Syslog UNIX Tool Set (UTS)

User Reference Manual

http://www.newnettechnologies.com/

Log Tracker Syslog UTS, User Reference Manual

Special Notes On This NNT Log Tracker Manual

NNT Log Tracker is based on the CorreLog Sigma Framework, an extensible security framework for small, medium and large enterprises. All ancillary references to "CorreLog" in this manual, and all screen shot details depicting references to "CorreLog", actually refer to this underlying framework structure of "NNT Log Tracker", or the Log Tracker program itself.

Except for this specific notation, identified above, all information herein refers to the latest version of "NNT Log Tracker". Any references to "CorreLog" herein are synonymous to either the "NNT Log Tracker" program, or the company "New Net Technologies Ltd", depending on the context of the reference.

Table of Contents

Section 1: Introduction

This document contains installation and application notes regarding the Log Tracker UNIX Agent and UNIX Tool Set (Log Tracker UTS), which is a compact set of software tools that instrument a UNIX platform system with special Syslog functions. These functions can be used to augment the existing native Syslog capability normally found on UNIX platforms, such as monitoring of streaming log files, file object existence, and encryption of Syslog messages.

The Log Tracker UTS is very lightweight and easy to install. Specific packages are included to support various different UNIX platforms, including (but not limited to) Linux, Solaris, HPUX, and AIX. Packages and versions can be downloaded from the "Home" screen of Log Tracker. The appropriate UNIX agent program can then be transferred to specific UNIX platforms of interest in an organization or enterprise.

If you wish to get started immediately with the installation of the Log Tracker UTS, see notes at the bottom of this current section.

This manual should be of interest to network managers and administrators, responsible for installing and maintaining Log Tracker. This manual will also be useful to any developers who are interested in using this software as a basis for larger enterprise management strategies.

Log Tracker UNIX Tool Set (UTS) Overview

The Log Tracker Syslog UNIX Tool Set (UTS) is a collection of executables and files that augment the native Syslog capability of a UNIX platform. In particular, this tool set includes a non-intrusive UNIX Agent program that monitors streaming log files, and relays Syslog messages to a Syslog receiver, permitting easy integration of the Log Tracker Security Correlation Server with UNIX platforms. Additionally, the tool set includes a "File Integrity Monitor" program, which can be installed to continuously test whether certain directories of files have been changed.

It should be immediately noted that the Log Tracker UTS package is optional software, and does not replace the native UNIX Syslog capability of a platform. All common UNIX operating systems have native Syslog capability. Therefore, it is not necessary to install the Log Tracker UTS software to fully use Log Tracker. Instead, the user can simply adjust the native Syslog process (via the "syslog.conf" file) to relay messages to Log Tracker. This will be entirely sufficient for many enterprises.

For those sites that need special Syslog functionality, such as monitoring of arbitrary streaming log files (such as Apache transfer and error logs) or monitoring of object access (such as the system password or hosts file) the Log Tracker UTS software provides this functionality as described here. This functionality enhances, but does not replace, the normal Syslog capability found in modern UNIX operating systems.

The Log Tracker UTS consists of the following programs.

- **Log Tracker Logfile Monitor Service.** This is a compact but powerful program, which allows arbitrary log files to be instrumented with match patterns. When specific match patterns are detected in streaming log files, Syslog messages of the appropriate severity and facility are sent to the Syslog server program.

- **Log Tracker UNIX File Integrity Monitor (FIM) Service.** This is a second agent program, which continuously tests directories of files (specified by the user via a configuration file.) The program tests to see if files have been added, deleted, or modified on the system, and sends a Syslog message if changes are detected.

- **Log Tracker Sendlog Utility**. This is a simple utility that can be used in scripts, or launched by application programs to send Syslog messages to a Syslog server host. The utility is a completely stand-alone executable that relies on no other files, hence is easily adapted to user specific applications.

The above programs are documented in this manual, including installation and configuration, along with extra application notes that describe how to perform advanced configuration of the system.

The Log Tracker UTS software is similar to the Log Tracker WTS software for Windows system. The UTS software operates in a similar fashion to the WTS software, including support for advanced encryption, the file import utility, source filtering of messages, remote configuration and other features. The user may wish to review the corresponding Windows documentation, included as part of the main Log Tracker distribution, for additional notes and discussions.

Log Tracker UNIX Tool Set Interoperability

Note that, although these programs contain special features (such as data encryption) that work with the Log Tracker Server, the programs also work with any standard Syslog receiver. Although this manual will assume that these programs are being installed to support the Log Tracker Server, all information in this manual (unless otherwise specifically noted) can be generalized to apply to any Syslog receiver program running on the network.

Because the Syslog is highly interoperable, the installer can leverage this standards based protocol to develop new management techniques that interoperate with a variety of other software.

The Log Tracker UTS software contains one special feature that can be used ONLY with the Log Tracker, and that is the data encryption functions of the software. The user can enable data encryption of Syslog messages between the Log Tracker UTS and Log Tracker Server for those situations that might warrant this (such as sending messages across a public internet.) More information on the data encryption function is found in the section of this manual dealing with the Syslog Message Server Configuration file.

The Log Tracker UNIX Tool Set, Fast Start

The remainder of this manual will deal with the various detailed aspects of the Log Tracker UTS software in detail. For those users wishing a quick start, the following information will get the Log Tracker UTS software up and running as quickly as possible on a UNIX platform, permitting you to immediately begin using the program.

The Log Tracker UTS is simple to install, and consists of a manual procedure. A single process and configuration file is required on the UNIX platform.

1. The UNIX root administrator can download the specific UNIX package with a web browser directly from the Log Tracker Server to the target UNIX

platform. On the "Home" screen of Log Tracker, the user clicks on the "Download UNIX Agent Packages" link to view the list of supported platforms. The default URL for the UNIX agents is as follows.

```
http://(Log Trackerserver)/s-doc/UNIX/
```

The administrator can browse the complete software package list from this location, and download the appropriate "tar.gz" file for a specific UNIX platform.

2. On the UNIX platform, the root administrator gunzips and untars the UNIX software, creating a directory such as "/opt/Log Tracker", or "/usr/local/Log Tracker". (The precise location of the software installation is not important.)

3. To install the log file monitor agent, the user edits the "CO-logmon.cnf" file to provide a "DestinationAddress" value, and then starts the "CO-logmon" process, or modifies the "initd" startup system to launch the CO-logmon program on system bootup.

4. The optional file integrity monitor agent is installed in a fashion similar to the above. To install the file integrity monitor agent, the user edits the "CO-fmon.cnf" file to provide a "DestinationAddress" value, optionally modifies the directory specifications of the file, and then starts the "CO-fmon" process, or modifies the "initd" startup system to launch the CO-fmon process on system bootup.

5. At the main Log Tracker Server, the administrator verifies a startup message is logged. The user can then edit the remote configuration of the UNIX agent using the standard Log Tracker "Edit Remote" function (i.e. can drill down on the IP address of the new device, and click "Edit Remote Configuration").

The entire installation steps, outlined above, will usually take just a few minutes to complete. A "root" type login is required.

Future sections will describe in detail the various other features, adaptations, customizations, and applications associated with the Log Tracker UTS system. The reader is encouraged to experiment with the system. In particular, almost all of the information required to understand the essentials of the Log Tracker UTS system has now been explained.

Section 2: Log Tracker UTS Installation

The Log Tracker UTS is usually delivered as an embedded component of the Log Tracker Server, and can be downloaded from the Log Tracker "Home" screen to various UNIX platforms via a web browser. The user can fetch specific "tar.gz" files for various UNIX platforms via a web browser, or push these packages to UNIX platforms using FTP. This provides a simple method of insuring connectivity between the UNIX platform and Log Tracker Server, as well as providing a consistent way of controlling downloads within the enterprise.

Log Tracker UTS is specifically designed not to scatter files into UNIX system directories. All UNIX files reside in a single directory selected by the root administrator. This directory is typically either "/opt/Log Tracker", or "/usr/local/Log Tracker", or some other directory chosen by the installer.

Once the UNIX files are installed on a platform, the root administrator configures and edits the "CO-logmon.cnf" and / or "CO-fmon.cnf" files (provided as part of the distribution) to supply the value for "DestinationAddress", which is the IP address of the Log Tracker Server. The root administrator then launches the "CO-logmon" program and / or "CO-fmon" program as a background process, and configures the program to launch via system startup.

This section provides a detailed discussion of installation requirements and generic procedural steps needed to install any version of the Log Tracker UTS software on a UNIX platform

Log Tracker UTS System Installation Requirements

The Log Tracker UTS programs are non-invasive, and can be installed on a variety of UNIX platforms and operating systems. An “Administrator” login is required to install the software. Specific system requirements of the Log Tracker Server are described below.

- **Disk Space.** The Log Tracker UTS has a small disk footprint of less than 1 Mbytes. The program should generally NOT be installed on a network drive. If possible, the program should be consistently installed in a standard location, either "/opt/Log Tracker" or "/usr/local/Log Tracker".

- **CPU Requirements.** The Log Tracker UTS makes minimal use of CPU, and can co-exist with other server components and applications. The actual CPU requirements will typically be much less than 1%, even under heavy load.

- **TCP Connectivity.** The Log Tracker UTS cannot be installed on a platform that does not have TCP connectivity. (This will typically not be a problem, but may occur in certain evaluation and test scenarios.) The program works with any normal network interface card.

- **Service Ports.** The Log Tracker UTS program requires access to the UDP 514 port of the configured destination host, which may require modifications to firewalls and port blockers. Additionally, the Log Tracker UTS will listen at TCP 55514 for optional remote configuration requests.

Basic Installation Steps

Each UNIX package installed on the Windows system is provided in Gzipped tar file format (i.e. "tar.gz" format) and includes as a file suffix the name of the platform, such as "ut-n-n-n-solaris.tar.gz"

1. Log into the target UNIX platform with a "root" login, run a web browser, and connect to the Log Tracker Server. Download the UTS package specific for UNIX platform from the Log Tracker Server. The following URL will list all the UTS packages, permitting download of any package or UTS documentation.

   ```
   http://(Log Trackerserver)/s-doc/UNIX/
   ```

   The value of (Log Trackerserver) in the above URL should be the location where the Log Tracker UTS software was installed in step 1 above, including any optional HTTP port number.

2. On the UNIX system, copy the "tar.gz" file, downloaded from the Windows platform above, to the directory where the "Log Tracker" directory is to be created, typically either the "/opt" directory, or the "/usr/local" directory. This can be accomplished using a web browser, or using standard binary "ftp".

3. Gunzip the "tar.gz" file, and then extract files using "tar –xvf". This will create the "Log Tracker" directory, which will contain all the UTS files for the platform.

4. Using a text editor, modify the "CO-logmon.cnf" file and specify the IP address of the Log Tracker Server.

   NOTE: This required step is sometimes missed, and the default address of "127.0.0.1" will not work. The destination for Syslog messages MUST be specified as part of the initial configuration.

5. Start the CO-logmon process, and optionally configure the CO-logmon process to start as a background process on the platform. This is typically accomplished by editing the "/etc/rc.local" file, or creating an "/etc/rc.d" startup script, depending upon the particular target operating system.

   NOTE: The CO-logmon process does not fork, and by default runs in foreground. Therefore, the user must specify a "&" character as part of the command invocation, to make this a background process. Failure to add a trailing "&" character to the command invocation may result in delays during node startup.

6. Optionally, repeat steps 4 and 5 above to install the "CO-fmon" file integrity monitor process. Using a text editor, modify the "CO-fmon.cnf" and specify the IP address of the Log Tracker Server, and then start the "CO-fmon" process as a background process. (See notes above for additional information.)

7. When the installation is complete, the CO-logmon and / or CO-fmon programs are installed and running. On startup, these processes each send a single Syslog message to the configured destination host. Check that host to verify a message was correctly sent and received.

No other steps are needed to install and start the UNIX program.

Installation Checkout

The most likely problem with the installation that a user may experience will be that a firewall or port blocker prevents the CO-logmon program from sending

Syslog messages across the network. This can be tested and verified as described here.

Once the system is running, the user can test the installation using the “sendlog” program, which is included as part of the UTS software. Brief help on how this program is used can be acquired by running the program at a UNIX shell prompt, and typing the command “sendlog –help”, which will show the syntax of the command.

Send an initial Syslog message to verify the Log Tracker Server is listening for messages. At a shell prompt on the Log Tracker platform, type:

```
sendlog (dest)  “First Test Message.” 7 1
```

The value of (dest) is the destination hostname or IP address of the platform running the Syslog receiver software, or the Log Tracker Server, entered into the installation dialog. This should cause a message from the platform to appear in the “Syslog” section of the web interface. The facility will be user(1), and the severity will be debug(7).

Section Summary And Additional Notes

1. The UNIX root administrator can directly FTP the UNIX UTS package from the Windows platform to the UNIX box. (The method outlined above, employing the web browser to distribute the software, is just one of various possible methods of obtaining the specific agent software.)

2. The UTS package contains the "CO-logmon" Log File Monitor process, which continuously monitors streaming log files. Additionally, the UTS contains the "CO-fmon" File Integrity Monitor process, which checks for changes to the file system. The user can install either or both of these processes as described in this section.

3. The administrator must modify the "CO-logmon.cnf" file and specify the IP address of the Log Tracker Server as the "DestinationAddress" value. This required step is sometimes skipped and the default address of "127.0.0.1" will not work. The destination for Syslog messages MUST be specified as part of the initial configuration

4. When launching the CO-logmon process, the user must include a "&" character as part of the command invocation. The CO-logmon process does not fork, and by default runs in foreground. Therefore, the user must specify a "&" character as part of the command invocation, to make this a background process. Failure to add a trailing "&" character to the command invocation may result in delays during node startup.

5. The details on how to launch the CO-logmon process on node bootup are system specific. UNIX root administrators should consult the documentation that comes with their system for best practices. Typically, the administrator will either modify the "/etc/rc.local" file, or will create a startup file in the "/etc/rcN.d" directory of their platform.

6. The "CO-fmon" process has its own configuration file and must be configured separately from the CO-logmon process. The "DestinationAddress" value must be configured, and the program configured to launch as a background process on node startup.

7. Neither the CO-logmon nor CO-fmon process rely on each other. Both processes operate as stand-alone agents and can be installed together, separately, or not at all.

8. To test the connectivity between the UNIX platform and Log Tracker, the administrator can execute the "sendlog" program, which can send an arbitrary message from the UNIX system to the Log Tracker Server.

Section 3: Log Tracker UTS Usage

The Log Tracker UTS program augments the existing Syslog capability of the site and is not required to comprehensively monitor a UNIX platform. Many sites will forego the installation of the UTS software, and rely completely on the native UNIX syslog capability.

At those sites requiring the monitoring of streaming log files, data filtering at the UNIX platform, or special data encryption functions, or the monitoring of file object modifications, the UTS software can be employed to generate Syslog messages

At many sites, the entire usage of the Log Tracker UTS will consist of installing the program (as discussed in the previous section) and then rarely if ever visiting that installation again. The Log Tracker UTS does not require program maintenance, and will not interfere with other system processes. The system configuration file (discussed in the next section of this manual) is ready-to-run and does not require any customization, other than the destination Syslog host supplied by the installation dialog.

This section provides detailed notes on the UTS software options and application notes suitable for use by administrators and developers wishing to extend the UNIX Syslog monitoring capabilities of their organization. The section will be of interest to other users wishing to assess the capabilities of the UTS tools, and Syslog protocol in general.

The CO-logmon Program

The CO-logmon program, normally residing in the "/opt/Log Tracker" or "/usr/local/Log Tracker" directory, executes as a single persistent background process, and as a standard UNIX daemon process. The program can be seen in via a "ps" command. The program is normally configured in the UNIX system to start automatically when the host platform boots.

The CO-logmon program monitors streaming log files. It reads the "CO-logmon.cnf" file to acquire the list of monitored files, and then continuously watches for lines to be appended to these files. When a new line is added to a file, the line is fetched, compared to match patterns, and then sent to the Log Tracker Server (as needed.) Any streaming log file can be monitored.

Additionally, the CO-logmon program can watch for file modification times, and send Syslog messages should specific files be modified or deleted from the system. For example, the CO-logmon program can report when the "passwd", "group", "hosts" or any other text or binary file is modified or deleted.

The destination address for all messages is configured in the “CO-logmon.cnf” file, which is in the same directory as the “CO-logmon” program. This file MUST exist in that location, and is read whenever the CO-logmon program starts. A detailed explanation of this configuration file, including all directives that can be included in the file, is provided in the next section of this manual.

The CO-logmon program creates the CO-logmon.log file in the same directory as the executable program and the configuration file. This log file can contain any errors, and can contain Syslog messages (if so configured). The file is overwritten each time the server starts, and typically has just a very few lines. This log file will not need any maintenance.

CO-logmon Program Features

Although the CO-logmon program is not specifically required for any UNIX platform in order to use the Log Tracker Server, there are various important functions that the CO-logmon agent can perform in an enterprise. In many circumstances, these features can be achieved ONLY through the installation and deployment of the CO-logmon program:

- **Text File Monitoring.** The CO-logmon program can monitor any arbitrary text file for appended lines. This includes any file created by the standard "syslogd" process, as well as transfer logs, apache logs, database logs, and audit logs. This extends the existing Syslog capability of the UNIX platform to include arbitrary log files.

- **File Modification Monitoring.** The CO-logmon program can monitor the timestamps of any file object, including text files, object files, and directories. This provides a simple way to watch for unauthorized changes to system password files, group files, or other security related data.

- **Source Filtering**. The CO-logmon program can extend the filtering of Syslog messages at the source platform, reducing network load. For example, the user can disable certain classes of messages based upon keyword within the message, or can override the facility or severity of messages at the UNIX source (rather than at the Log Tracker Server.)

- **Remote Management.** The CO-logmon program supports the remote management functions of the main Log Tracker Server, so that filters and match patterns can be remotely modified (via the Log Tracker web interface or using the standard "rsmconf.exe" Log Tracker utility.)

- **Remote Import Of Files.** The CO-logmon program supports the Log Tracker "Import" facility, which allows the user to import log files from the UNIX platform into Log Tracker. This simplifies the import process for UNIX platforms.

- **Data Encryption.** The CO-logmon program supports both the native encryption functions of Log Tracker, as well as the AES-256 / TLS encryption for advanced internal security, including remote key exchange with the Log Tracker Server. This provides a higher degree of security for the Syslog transport layer.

One important application of the CO-logmon program is to provide management of homegrown applications and scripts. For example, a UNIX cron job can periodically perform system checks using a simple script (such as whether the "syslogd" process is running) and log the results to a file, which is monitored by CO-logmon and the Log Tracker Server. This specific example prevents a user from shutting down the syslogd or other important process on the system without notification.

The CO-fmon Program

The CO-fmon program complements the operation of the CO-logmon process, but provides a totally different function. The CO-fmon process scans directories of files and sends a syslog message if a file has been added, deleted, or changed. This protects certain system directories against changes, and verifies that no malicious software (or undocumented change) has been added to the system. This class of program is often referred to as a "File Integrity Monitor" (FIM), since it continuously monitors whether the file system has been monitored, thereby assuring the integrity of the file system.

The CO-fmon program reads the "CO-fmon.cnf" file to acquire a list of monitored files, creating an "image" file. Subsequently, each time a listing of files is acquired (which is scheduled to occur hourly, daily, or weekly) the CO-fmon program compares the new file list against the image file and reports changes.

The destination address for all messages is configured in the “CO-fmon.cnf” file, which is in the same directory as the “CO-fmon” program. This file MUST exist in that location, and is read whenever the CO-fmon program starts. A detailed explanation of this configuration file, including all directives that can be included in the file, is provided in the next section of this manual.

CO-fmon Program Features

The CO-fmon program operates in a fashion that is almost identical to the Windows File Integrity Monitor, documented in a separate manual. The Log Tracker FIM is particularly important for PCI-DSS compliance, which requires file integrity monitoring to be implemented at a managed site.

The Log Tracker FIM is designed to support the enterprise requirements for security, with special regard to PCI/DSS (as well as other) security guidelines. The program is easy to install and use, and contains the following specific features and functions.

- **Fast File Scans.** The File Integrity Monitor is designed to monitor large numbers of files quickly and non-intrusively. The program will typically scan 10,000 files within one or two minutes, permitting hourly checks of file integrity.

- **Ability To Monitor Files By Directory.** The File Integrity Monitor is easy to configure, and allows the user to specify files by directory, including the ability to match and exclude files by directory name, file suffix, file prefix, or other keywords. This allows an operator to precisely target special files on the managed system.

- **Ability To Perform File Checksums.** The File Integrity Monitor checks the file creation time, modify time, and file size to determine whether a file has been modified on the system. As an additional feature, the user can enable the calculation of checksums on each file to check whether any single bit in the file has been changed.

- **Remote Configuration Capabilities.** The File Integrity Monitor allows the user to remotely access and adjust (with authentication) the program configuration data, permitting the user to make changes to the file integrity monitor while it is running. Additionally, the user can obtain real-time status from the File Integrity Monitor, to view the remote status and state of the program.

Additional information on the operation of the CO-fmon program can be found in the "Windows File Integrity Monitor User Manual", which describes the Windows version of this program.

Note that, unlike the UNIX FIM, the Windows FIM is documented separately, and is not a standard part of the Windows Tool Set. On Windows platforms, the Windows FIM is downloaded and installed separately.. On UNIX systems, the FIM component is included in all packages and can be installed (or not) depending upon the specific requirements of the user, with no other download to the UNIX platform required.

The Sendlog Program

The Sendlog program, normally residing in the same directory as the CO-logmon program, is a simple but powerful utility that allows a user to send an arbitrary Syslog message of arbitrary severity and facility to a Syslog host. The program does not require the execution of the CO-logmon program, or any other file or program on the UNIX platform. It is completely stand-alone, and can be copied or moved to any location on any system.

The command can be used at a shell prompt, in a UNIX shell script, and can also be launched from other scripting languages such as Perl, PHP, Ruby, and many others.

Sendlog Command Line Arguments

The arguments to the UNIX sendlog program are quite trivial, especially when considered next to various SNMP trap generating programs and utilities. The command requires three, four, or five arguments. The basic syntax of the command is as follows:

```
Sendlog (destname) "(message)" (sevnum) (facilnum)
```

Each argument is explained as follows:

(destname)

The first argument is required, and is the hostname or IP address of the device that is running the Syslog receiver, which must be listening to port 514 at the address. Either an IP address, or an official hostname or alias is required as the first argument.

(message)

The second argument is required, and is the message to send to the remote Syslog server. The message will require double quote marks if the message contains any spaces. The message can be up to 1024

characters long, and should generally not contain any strange characters, to promote readability of the message by end-users.

(sevnum)

The third argument is optional, and is the severity number, ranging from 0=emergency, to 7=debug. If this option is not specified, then the default severity for sending a message is 7=debug. A list of severities is provided at the end of this section.

(facilnum)

The fourth argument is optional, and is the facility number, ranging from 0=kernel to 23=local7. If this option is not specified, then the default facility for sending a message is 1=user. A list of facilities is provided at the end of this section. NOTE that, if this option is specified, the user must also specify a severity number, described above.

Sendlog Command Line Examples

To illustrate the operation, consider the following examples of Sendlog program usage that might be typical in an operations center. These messages might be incorporated into a batch file, or might be the result of some program check that is launched periodically by a UNIX script or cron job.

Sendlog 127.0.0.1 “This is a debug message”

The above command sends a message to the Syslog program running 127.0.0.1. The severity and facility is not specified. These items default to the values of “debug” and “user”.

Sendlog 192.168.1.1 “The system is restarting” 6

The above command sends a message to the Syslog program running 192.16.1.1. The severity is “info”, and the facility is not specified, so defaults to “user”.

Sendlog myhost “Error during file transfer” 3 11

The above command sends a message to the Syslog program running at the “myhost” platform. The severity is “error”, and the facility is “ftp”. The command is fully qualified and uses all the available arguments. Note that the user can specify either an IP address, or an official hostname, as shown here.

As shown above, the severities and facility codes are useful for categorizing the message at the Syslog receiver program. Although these values are expressed by their numeric values, they are easily referenced.

Executing the “sendlog” program with no command line arguments will display a list of all the numeric severity and facility numbers, along with brief notes on usage.

Section Summary And Additional Notes

1. The “Log Tracker Syslog Message Service” monitors UNIX event logs for changes, and sends Syslog messages to the destination host.

2. The destination host address is configured in the CO-logmon.cnf file, which is in the same location as the CO-logmon program, by default the directory “/opt/Log Tracker” or "/usr/local/Log Tracker". (The root administrator determines the precise directory during installation.)

3. The "Log Tracker File Integrity Monitor Service" monitors UNIX file systems for changes to files, and sends Syslog messages to the destination host when changes are detected.

4. The destination host address for the file integrity monitor is configured in the CO-fmon.cnf file, which is in the same location as the CO-fmon program.

5. Configuration files MUST exist in the above directory, and specify a variety of parameters and configuration items detailed in the next section.

6. The sendlog program is a stand-alone executable that permits programmers to send Syslog messages to arbitrary hosts, using arbitrary messages, severities and facility codes.

7. Additional information on the operation of the CO-fmon program can be found in the "Windows File Integrity Monitor User Manual", which describes the Windows version of this program.

8. Unlike the UNIX FIM, the Windows FIM is documented separately, and is not a standard part of the Windows Tool Set. On Windows platforms, the Windows FIM is downloaded and installed separately.

9. On UNIX systems, the FIM component is included in all packages and can be installed (or not) depending upon the specific requirements of the user, with no other download to the UNIX platform required.

Section 4: CO-logmon Config File

The CO-logmon.cnf file contains all the parameters and specifications related to the program's operation. This file is found in the same directory as the CO-logmon program, by default the "/opt/Log Tracker" or "/usr/local/Log Tracker" directory. An example of this file is found in Appendix A of this document.

During installation, this file must be edited by root to specify the location of the Log Tracker Server. If a user wishes to fine-tune the parameters of the Syslog messages, or wishes to monitor a new streaming log files, the file can be edited with a standard text editor, or modified via the remote configuration functions of Log Tracker

If the configuration file changes via a manual edit, the user must stop the CO-logmon service and restart the service. Any errors detected while reading the configuration file are logged to the CO-logmon.log file, in the same directory as the CO-logmon program and CO-logmon.cnf file. If the configuration file is changed via a remote configuration operation, no restart of the CO-logmon program is required.

Configuration Items

In addition to specifying the destination address and port number, the configuration file contains a number of other settings that can be used to specify log files as well as match patterns that set the facility and severities of the various Syslog messages. The CO-logmon.cnf file contains the following sections.

- **Destination Address And Port Number.** The top of the file contains the destination and port number for Syslog messages. These two items are required, and must be manually adjusted during installation, as described previously.

- **Remote Configuration Parameters.** The next section of the file contains information regarding the remote configuration capability of the program, including the type of authentication and optional passkey required to permit remote configuration.

- **Optional Parameters**. Following the above fields, the user can specify ancillary parameters, such as whether encryption is to be used.

- **Log File Specifications.** Following the “Optional Parameters” section are multiple entries that allow the user to specify streaming log files, which can be continuously monitored by the program. The user can configure multiple log files, each with multiple patterns to control multiple facilities and severities, using the “MatchKeyWord” directive.

Each of the above items is explained in more detail within the pages that follow. Refer to the end of this section for a listing of the default CO-logmon.cnf file.

Destination Address And Port Number Directives

The CO-logmon program requires only two directives, which must be configured in the CO-logmon.cnf file. The initial configuration of these two directives is performed by the installation procedure, however the user can modify the values after installation to change the destination of Syslog messages. These directives typically appear at the top of the file. Only the first occurrence of these directives is read. If these directives occur elsewhere in the file, these extraneous directives are ignored. The following two directives are required.

DestinationAddress

This directive should be followed by a an IP address, which corresponds to the location of the Log Tracker Syslog receiver (typically the IP address of the Log Tracker web server.) If this value is invalid, the CO-logmon program will not send Syslog messages.

Destination Port

This directive should be an integer number of 514, which is the standard UDP port number used by Syslog. Generally, this value is provided mainly for reference and cannot be easily changed.

These directives are required at the very top of the file, and cannot be moved to some other section of the file. If there are multiple entries, the last entry recorded in the file for these parameters is used, and the other directives are ignored.

Remote Configuration Parameters

The CO-logmon program supports remote configuration directives by the main Log Tracker Server, or by the "rsmconf.exe" remote configuration utility. Following the Destination Address and Destination Port directives are a series of optional parameters to support this function. The following three directives are optional.

ListenAuthMode

This directive specifies the authentication mode used when processing remote requests. The directive is followed by an integer number between 0 and 3 as follows: 0=No authentication; 1=Authentication by source address; 2=Authentication by passkey; 3=Authentication by both source address and passkey.

ListenPassKey

This directive is the passkey used with remote configuration when the ListenAuthMode is 2 or 3. The value serves as a simple password. (The corresponding password in the Log Tracker Server is found in the System > Parameters tab of the web interface.)

ListenPort

This directive should be the integer number of 55514, which is the TCP port at which the CO-logmon program listens for remote requests. Generally, this value is provided mainly for reference and cannot be easily changed.

If these three directives are commented out or removed from the configuration file, then remote configuration is disabled and only manual configuration of the CO-logmon program is permitted.

Optional Parameters Section

Following the Remote Configuration directives are a series of optional parameters. These can be commented out or deleted. The Optional Directives that can appear in the file are as follows:

MessagePrefix

This is a single word that will prefix any messages sent by the system. If the directive is omitted, the message is not prefixed by any special text. This can be used to distinguish the messages, such as by placing a keyword, or the device name, or the organization, or some other keyword at the very start of any message.

MsgDelayMsecs

This is an integer number ranging from 10 to 5000, indicating the number of milliseconds to wait after sending a message. This is a way of throttling the number of messages that can be sent, ranging from 10 per second to only 12 per minute. This prevents any single Syslog process from flooding a Syslog message receiver. The default value, if this directive is omitted is 100 Msecs.

LogLocal

This value is set to either “True” or “False”. If the value is “True”, then all Syslog messages sent by the CO-logmon program are also logged in the CO-logmon.log file (along with any error messages encountered by the program.) This provides a simple way to verify whether UDP messages are being dropped. Note that the CO-logmon.log file is restarted each time the service is started, hence the file does not grow without bounds. If this directive is omitted, it is interpreted to be “False”.

EncryptData

This value is set to either “True” or “False”. If the value is “True”, then the message data is encrypted before it is transmitted, which is the default. This setting will make the CO-logmon program usable ONLY with the Log Tracker Server. The program WILL NOT operate with any other Syslog server if this value is set to “True”. If this directive is omitted, it is interpreted to be “False”. (See later notes in this section for more details.

Log File Monitor Specifications

Following the Optional Parameters section, are a series of Log File Monitor specifications. The user can configure one or more log file monitors, default facilities and severities, and match patterns that overwrite these defaults. The following directives are supported.

LogFile

This directive indicates the pathname to a streaming text log file on the system. The user can specify the pathname as a relative pathname, with respect to the location of the CO-logmon program, or absolute pathname, using either forward or backward slashes. All the directives that follow, until the next “LogFile” directive, apply to the specified log file. This directive can contain Time Format values, such as "%y", "%m, "%d", to

respectively match the two digit year, two digit month, or the two digit day. For example the file specification "/var/logs/f-%y%m%d.log" can be used to monitor a file with a name such as "f-091231.log".

LogName

This optional directive, if it exists, must follow LogFile directive. It is the name of the log file (or subsystem) that is displayed in the Syslog message. If this value is not provided, the event message is not identified with the log file other than in the message content. The value can be any arbitrary text string of up to 500 characters. It is commonly punctuated with a trailing colon, supplied by the user, such as “Oracle Data:” or “HTTP Log File:”

MaxSizeChange

This optional directive, if it exists, must follow the LogFile directive. It is an integer size, in bytes. If the file increases by this amount of bytes or more during a single 500 msec interval, it will trigger a special message indicating that the file has increased rapidly in size. If this value is not furnished, the value of 10,000 bytes is used. (The value may be increased to 1 Mbyte.) This parameter helps prevent excessive Syslog messages from being generated should a file undergo extremely rapid updates, such as a new file being copied into place. Each log file has its own MaxSizeChange value.

LogStatChange

This optional directive, if it exists, must follow the LogFile directive, and have a value of "disable", "enable". The directive indicates that the monitor agent is not to read the file, but only send a Syslog message (with the "DefaultFacilty" and "DefaultSeverity") should the file modification time change. This is useful for monitoring file objects that are not necessarily log files. The file object specified by "LogFile" can be a directory or any file, including an executable file or configuration file. Note that this directive cannot be used with any "MatchKeyword" expressions. If no "LogStatChange" directive exists, then changes to the file modification times are not monitored.

DefaultFacility

This optional directive may follow the LogFile directive. The value specifies a facility name (or an official facility number between 0 and 23), which identifies the default facility code used in all messages that are logged to the specified file. If this directive is omitted, the default facility is assumed to be “user”.

DefaultSeverity

This optional directive may follow the LogFile directive. The value specifies a severity name (or an official facility number between 0 and y),

which identifies the severity code used in all messages that are logged to the specified. The value of this directive is commonly “disabled” or “-1”, indicating that no message is processed unless in matches one of the “UseSeverity” patterns (described below). If this directive is omitted, the default severity is assumed to be “disabled”.

UseFacility

This directive may follow the “DefaultFacility” directive, and is followed by one or more “MatchKeyWord” directives. This operates identically to the Event Log monitor directive, described previously. The directive is followed by a series of match patterns, any of which will cause the “UseFacility” value to be specified as the message facility. Multiple “UseFacility” directives, each followed by multiple “MatchKeyWord” directives, can be configured.

UseSeverity

This directive is similar to the “UseFacility” directive above, but affects the message severity instead of the facility code. This operates identically to the Event Log monitor directive, described previously. The directive is followed by a series of match patterns, any of which will cause the “UseSeverity” value to be specified as the message facility. Multiple “UseSeverity” directives, each followed by multiple “MatchKeyWord” directives, can be configured.

MatchKeyWord

This directive operates identically to the Event Log monitor directive, discussed previously. The directive nested within a “UseFacility” or “UseSeverity” directive, and specifies a single match keyword, with possible ‘*” or ‘?” wildcards. If a new log file entry matches the specified pattern, then the related severity or facility is used. Multiple patterns can be specified, without limit. The “MatchKeyWord” list is ended by any other directive, so the “MatchKeyWord” directives must all be contiguous within a single “UseFacility” or “UseSeverity” block.

Special Log File Names

The "LogFile" specification permits the user to incorporate a Time Format specification into the file name. This allows Log Tracker to monitor log files whose names change each day. Log Tracker employs standard UNIX type time formatting of file names, where the following symbols have special significance in a file name:

%a	Abbreviated weekday name
%A	Full weekday name
%b	Abbreviated month name
%B	Full month name

%d Day of month as decimal number (01 – 31)
%H Hour in 24-hour format (00 – 23)
%I Hour in 12-hour format (01 – 12)
%j Day of year as decimal number (001 – 366)
%m Month as decimal number (01 – 12)
%M Minute as decimal number (00 – 59)
%U Week of, with Sunday as first day (00 – 51)
%w Weekday as decimal number (0 – 6; Sunday is 0)
%W Week of year, with Monday as first day (00 – 51)
%y Year without century, as decimal number (00 – 99)
%Y Year with century, as decimal number
%z Time-zone name or abbreviation.
%% Percent sign

For example, consider the case where a log is created each night with the month and date, and placed in a folder each night with the name of the specified year. Such a file might be named: /var/logs/2014/ex0620.log". The user can specify this file in the "LogFile" directive as "/var/logs%Y/ex%m%d.log", which will correctly resolve to the correct name without any further adjustments.

The Special “disabled” Severity

There is no explicit “ExcludeKeyWord” type of statement. However, the user can easily exclude any message with a particular content by specifying a “UseSeverity” statement with a severity of “disabled”. This special severity is the highest rank (actually the lowest number) and permits the user to filter or exclude any keyword that matches one of the “MatchKeyWord” directives.

For example, the user can configure a directive such as “UseSeverity disabled” (or “UseSeverity –1”) and then follow this directive with a series of MatchKeyWord values, any of which will exclude a message from the event log, regardless if a match is found elsewhere in the event log specification. The “disabled” keyword can be used only in the configuration file, is given a rank of –1 (below “emergency” = 0) and is taken as the highest severity of the system.

If the "DefaultSeverity" value is set to "disabled", then a message must specifically match one of the "MatchKeyWord" values for it to be sent. This is a way of sending messages by exception, useful for targeting only those messages of interest on the system. By default, the "Security" log, uses this technique to reduce the number of security messages sent to the Log Tracker Server.

Notes On Log File Specifications

As shown above, each monitored log has a "DefaultFacility" and “DefaultSeverity” directive, followed by multiple optional “UseFacility” and

“UseSeverity” statements. Each “UseFacility” and “UseSeverity” statement can have multiple “MatchKeyWord” statements. This provides a simple way to configure facilities and severities for any particular message.

It is quite possible (and even likely) that a message's content might match multiple “UseFacility” or “UseSeverity” statements. In that case, the following rules apply:

If a message matches multiple “UseSeverity” statements, then the severity that is actually used will be the highest severity (actually the lowest number) of any severity matched. For example, if a message matches two “UseSeverity” statements, one of which is “info”, and one of which is “critical”, then the “critical” severity is used in the transmitted message.

Likewise, if a message matches multiple “UseFacility” statements, then the facility with the highest number facility code is used as the facility in the transmitted message. If no facility is matched, but a severity is matched, then the DefaultFacility” is used.

One useful technique, to filter out data that is not important, is to make the “DefaultSeverity” for each log file “disabled”. The default severity is applied ONLY if no other severity specification is found. In this way, only those messages that have assigned severities will be sent as Syslog messages. This reduces the load on the Syslog server, especially if there exist many thousands of log file monitors. The administrator can specifically target a key set of messages using this technique.

The "LogStatChange" directive permits the user to monitor for the existence or modification of any file system object. This directive should not be used with any "UseFacility" or "UseSeverity" values, or any "MatchKeywords". The directive permits the operator to watch for changes to critical file system objects, such as password files, configuration files, or directories. This directive can be used only with the "LogFile" directive.

Finally, note that there is a special and optional “MaxSizeChange” directive associated with log file monitoring. If the log file jumps to a very large value, rather than sending out many Syslog messages, the program sends out a single “File Size Changed” Syslog message to indicate this condition. This prevents a situation where the administrator truncates the file by hand, or copies some other file on top of the monitored file (as may often occur.)

Message Encryption

As a special capability, the CO-logmon program encrypts messages sent to the Log Tracker Server system. The administrator edits the “EncryptData” directive, and sets the value to “"False” in order to disable this function. The encryption

prevents casual snooping of the data by using a block rotating, time-based cipher that is built into both the Log Tracker Server, and the CO-logmon program. There will be no apparent change to the data displayed. However, if the destination address is made some other Syslog server, it will be apparent that the data is actually encrypted.

The encryption provides a fair degree of protection against network sniffers. However, since a single 1024 bit private key is used for all the transmissions, this encryption does not protect against man-in-the-middle type attacks, or replay attacks. This encryption is mainly useful for sending Syslog messages across a public Internet, where casual observers might intercept and observe the message content.

Section Summary And Additional Notes

1. The Log Tracker Log File monitor configuration file resides in the same directory as the CO-logmon executable, and is the CO-logmon.cnf file. By default, this file is located in the "/opt/Log Tracker" or "/usr/local/Log Tracker" directory. (The root administrator selects the precise location during installation.)

2. This file is read on program startup, and contains the name of the destination host, as well as other directives.

3. A user can tailor the file with specific log files, as well as match patterns that filter and set the severities and facilities associated with log messages.

A good way to learn about the configuration items is to experiment with the file, adding directives, and then possibly running the CO-logmon program in foreground. With this technique, a user can quickly target specific messages on the system.

Section 5: CO-fmon Config File

The CO-fmon.cnf file contains all parameters and specifications related to the Unix File Integrity Monitor (FIM). This file is found in the same directory as the CO-fmon program, by default the "/opt/Log Tracker" or "/usr/local/Log Tracker" directory. An example of this file is found in Appendix B of this document.

During installation, this file must be edited by root to specify the location of the Log Tracker Server. The administrator must modify the "DestinationAddress" and "DestinationPort" configuration directives to specify the location of the Log Tracker Server. (The CO-fmon.cnf file does not rely on any information contained in the CO-logmon.cnf file. Both of these directives will usually agree with each other, but can be different depending upon the management strategy of the user.)

Additionally, the user will usually want to tailor the CO-fmon.cnf configuration file to include special folders and match patterns appropriate to the FIM activity. Specifically, the user can add folders to be monitored via the "Directory" configuration item, such as the special applications supported by the UNIX platform.

If the configuration file changes via a manual edit, the user must stop the CO-logmon service and restart the service. If the configuration file is changed via a remote configuration operation, no restart of the CO-fmon program is required.

The CO-fmon.cnf file is similar to the CO-logmon.cnf file, and similar to the same file found in the Windows FIM program, documented in the "Windows File Integrity Monitor" manual, provided elsewhere.

Configuration Items

The CO-fmon.cnf file contains the following sections.

- **Destination Address And Port Number.** The top of the file contains the destination and port number for Syslog messages. These directives are identical to the directives of the CO-logmon.cnf file.

- **Remote Configuration Parameters.** The next section of the file contains information regarding the remote configuration capability of the program, including the type of authentication and optional passkey required to permit remote configuration.

- **Required Parameters**. Following the above fields, the user can specify ancillary parameters, such as the severity of messages, and other values.

- **Directory Specifications.** Following the “Required Parameters” section are multiple entries that list all the files and facilities to be monitored. The user can configure multiple directories, and each directory can contain multiple match patterns and exclude patterns.

Each of the above items is explained in more detail within the pages that follow. Refer to the end of this section for a printout of the default CO-fmon.cnf file.

Destination Address And Port Number Directives

The "CO-fmon" program requires two directives, which must be configured in the CO-fmon.cnf file.

DestinationAddress

This directive should be followed by a an IP address, which corresponds to the location of the Log Tracker Syslog receiver (typically the IP address of the Log Tracker web server.) If this value is invalid, the CO-fmon program will not send Syslog messages.

Destination Port

This directive should be an integer number of 514, which is the standard UDP port number used by Syslog. Generally, this value is provided mainly for reference and cannot be easily changed.

These directives are required at the very top of the file, and cannot be moved to some other section of the file. If there are multiple entries, the last entry recorded in the file for these parameters is used, and the other directives are ignored.

Remote Configuration Parameters

The CO-fmon.exe program supports remote configuration directives by the main Log Tracker Server, or by the "rfmconf.exe" remote configuration utility. Following the Destination Address and Destination Port directives are a series of optional parameters to support this function.

The following three directives are optional.

ListenAuthMode

This directive specifies the authentication mode used when processing remote requests. The directive is followed by an integer number between 0 and 3 as follows: 0=No authentication; 1=Authentication by source address; 2=Authentication by passkey; 3=Authentication by both source address and passkey. The default value is 3.

ListenPassKey

This directive is the passkey used with remote configuration when the ListenAuthMode is 2 or 3. The value serves as a simple password. (The corresponding password in the Log Tracker Server is found in the System > Parameters tab of the web interface.)

ListenPort

This directive should be the integer number of 55515, which is the TCP port at which the CO-fmon program listens for remote requests. Generally, this value is provided mainly for reference and cannot be easily changed.

If these three directives are commented out or removed from the configuration file, then remote configuration is disabled and only manual configuration of the CO-fmon program is permitted.

Required Parameters Section

Following the Remote Configuration directives are a series of required parameters as follows:

Schedule

This is the time at which periodic checks are scheduled: if "hourly", then a check is performed at the start of each hour; if "daily", then a check is performed each day at midnight; if "weekly" then a check is performed on Monday morning at midnight; if "monthly" then a check is performed at midnight at the start of each month. No other settings are valid.

ChangeSeverity

This is the severity of messages sent when a file change is detected. The default value is "warning", but any valid severity can be specified,

including the special "disabled" severity, which disables any notification if a file is changed.

AddSeverity

This is the severity of messages sent when a new file is detected. The default value is "notice", but any valid severity can be specified, including the special "disabled" severity, which disables any notification if a new file is detected.

DeleteSeverity

This is the severity of messages sent when a file deletion is detected. The default value is "notice", but any valid severity can be specified, including the special "disabled" severity, which disables any notification if a file is deleted.

AutoGenImage

This setting is either "True" or "False". The default value is "False", which means that the Image File must be generated manually. If this setting is "True", then the Image File" is replaced with the latest list of files each time that a check is performed (which means that each change is reported once, instead of continuously until a new Image File is created.)

UseChecksum

This setting is either "True" or "False". The default value is "False", which means that changes are detected to files based upon the file creation date, modification date, and / or file size. When set to "True", file checksums are also generated and compared to detect changes. This value can degrade the speed of checks and increase CPU usage, but provides the most reliable way to detect changes to files.

PollDelayMsec

This setting is a numeric value in the range of 1 to 100, which indicates the number of milliseconds to pause after testing each file on the system. The value can be used to reduce the CPU time consumed by the file checks. The default value is 10 milliseconds, which is adequate for most systems. Increasing this value too high will reduce CPU time, but increase the time to perform file checks.

Directory Specifications

Following the required parameters section are the directory specifications, which indicate which directories and files are to be monitored. Up to 50 different directories can be specified.

Each directory on the system is identified along with a series of optional match and exclude patterns, which limit the range of files to be scanned. When a

directory is specified, all sub-directories to that directory are also scanned unless the directory names are specifically excluded. The following directives are supported.

Directory

This directive is followed by the name of a Windows directory, with forward (UNIX style) slashes to delimit subdirectories. The pathname can include an environmental variable. All files in the directory, as well as all files in all subdirectories (unless specifically excluded) are scanned.

MatchPatt

This directive is must be preceded by the "Directory" directive, and specifies a pattern that must be matched in the pathname before the file is monitored. The user can include an arbitrary number of match patterns following each "Directory" directive.

ExclPatt

This directive is must be preceded by the "Directory" directive. The value specifies a pattern that excludes a directory filename from monitoring. For example, the "temp" pattern will defer monitoring any file or pathname that contains the "temp" keyword. The user can include an arbitrary number of exclude patterns following each "Directory" directive. The "ExclPatt" can include file suffixes, such as "*.log", but more typically includes subdirectory names that are excluded from the monitor.

Modifying the Configuration File

The configuration directives are read during startup, and are re-read only if the user uploads a new configuration file to the system, such as via the Log Tracker Server File Integrity Monitor screen. When a configuration file is uploaded, this automatically causes a new Image File to be generated based upon the changes to the file.

Normally, the configuration file is modified only at the Log Tracker Server, using the "File Integrity Monitor" configuration screen. This means that port 55515 must be open to the agent.

Note that if the configuration file is directly modified on the managed system the changes are not read until the next time that the CO-fmon process restarts, and no image file is generated. Therefore, the next time that the system starts, the CO-fmon program will likely report a large number of file additions, deletions, or modifications. In this case, the user should regenerate the image file manually.

Section Summary And Additional Notes

1. The Log Tracker File Integrity Monitor configuration file resides in the same directory as the CO-fmon executable, and is the CO-fmon.cnf file.

2. This file is read on program startup, and contains the name of the destination host, as well as other directives.

3. The file does not need to be modified, and comes ready-to-run. However, a user can tailor the file with directory names, match specifications, exclude specifications, and other parameters.

4. If the configuration file is manually modified directly on the system, the file is read only on service startup, which means the next time the agent starts there will be a large number of changes reported to Log Tracker.

The best way to learn about the configuration items is to experiment with the file, adding directives, and then possibly running the CO-fmon program in foreground (using the “-foreground” option.) With this technique, a user can quickly target specific messages on the system.

Section 6: Remote Configuration

The behavior and operation of the CO-logmon and CO-fmon programs is completely driven by its single configuration file, residing in the same directory as each program with a ".cnf" suffix. Depending on the organizational requirements, it may be necessary to make changes to this file in order to receive particular messages of interest.

The user can manually edit a configuration file, and restart the CO-logmon of CO-fmon program. This requires administrative access to the UNIX platform that is hosting the program and configuration file, and is the most secure way of implementing a change

As a special facility, configuration files can also be remotely accessed, downloaded and uploaded to effect changes in an automated way. This requires various permissions and adaptations described in this section. Specifically, remote configuration capability is limited by the value of the "ListenAuthMode" directive within CO-logmon.cnf and CO-fmon.cnf files, which controls and limits remote request via the source address of the client, or via passkey, or both. The most secure "ListenAuthMode" setting is 3, which requires both a valid passkey, and also the client to be at the same IP address as the destination address.

Remote configuration capabilities of the CO-logmon and CO-fmon programs permit a high degree of flexibility, security, and maintainability of this program. This section will be of interest to system installers, administrators, and operations personnel.

Authentication Of Remote Configuration Requests

Log Tracker agent programs listens for remote configuration requests at special TCP ports. The CO-logmon program listens for requests at port 55514. The CO-fmon program listens for requests at port 55515. This port number is included in the configuration file for the program, but cannot be easily changed. If this port is busy when an agent program starts, or if the "ListenPort" directive is commented out of the file, then no remote configuration is possible.

Three different modes of operation are possible, as determined by the "ListenAuthMode" setting of the configuration file:

- **Auth Mode 0.** Setting the ListenAuthMode to a value of "0" disables authentication of requests. This value should probably never be used except in those very special circumstances where the CO-logmon or CO-fmon program is executing on a detached network where security is not a concern.

- **Auth Mode 1.** Setting the ListenAuthMode to a value of "1" causes authentication of remote configuration requests based upon the IP address of the requesting platform. If this auth mode is used, then any remote configuration request to the agent program that originates on a platform other than the localhost "127.0.0.1" address, or the value of the DestinationAddress directive, is rejected. The requesting program must be at the location that receives the Syslog messages, or on the localhost.

- **Auth Mode 2.** Setting the ListenAuthMode to a value of "2" causes authentication of remote configuration requests solely based upon the configured passkey. The value of the "ListenPassKey" value must agree precisely with the value passed to the "rsmconf.exe" program (discussed below) or the value configured at the Log Tracker Server platform on the "System > Parms" screen. Initially, both of these passkey values are set to the keystring "Default", so no special configuration is required out-of-box.

- **Auth Mode 3.** Setting the ListenAuthMode to a value of "3" causes authentication of remote configuration requests to occur based both on the passkey (used in Auth Mode 2) and the source IP address (used in Auth Mode 1). This is the most secure way of managing the remote configuration process, and is the default out-of-box setting for the CO-logmon program.

The values of "DestinationAddress", "DestinationPort", "ListenAuthMode", "ListenPassKey" and "ListenPort" cannot be change by the remote configuration process. Each of these values can be changed only by manually editing the CO-logmon and / or CO-fmon configuration file. Attempts to modify any of these

values are silently bypassed. This enhances security by ensuring that these values can be changed only by remotely logging into the host platform with an administrative login, editing the configuration file manually, and then restarting the CO-logmon program.

PassKey Configuration

In some circumstances, the best (or only) type of authentication available will be with Auth Mode 2, which is "passkey" authentication. In particular, using a passkey as the sole authentication will be necessary on networks that are using NAT (Network Address Translation) or if the Log Tracker Server is multi-homed, or if tunneling software is being used. In these cases, the destination address for Syslog messages may not be the same as the location of remote configuration requests, making the use of Auth Mode 1 or Auth Mode 3 difficult or impossible.

The passkey is simply a text string of 40 characters or less. The value is case-sensitive, but can contain any printable characters, including spaces. The value is passed as an argument to the "rsmconf.exe" and "rfmconf.exe" programs (discussed below) and also is configured in the Log Tracker Server via the "System > Parms" tab. Because this value is "well-known", it is important to change this value across the enterprise when relying solely on passkey authentication.

In general, for extra security, the "passkey" should be used to supplement the source IP address authentication. There is no "downside" to using passkey authentication other than making firewall issues slightly more complex to troubleshoot.

The passkey is not transmitted across the network in clear text. The value is encrypted, hence is secure from attack by network sniffers. However, the value is in clear text within the CO-logmon.cnf file, hence this file should be protected from unauthorized access (such as by limiting access to the host machine.)

Remote Config Via The Log Tracker Web Interface

To execute a remote configuration operation at the Log Tracker Web interface, the user may first need to enable remote configuration by accessing the "Device Information" screen for the host device. This is accomplished by clicking on the target device hyperlink (found in various locations within Log Tracker, in particular in the "Messages > Device" tab.)

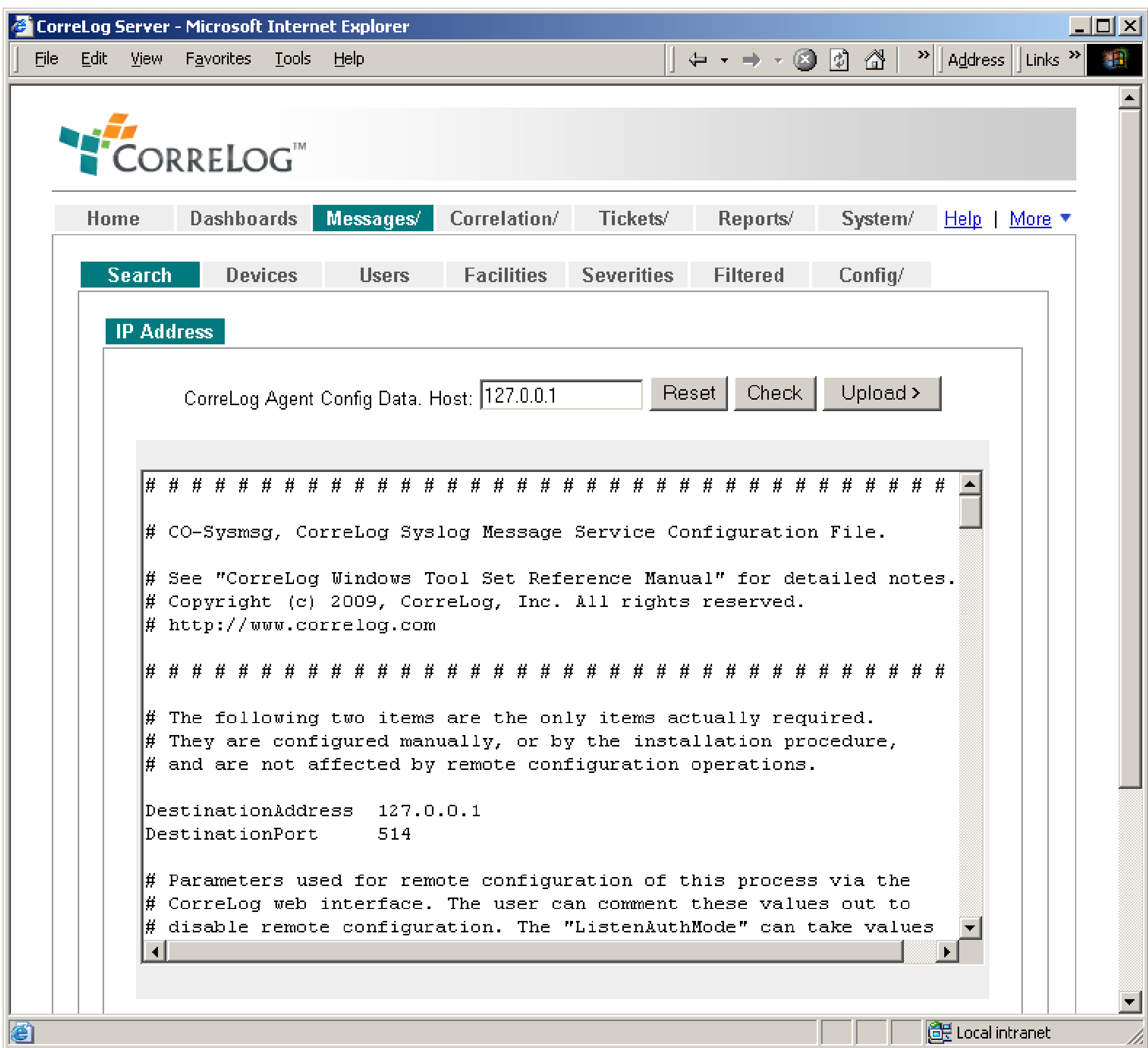
To enable remote configuration, on the Log Tracker "Device Information" screen, the user clicks the "Edit Device Info" hyperlink, and sets the value of "Enable Remote Config Editor" to be "Yes", and commits the data. This causes a new "Edit Remote Config" link to appear on the Device Information screen whenever

it is accessed. Clicking the "Edit Remote Config" hyperlink will download the remote configuration, permit editing, and uploading of this data.

Normally, remote configuration of both the CO-logmon and CO-fmon programs is enabled when Log Tracker receives a message from these agent programs, so this configuration step is needed ONLY if the first message (i.e. the CO-logmon or CO-fmon startup message) is missed by Log Tracker.

When the user clicks on the "Edit Remote Config" hyperlink, the Log Tracker "Remote Configuration Editor" screen downloads the remote configuration and displays an edit dialog that permits the user to modify and upload the data back to the agent process.

A typical Remote Configuration Editor screen is depicted below:



As shown above, the user makes changes to the remote configuration, can check the information, or upload the information back to the remotely executing agent program.

Clicking the "Upload" button automatically initiates a check of the data. The data cannot be uploaded if any errors are encountered. The user can also manually check the data before uploading via the "Check" button.

Note that it is possible to download the CO-logmon.cnf file from one platform, and upload the file to another platform, affording a simple method of copying the configuration of one CO-logmon.cnf to another. The user simply changes the target IP address to be that of the receiving platform before clicking the "Upload" button. The specified platform must be executing the CO-logmon program, and must be configured to permit remote configuration.

CO-logmon Configuration Via The Rsmconf.exe Utility

An alternative to remotely configuring the CO-logmon program via the Log Tracker web interface is to use the "rsmconf.exe" program, which is included in the main Log Tracker Server (within the "system" directory).

This utility permits the user to perform remote configuration at a command line, possibly within a batch file. The "rsmconf.exe" program accepts the following command line arguments:

rsmconf -download ipaddr passkey filename

These command options download the remote configuration file from the specified IP address, using the specified passkey. The resulting configuration data is placed in the specified filename.

rsmconf -upload ipaddr passkey filename

These command options upload the remote configuration file to the specified IP address, using the specified passkey. The data residing in the specified filename is uploaded.

rsmconf -check filename

These command options check the specified filename for errors. (This check is also always performed when using the "-upload" option.) This permits the user to check configuration data that has been previously downloaded. No passkey or IP address is specified or required.

Note that, if the ListenAuthMode of the CO-logmon process is set to a value of 1 or 3, the rsmconf.exe program can only be executed on the platform specified by the DestinationAddress. If the ListenAuthMode is set to 2 or 3, then the passkey must be correctly specified, otherwise it is ignored.

The "rsmconf.exe" program is especially useful in performing batch configure operations, where the command is repeated multiple times within a UNIX ".bat" file, needed to effect reconfiguration on many different platforms.

CO-fmon Configuration Via The Rfmconf.exe Utility

As with the CO-logmon agent process, a command line utility is provided with Log Tracker to remotely configure the CO-fmon process. The "rfmconf.exe" program is included in the main Log Tracker Server (within the "system" directory).

This utility permits the user to perform remote configuration of the CO-fmon process at a command line, possibly within a batch file. The "rfmconf.exe" program accepts arguments similar to the "rsmconf.exe" program, but with slightly different options. The following command line arguments are supported.

This utility permits the user to perform remote configuration at a command line, possibly within a batch file. The "rfmconf.exe" program accepts the following command line arguments:

rfmconf -download ipaddr passkey filename

These command options download the remote configuration file from the specified IP address, using the specified passkey. The resulting configuration data is placed in the specified filename.

rfmconf -upload ipaddr passkey filename

These command options upload the remote configuration file to the specified IP address, using the specified passkey. The data residing in the specified filename is uploaded.

rfmconf -generate ipaddr passkey

These command options cause the system to immediately generate a new Image File, based upon the current configuration file. The program returns within one second with an "OK" indication or a "Busy" indication. If "Busy", then the FMON Agent is busy generating a new listing and the request is deferred.

rfmconf -diff ipaddr passkey

These command options cause the system to immediately generate a new Status Change File, based upon the current configuration file, including the sending of any Syslog messages to the Log Tracker Server. The program returns within one second with an "OK" indication or a "Busy" indication. If "Busy", then the FMON Agent is busy generating a new listing and the request is deferred.

rfmconf -getimg ipaddr passkey

These command options cause the system to return with the image list, to standard output. This allows the remote user to inspect the contents of the FMON Agent Image File..

rfmconf -getdiff ipaddr passkey

These command options cause the system to return with the change list, to standard output. This allows the remote user to inspect the results of the last comparison on the system, and determine which files have been added, removed, or changed.

rfmconf -status ipaddr passkey

These command options cause the system to return with the current status of the FMON Agent, to standard output. These values are similar to the values depicted on the File Integrity Monitor screen, discussed previously.

Note that, if the ListenAuthMode of the CO-fmon.exe process is set to a value of 1 or 3, the rfmconf.exe program can only be executed on the platform specified by the DestinationAddress. If the ListenAuthMode is set to 2 or 3, then the passkey must be correctly specified, otherwise it is ignored.

The "rfmconf.exe" program is especially useful in performing batch configure operations, where the command is repeated multiple times within a Windows ".bat" file, needed to effect reconfiguration on many different platforms.

Other Security Features

Any attempt made to access the CO-logmon program without proper authenticated credentials causes the CO-logmon program to transmit a Syslog message to the destination address. The occurrence is also logged in the program error log (which resides in the same location as the executable, with a ".log" suffix.) The resulting Syslog message indicates the time of the error, and the client IP address. This can be used to monitor unauthorized access.

Additionally, successful remote reconfiguration is also logged, providing an audit trail of changes to the remote configuration data. These messages cannot be disabled.

Section Summary And Additional Notes

1. The remote configuration capability of the CO-logmon program increases program maintainability by permitting administrators to access, modify, and upload configuration changes.

2. The CO-logmon and CO-fmon programs authenticate remote configuration requests by IP address, passkey or both. These values

cannot be changed by the remote configuration process, but must be manually set in the configuration file.

3. The user can perform the remote configuration via the Log Tracker web interface by first enabling remote configuration on the "Device Information" screen.

4. Remote configuration of the CO-logmon and CO-fmon programs is automatically enabled when Log Tracker receives a startup message from these agent programs. If the startup message is not received, the administrator should enable remote configuration as described above.

5. Both the CO-logmon and CO-fmon programs have separate configuration screens that operate in a similar manner, but which perform different purposes.

6. The user can make changes to the CO-logmon and CO-fmon programs via the Log Tracker web interface, including directly editing the configuration files of remote agent programs while they are executing.

7. The user can execute remote configuration of the CO-logmon program via the "rsmconf.exe" program, which is a command line utility program that can download, upload, and check remote configuration data.

8. The user can execute remote configuration of the CO-fmon program via the "rfmconf.exe" program, which is similar to the "rsmconf.exe" program described above (but which works with the CO-fmon program and not the CO-logmon program.)

9. The remote configuration process provides a secure method of accessing and maintaining the remote configurations of multiple CO-logmon programs on the network.

Appendix A: The CO-logmon.cnf File

This appendix provides an example of the CO-logmon.cnf file, which is the central configuration file used by the Log Tracker Syslog Message Service. An administrator or system developer can edit this file to specify the facility and severity codes used by the Event Log monitor. The file also allows users to monitor arbitrary streaming log files on the system (that is, any file which is continuously appended, such as Oracle error logs, HTTP server logs, and many other types of log files.)

The CO-logmon.cnf file is documented in detail within Section 4 of this manual. As stated in that section, the configuration file does not necessarily EVER have to be modified by a user. The default configuration, created by the installation utility, is adequate for many (perhaps most) environments. However, if the user wishes to create a highly customized installation, targeting specific types of event log messages, that capability readily exists through the directives in the file.

This file resides in the same directory as the CO-logmon program (which corresponds to the Log Tracker Syslog Message UNIX Service.) The file provided here is the default configuration that comes with the system.

```
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #

# CO-Logmon, Log Tracker Log Monitor Message Service Configuration
File.
# (Linux Version)

# See "Log Tracker Unix Tool Set Reference Manual" for detailed notes.
# Copyright (c) 2014, Log Tracker, Inc. All rights reserved.
# http://www.Log Tracker.com

# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #

# The following two items are the only items actually required.
# They are configured manually, or by the installation procedure,
# and are not affected by remote configuration operations.

# The location of the Log Tracker Server (or other syslog host) must
# be configured below. The value must be properly configured by the
# administrator

DestinationAddress  127.0.0.1
DestinationPort     514

# Parameters used for remote configuration of this process via the
# Log Tracker web interface. The user can comment these values out to
# disable remote configuration. The "ListenAuthMode" can take values
# 0=No Auth, 1=Source Address, 2=PassKey, 3=Address and Key. These
# values cannot be changed via remote configuration.

ListenAuthMode      0
ListenPassKey       Default
ListenPort          55514

# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #

# The next section provides a list of filenames, match keywords and
# the facility and severity of the resulting syslog message. The
# following default values can be augmented or modified.

LogFile             /var/log/auth.log
LogName             Auth
MaxSizeChange       10000
DefaultFacility     auth
DefaultSeverity     info
UseSeverity         error
MatchKeyWord        fail

LogFile             /etc/passwd
LogName             Password File:
LogStatChange       enabled
DefaultFacility     audit
DefaultSeverity     warning

LogFile             /etc/group
LogName             Groups File:
LogStatChange       enabled
```

```
DefaultFacility     audit
DefaultSeverity     warning

LogFile             /etc/hosts
LogName             System Host File:
LogStatChange       enabled
DefaultFacility     audit
DefaultSeverity     warning

# Additional directives can be appended.
# See "Log Tracker Unix Tool Set Reference Manual" for detailed notes.

# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #

# END OF FILE
```

Appendix B: The CO-fmon.cnf File

This appendix provides an example of the CO-fmon.cnf file, which is the central configuration file used by the Log Tracker File Integrity Monitor service. An administrator or system developer can edit this file to specify the directories and parameters used by the File Integrity Monitor.

The CO-fmon.cnf file is documented in detail within Section 5 of this manual. As stated in that section, the configuration file does not necessarily EVER have to be modified by a user. The default configuration, created by the installation utility, is adequate for many (perhaps most) environments. However, if the user wishes to create a highly customized installation, targeting specific types of event log messages, that capability readily exists through the directives in the file.

This file resides in the same directory as the CO-fmon program. The file provided here is the default Windows configuration that comes with the system. The actual file depends upon the particular package that is installed, and varies between UNIX target systems.

```
# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #

# FMON - Log Tracker File integrity Monitor, Configuration File.

# See "Log Tracker File Integrity Monitor Manual" for detailed notes.
# Copyright (c) 2014, Log Tracker, Inc. All rights reserved.
# http://www.Log Tracker.com

# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #

# The following two items are the only items actually required.
# They are configured manually, or by the installation procedure,
# and are not affected by remote configuration operations.

DestinationAddress  192.168.1.100
DestinationPort     514

# Parameters used for remote configuration of this process via the
# Log Tracker web interface. The user can comment these values out to
# disable remote configuration. The "ListenAuthMode" can take values
# 0=No Auth, 1=Source Address, 2=PassKey, 3=Address and Key.

ListenAuthMode      3
ListenPassKey       Default
ListenPort          55515

# General Parameters

Schedule            hourly
ChangeSeverity      warning
AddSeverity         notice
DeleteSeverity      notice
AutoGenImage        True
UseChecksum         False
PollDelayMsec       1

# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #

# Directory Monitor parameters.

Directory           /bin
MatchPatt           *
ExclPatt            temp

# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #

Directory           /etc
MatchPatt           *.cnf
ExclPatt            temp

# # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #

# Up to 50 directories may be added.

# END OF FILE
```

Appendix C: Facilities And Severities

This section provides a reference guide explaining the Syslog Facility and Severity codes used with the “sendlog” program, and within the CO-logmon.cnf configuration file. Information on the Syslog protocol, as well as more information on the various facilities and severities, can be found in the Log Tracker Server Users manual. The information provided in this appendix is briefer, suitable as a quick reference.

Syslog Facility Codes And Their Meaning

The basic facilities, defined by RFC 3164, are discussed below.

Kernel	0	These are messages related to the Unix kernel process, or generated by very low-level driver software and system programs.
User	1	These are typically user-defined messages. This facility is used (and over-used) as a central way of defining messages that have not been otherwise classified.
Mail	2	These are messages related to the SMTP system, Microsoft Exchange, as well as mail relay systems, and sometimes e-mail programs.
System	3	This is another catchall type of facility that is often over-used, but generally related to system services, Unix daemons not otherwise classified. It can also indicate
Security	4	These are messages related to security processing, such as login detection, virus protection, and intrusion detection systems. Other security related messages are found in the “Audit(13)” and “Alert(14)” facilities.
Internal	5	Originally, these were messages related only to the internal operations of the Syslog protocol process, but have evolved to include general internal processes, often related to performance monitoring, but occasionally simply the internal workings of the system.
Printer	6	These messages are related to the Unix “lpd” process, and can also indicate problems with printer hardware, printer queues, and other types of queues that are not particularly related to printers.
News	7	These are messages related to the Network News processes, which have been fairly deprecated, although are still found on many Unix and mainframe systems. This facility is sometimes used to indicate low severity news events, such as a system being brought down.
Network	8	These messages are related to the Unix “uucpd” process. (UUCP is an acronym for “Unix to Unix Copy”.) They may also refer to network events, such as interfaces being enabled or disabled.

Lock	9	This facility is listed in the RFC as “clock”, but is often renamed as “lock”, and used for locking mechanisms, such as file locking queues It is often substituted for the “Clock(15)” facility code. It may (on some systems) be identical to the Clock(15) facility.
Auth	19	This is similar to the “Security(4)” facility, but is generally reserved for authorization errors, such as invalid logins. It is somewhat synonymous with both “Security(4)” and “Audit(13)”. It represents one of the areas of the RFC that is not clearly delineated, hence is subject to interpretation.
FTP	11	These messages are related to the Unix “ftpd” process, and FTP program, which is somewhat deprecated but still in use. This facility is sometimes used for non-FTP protocol messages related to file transfers.
NTP	12	These messages are related to the Unix “ntpd” (News Transport Protocol) processes. This is somewhat deprecated, but can still be found on a variety of Unix platforms.
Audit	13	This is similar to the “Security(4) and “Auth(19)” facility codes, but mainly appropriate for audit processing, including performance monitoring. For example, a performance monitor might use this facility to periodically send the disk space and disk utilization statistics to the Syslog process for data collection. The messages that use this facility should be pertinent to performance reporting.
Alert	14	This is a general-purpose (hence heavily overused) facility to indicate an Alert condition. This may be somewhat confusing, because this is really a severity rather than a facility. Ideally, these messages would represent problems with the alerting process rather than actual alerts.
Clock	15	These messages are related to the Unix clock daemons, and other processes involved with time synchronization and maintenance. This facility is also sometimes used to mark event times, such as by issuing a Syslog message via the Unix “cron” or UNIX “at” program. Some scheduler programs use this facility. Occasionally, due to ambiguities in the RFC, this facility is confused with, and substituted for the “Lock(9)” facility.

Local0	16	This is a user definable facility, used by Cisco and many other vendors. It is of the used in application software, and is an ideal candidate for being modified by the Log Tracker Server to provide a more meaningful facility name, based upon the message content.
Local1	17	This is another user definable facility. See notes regarding the Local0(16) facility.
Local2	18	This is another user definable facility. See notes regarding the Local0(16) facility.
Local3	19	This is another user definable facility. See notes regarding the Local0(16) facility.
Local4	20	This is another user definable facility. In particular, this is commonly used by RedHat clustering software, and is used by the Cisco PIX software, and is used in some Perl scripts. See notes regarding the Local0(16) facility for more information.
Local5	21	This is another user definable facility. See notes regarding the Local0(16) facility.
Local6	22	This is another user definable facility. See notes regarding the Local0(16) facility.
Local7	23	This is another user definable facility. See notes regarding the Local0(16) facility.

Syslog Severity Codes And Their Meaning

The basic severities, defined by RFC 3164, are discussed below.

Debug	7	The lowest severity, reserved strictly for debugging the system. In practice, debug messages can be totally ignored by everyone.
Info	6	These are informational messages, which can be reviewed later (having some pertinence) but which can be operationally ignored because they have no effect on management activities.
Notice	5	These are messages that are sent with the intention of being noticed. They have a fairly significant level of importance. A filter should generally not remove arbitrarily remove all messages with this severity.
Warning	4	A significant message. It signifies a non-trivial degree of risk. There may not be any corrective action needed with this type of message.
Error	3	A highly significant message. The message indicates that corrective action, manual intervention, or operational change is necessary.
Critical	2	A critical situation exists that requires immediate attention. All other activities should be set aside and the problem addressed as soon as possible. Possible risk to security or data or infrastructure is eminent.
Alert	1	An extremely critical condition exists that will require immediate intervention by the highest levels of management, requiring whatever resources necessary to immediately fix. Data has been lost, security has been breached, or infrastructure has been damaged.
Emergency	0	This severity should NEVER be used, reserved for situations where human safety, or the over-all health of the organization has been compromised or is in extreme jeopardy.

Appendix D: Package Selection

This section provides a reference guide for selecting UNIX packages for specific systems.

Each Log Tracker package is designed to be as portable as possible across a wide variety of different UNIX operating systems. For example, the Linux package will operate on a variety of implementations, requiring only the "libc6" library, found in all modern Linux systems. In general, packages are written to support a specific machine architecture and base operating system (for example X86, Itanium, RISC, SPARC, etc.)

The tables below provide specific examples of the types of machines supported by the various UNIX distribution packages. The lists below are intended only for basic reference, and are not intended to be a definitive list of operating systems supported by the agent programs.

Log Tracker Package Name	Package Description	Supported Operating Systems
ut-linux.tar.gz	Log Tracker Agent for all Linux X86 Systems.	CentOS 4.x CentOS 5.x Debian 4.0 32 bit Debian 4.0 64 bit FreeBSD 6.1 NetBSD 3.1 OpenBSD 3.5 OpenSuSE 10.2 OpenSuSE 11 Fedora 9 Fedora 10 Fedora 11 Fedora 12 Red Hat Enterprise Linux 3 Red Hat Enterprise Linux 4 Red Hat Enterprise Linux 4 Red Hat Enterprise Linux 5 Red Hat Enterprise Linux 5.1 RedHat 5.1 RedHat 6.1 RedHat 6.2 RedHat 7 Redhat Linux 7.1 RedHat Linux 7.2 RedHat 8 RedHat 9 SuSE Linux 8.1 SUSE 9 SP2 SuSE 9.3 SuSE 10, 64-bit SuSE Linux Enterprise Server 10 SuSE Linux Enterprise Server 10 SP1 SuSE Linux Enterprise Server 11 (32-bit) Ubuntu Server

ut-solx86.tar.gz	Log Tracker Agent for all Sun Solaris X86 Systems.	
ut-solaris.tar.gz	Log Tracker Agent for all Sun Solaris SPARC	
ut-hpux.tar.gz	Log Tracker Agent for HP-UX RISC Systems	HP-UX 11.00 (32-Bit) HP-UX 11.00 (64-Bit) HP-UX 11i (32-Bit & 64-Bit) HP-UX 11i (11.11) HP-UX 11i2 (11.23) HP-UX 11i2 (11.23) v3 HP-UX 11iV3 (11.31) HP-UX 11i 64-bit
ut-hpux-ia64.tar.gz	Log Tracker Agent for HP-UX IA-64 Itanium Systems	HP Itanium 64 Bit (HP-UX) HP-UX 11i2 HP 11.23 (HP11i v2)
ut-aix.tar.gz	Log Tracker Agent for IBM AIX Systems.	AIX 4.3.3 (64-bit) AIX 5.1 (64-bit) AIX 5.2 AIX 5.2 (32-bit) AIX 5.2 (64-bit) AIX 5.3 AIX 5.3 (64-bit) AIX 5.2 (32-bit) AIX 6.1

For Additional Help…

Detailed specifications regarding the Log Tracker Server, add-on components, and resources are available from our corporate website. Test software may be downloaded for immediate evaluation. Additionally, NNT is pleased to support proof-of-concepts, and provide technology proposals and demonstrations on request.

NNT, New Net Technologies, Ltd, has produced software and framework components used successfully by hundreds of government and private operations worldwide. We deliver security information and event management (SIEM) software, combined with deep correlation functions, and advanced security solutions.

We are committed to advancing and redefining the state-of-art of system management, using open and standards-based protocols and methods. Visit our website today for more information.

NNT, New Net Technologies, Ltd.
http://www.newnettechnologies.com/

Alphabetical Index

A

B

C

D

E

F

G

H

I

K

L

M

N

O

P

R

S

T

U

W

Y

NNT, New Net Technologies, Ltd.
http://www.newnettechnologies.com/