description
configType	The type of data source. You must use custom.
hostname	The host name of the machine where the data that the data source collects resides.
username	The user name that is associated with the data source. This parameter is optional.
password	The password that is associated with the user name. This parameter is optional.
datasourcepath	The full path and log file name of the log file that is loaded by this data source.
sourceType	The identifier of the source type that is associated with the data source.

Table 94. Input parameters for POST method (continued)

Parameter	Description
rollingfile	If you want to use the data source to collect rolling log files, set this parameter to True. If you do not, set it to False.
filepattern	The file pattern that is associated with the data source. This parameter is optional.
name	The name of the data source.
description	The description for the data source. This parameter is optional.
group	The name of the group hierarchy that is associated with the data source. This parameter is optional.

For example:

```
{
  "configType": "local",
  "hostname": "example.com",
  "username": null,
  "password": null,
  "datasourcepath":
    "/home/....../SystemOut2.log",
  "sourceType": <sourcetype id>,
  "rollingfile": false,
  "filepattern": null,
  "name": "Test",
  "description": "",
  "group": ""
}
```

The following table summarizes some of the response codes. The error code for all of these errors is 400 (Bad request).

Table 95. Response codes for POST methods

Error	Input JSON	Example API response in JSON
The source type ID that is specified in the input JSON is not valid.	"sourceType"	{ "isvalid": true, "message" : "CTGLA0773E : Invalid source type ID" "<source_type_id>" }
The collection ID that is specified in the input JSON is not valid.	"collectionID"	{ "isvalid": true, "message" : "CTGLA0774E : Invalid collection ID" "<collection_id>" }
The host name and log path combination is already in use.	"hostname" and "datasourcepath"	{ "isvalid": true, "message" : "CTGLA0750E: Data source for this host name and path combination already exists (<host_name>, <log_path>" }

Table 95. Response codes for POST methods (continued)

Error	Input JSON	Example API response in JSON
The data source name that is specified in the input JSON is already in use.	"name"	{ "isvalid":true,"message": "CTGLA0759E : A Log source for the name <data source name> already exists" }
The data source name input that is specified in the JSON is blank or contains more than 30 characters.	"name"	{ "isvalid":true,"message": "CTGLA0775E : The data source name must not be blank or longer than 30 characters" }

Update (PUT method)

To update a data source, use a PUT method and the following URL:

```
https://<server>:<port>/Unity/DataSourceConfig
```

You define the parameter values in the JSON format in the HTTP message body. The input values are the same as the ones that are specified in table 3.

Table 96. Input parameters for POST method

Parameter	Description
logsource_pk	The identifier for the log source that you want to load.
configType	The type of data source. This must be custom.
hostname	The host name of the machine where the data that the data source collects resides.
username	The user name that is associated with the data source. This parameter is optional.
password	The password that is associated with the user name. This parameter is optional.
datasourcepath	The full path and log file name of the log file that is loaded by this data source.
sourceType	The identifier of the source type that is associated with the data source.
rollingfile	If you want to use the data source to collect rolling log files, set this parameter to True. If you do not, set it to False.
filepattern	The file pattern that is associated with the data source. This parameter is optional.
name	The name of the data source.
description	The description for the data source. This parameter is optional.

Table 96. Input parameters for POST method (continued)	
Parameter	Description
group	The name of the group hierarchy that is associated with the data source. This parameter is optional.

For example:

```
{
  "description": "",
  "configType": "local",
  "rollingfile": false,
  "hostname": "nc91209819.in.ibm.com",
  "password": null,
  "sourceType": "7",
  "datasourcepath": "/home/unity/IBM/db2diag.log",
  "logsource_pk": 1,
  "name": "sample-db2",
  "filepattern": null,
  "collectionId": 14,
  "username": null,
  "group": "Service:IP0"
}
```

Delete (DELETE method)

To update a data source, use a DELETE method and the following URL:

```
https://<server>:<port>/Unity/DataSourceConfig?logsource_pk
=<logsource_pk>
```

where `<logsource_pk>` is the identifier of the log source that you want to delete.

REST API for asynchronous searches

You can use HTTP methods such POST, GET, and DELETE to customize your asynchronous searches.

GET method

You can use the GET method to return the details of all current searches.

The user ID is implied during authentication when the method is called.

The GET method returns the information that is contained in the search request properties.

You can use the `currentState` property to determine the status of a search.

Request values

Table 1 outlines the values that you must specify in the request.

Table 97. Request values for GET			
Name	Type	Default value	Description
Size	Number	30	Indicates the maximum number of records that are returned. To return all records, specify -1.
Start	Number	0	Index number for the first record that is returned.

Returned values

Table 2 outlines the values that are returned by the function.

Table 98. Returned values for GET	
Attribute	Description
currentState	Returns the status of a search. Possible values are QUEUED, PARSING, RUNNING, PAUSED, FINALIZING, FAILED, DONE.
completionProgress	A number from 1 - 100 that indicates an approximation of the progress of the search.
isDone	Indicates that the search is finished.
isFailed	Indicates that the search failed with an unrecoverable error. For example, if the search string uses an incorrect syntax.
isPaused	Indicates that the search is paused.
messages	Contains the error and debugging messages that are generated during the search.
priority	A number from 1 - 10 that indicates the priority of the search.
searchRequest	Lists the JSON file that is used to define the search.

POST method

You can use the POST method to start a search and return the search request ID.

You use the search parameter to specify a JSON file. The file specifies a search query that contains information about the search string, time filter, facets, and more.

The function returns the search request ID. You can add these amendments to the search URL to view and manage the search. Table 1 outlines these amendments.

Table 99. Search URL amendments	
URL amendment	Description
search/<search_request_ID>	View the status of the search request.
search/<search_request_ID>/<action>	Run the action commands that you specify in the <action> parameter. Some possible actions are pause, cancel, and preview.
search/<search_request_ID>/results	View the search results.

Request values

Table 2 outlines the values that you must specify in the request.

Table 100. Request values for POST function			
Name	Type	Default value	Description
search	JSON	n/a	JSON file that specifies the search query details such as the search string, time filter, facets and more. The JSON specification is required for this function.

Table 100. Request values for POST function (continued)			
Name	Type	Default value	Description
mode	Enumeration	normal	The valid values are blocking and async. If the mode is set to async, the search runs asynchronously. If the mode is set to blocking, the function returns the search request ID when the search finishes.
timeout	Number	86400	The number of seconds that the search is retained for after processing stops.

Returned values

If it is successful, the function returns the status and the service request ID. If it is not successful, the function returns an error code.

DELETE/search/<search_request_id> method

You can use the DELETE/search/<search_request_id> method to delete the search that is specified by the GET method.

Request values

You do not need to specify any parameters for the request. The operation deletes the search request that is specified in the GET operation.

Response codes

The two possible response codes are outlined in table 1.

Table 101. Response codes	
Status code	Description
200	Search deleted successfully.
404	The search request does not exist.

Returned values

The operation returns the same values as the GET operation. For more information, see [“GET method” on page 360](#).

GET/search/<search_request_id> method

You can use the GET/search/<search_request_id> method to return details about the search request that is associated with the user who runs the call.

The method returns information about the search request properties such as the time taken to complete the search.

The parameters to POST /search provides details on search request properties when creating a search.

Request values

There are no request values as the operation uses the user ID of the user who runs the operation to identify the search request ID.

Response codes

The response codes are outlined in table 1.

Table 102. Response codes	
Status code	Description
200	Search request was retrieved successfully.
403	Cannot retrieve search request. User does not have the authorization to view the search request.
404	The search request does not exist.

Return values

The operation also returns a JSON file that contains information about the search request.

POST/search/<search_request_id>/action method

You can use the POST/search/<search_request_id>/action method to request an action for a specific search request.

Request values

Table 1 outlines the request values that you can use with this operation.

Table 103. Request values			
Name	Type	Default	Description
name	Enumeration	n/a	The action that is carried out. Valid values are pause, unpause, cancel, setpriority

The following values are valid for the name request:

pause

Suspends the current search.

unpause

Resumes the current search if it is paused.

cancel

Stops the current search and deletes the cached results.

setpriority

Sets the priority of the search request. This value can be any value from 1 - 10.

Response codes

The response codes are outlined in table 2.

Table 104. Response codes	
Status code	Description
200	Search updated successfully.

Table 104. Response codes (continued)	
Status code	Description
403	User does not have the authorization that is required to edit the search request.
404	The specified search request does not exist.

Returned values

The operation returns the status of the search request and any related messages.

GET/search/<search_request_id>/results method

You can use the GET/search/<search_request_id>/results method to return the results of the specified service request at the time the request is made.

Response codes

The response codes are outlined in table 1.

Table 105. Response codes	
Status code	Description
200	Results that are returned successfully.
204	Search exists but the search results are not ready. Run the request again.
403	User does not have the authorization that is required to display the specified search request.
404	The specified search request does not exist.

Returned values

The request returns the status and the related values. For example, a successful request returns the JSON file that contains the search results and related facets.

Appendix A. Notices

This information was developed for products and services that are offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America*

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

*Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan*

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

*IBM Corporation
224A/101
11400 Burnet Road
Austin, TX 78758 U.S.A.*

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

COPYRIGHT LICENSE:

This information contains sample application programs in source language, which illustrate programming techniques on various operating platforms. You may copy, modify, and distribute these sample programs in any form without payment to IBM, for the purposes of developing, using, marketing or distributing application programs conforming to the application programming interface for the operating platform for which the sample programs are written. These examples have not been thoroughly tested under all conditions. IBM, therefore, cannot guarantee or imply reliability, serviceability, or function of these programs. The sample programs are provided "AS IS", without warranty of any kind. IBM shall not be liable for any damages arising out of your use of the sample programs.

© Copyright IBM Corp. 2015. All rights reserved.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at www.ibm.com/legal/copytrade.shtml.

Terms and conditions for product documentation

Permissions for the use of these publications are granted subject to the following terms and conditions.

Applicability

These terms and conditions are in addition to any terms of use for the IBM website.

Personal use

You may reproduce these publications for your personal, noncommercial use provided that all proprietary notices are preserved. You may not distribute, display or make derivative work of these publications, or any portion thereof, without the express consent of IBM.

Commercial use

You may reproduce, distribute and display these publications solely within your enterprise provided that all proprietary notices are preserved. You may not make derivative works of these publications, or reproduce, distribute or display these publications or any portion thereof outside your enterprise, without the express consent of IBM.

Rights

Except as expressly granted in this permission, no other permissions, licenses or rights are granted, either express or implied, to the publications or any information, data, software or other intellectual property contained therein.

IBM reserves the right to withdraw the permissions granted herein whenever, in its discretion, the use of the publications is detrimental to its interest or, as determined by IBM, the above instructions are not being properly followed.

You may not download, export or re-export this information except in full compliance with all applicable laws and regulations, including all United States export laws and regulations.

IBM MAKES NO GUARANTEE ABOUT THE CONTENT OF THESE PUBLICATIONS. THE PUBLICATIONS ARE PROVIDED "AS-IS" AND WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY, NON-INFRINGEMENT, AND FITNESS FOR A PARTICULAR PURPOSE.

IBM Online Privacy Statement

Privacy Policy Considerations

IBM Software products, including software as a service solutions, ("Software Offerings") may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user, or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering's use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session and persistent cookies that collect each user's user name and password for purposes of session management, authentication, enhanced user usability, and single sign-on configuration. These cookies cannot be disabled.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, see IBM's Privacy Policy at <http://www.ibm.com/privacy> and IBM's Online Privacy Statement at <http://www.ibm.com/privacy/details> in the section entitled "Cookies, Web Beacons and Other Technologies" and the "IBM Software Products and Software-as-a-Service Privacy Statement" at <http://www.ibm.com/software/info/product-privacy>.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Adobe, Acrobat, PostScript and all Adobe-based trademarks are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, other countries, or both.



Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and service names might be trademarks of IBM or other companies.



Part Number:
Product Number:

(1P) P/N: