Heureka Endpoint
Installation & Configuration Guide

CONTENTS

Introduction

Document Formatting

In this installation guide there are commands and text which can be typed and will show up in a monospaced typeface like this:

```
Hello There!
```

Sometimes these commands are very long and can wrap into the next line. When this happens, a continuation character will be used to denote that the command has wrapped and does not actually contain two lines. For example:

```
This is a long line that is wider than this page and therefore will
↳look like two lines when it is really just one long line
```

Auto Update

The endpoint software includes an auto-updating service that reaches out to an update service to obtain the latest version. The initial install can take some time due to it reaching out to the update service and downloading the update. The log file can be viewed to see download progress during this process.

For upgrades that cannot occur via auto-updates, new installers will be provided.

Deployment Considerations

An initial crawl and index of a filesystem can be resource intensive. The indexer has properties to control throttling of resource usage to avoid disrupting the workflow of users. It is strongly recommended to deploy to a small representative set of users or test machines to allow for tuning of the indexing settings before performing a larger deployment. Here are some of the settings that can be configured to ensure a successful deployment:

- Changing the CPU max and number of threads used by the indexer.
- Excluding paths or file types.
- Configuring antivirus programs to not scan temporary files created by the indexer during the indexing process.

Please note that one an endpoint is registered with an appliance, a file called `agent.local.config` is created. This file is loaded *last*, which means it overrides any values in the `agent.config` file. So after deployment, it is best to modify values in `agent.local.config` to ensure they are not overridden.

For a list of all the configurable parameters, please see the appendix.

Files Provided

The following files will be provided for the endpoint installation:

- An auto-updater configuration file: `updater.config`
- An indexer configuration file: `agent.config`
- Platform-specific installers:
  - For Windows
    - 32-bit: `endpoint-installer-<version>-x32.msi`
    - 64-bit: `endpoint-installer-<version>-x64.msi`
  - For Mac: `endpoint-installer-<version>.pkg`
  - For Linux
    - Deb-based: `endpoint-installer-<version>.deb`
    - RHEL rpm-based: `endpoint-installer-rhel-<version>.rpm`
    - Fedora rpm-based: `endpoint-installer-fedora-<version>.rpm`

The basic install process for all platforms is to run the platform-specific installer and then copy the configuration files into the configuration directory within the service's installation folder.

Pre-install checklist

Before starting an installation ensure that you have the following:

- Administrative access to the machine.
- Configuration files and installer downloaded to the machine
- Network connectivity to the URLs and ports listed in the `agent.config` file and the `updater.config` file so that the endpoint can communicate with backend services
  - This may require firewall changes

Installation Options

The `agent.config` file contains properties used by the Endpoint service. It may be desirable to change some of the options, for example the crawl path that the indexer should traverse. By default, the crawl path is set to `C:/Users` on Windows, `/Users` on Mac, and `/home` on Linux. To change this path, modify the value of the `endpoint.indexer.crawlPaths` property in the `agent.config` file. This should be configured before copying the config file to its destination, so that indexing does not start in the wrong location.

This property is a comma-separated list of directories to crawl. An example property is present in the provided config file:

```
# endpoint.indexer.crawlPaths=...
```

This line can be uncommented (remove the # from the front of the line) and modified, or a new one can be added on its own line anywhere in the file.

Customizing the Crawl Path

Below are examples of the `endpoint.indexer.crawlPaths` property setting in the `agent.config` file for different crawl paths on different operating systems. Please note that a symbolic link cannot be used as a crawl path and the agent does not transverse links.

Windows Crawl Path Examples

Windows paths must use forward slashes `/`, not backslashes.

Index the whole C drive:

```
endpoint.indexer.crawlPaths=C:/
```

Index the folder My Stuff in John Doe's Desktop:

```
endpoint.indexer.crawlPaths=C:/Users/John.Doe/Desktop/My Stuff
```

Index both the C and D drives:

```
endpoint.indexer.crawlPaths=C:/,D:/
```

Mac Crawl Path Examples

Index the whole machine:

```
endpoint.indexer.crawlPaths=/
```

Index the folder My Stuff in John Doe's desktop:

```
endpoint.indexer.crawlPaths=/Users/johndoe/Desktop/My Stuff
```

Index the folders Folder 1 and Folder 2 on John Doe's desktop:

```
endpoint.indexer.crawlPaths=/Users/johndoe/Desktop/Folder
↳1,/Users/johndoe/Desktop/Folder 2
```

Linux Crawl Path Examples

Index the whole machine:

```
endpoint.indexer.crawlPaths=/
```

Index the folder My Stuff in John Doe's desktop:

```
endpoint.indexer.crawlPaths=/home/johndoe/Desktop/My Stuff
```

Crawl Path Exclusions

Patterns can be used to exclude paths from being indexed. The property in `agent.config` that controls this is `endpoint.indexer.exclusions`. It is a comma-separated list of regex patterns to exclude, that takes format `regex:<regex_pattern_here>`. It is recommended to prefix the pattern with ^ and end with a $ to ensure matches are going against a full path.

Please note, exclusions will not take effect for files that are located inside of containers.

Examples

Exclude the folders `/home/johndoe/private stuff` and `/home/johndoe/blacklisted.xlsx`:

```
endpoint.indexer.exclusions=regex:^/home/johndoe/private
↳ stuff$,regex:^/home/johndoe/blacklisted.xlsx$
```

Windows requires some special escaping in regular expressions to handle backslashes. The general rule is to replace and backslash with four backslashes. Here is an example to exclude the folder `C:\Users\John.Doe\Desktop\Private Stuff` and the file `C:\temp\testing.xlsx` from being crawled:

```
endpoint.indexer.exclusions=regex:^C:\\\\Users\\\\John.Doe\\\\Desktop\\

↳ \\Private Stuff$,regex:^C:\\\\temp\\\\testing.xlsx$
```

You can use other regular expression syntax to do things such as exclude the `ignore` folder in any users desktop:

```
endpoint.indexer.exclusions=regex:^C:\\\\Users\\\\.*\\\\Desktop\\\\
↳ ignore$
```

Example excluding all `.xlsx` Excel files, note the `"."` part of the file extension needs to be escaped because in regular expressions a period means any character whereas an escaped period is a literal period.

```
endpoint.indexer.exclusions=regex:^.*\.xlsx$
```

Additional Information

If you need more information on how to write regular expressions in the properties file, please see https://docs.oracle.com/javase/8/docs/api/java/util/regex/Pattern.html.

The configuration files follow the standard specifications for Java properties files. The specification for this format can be found at https://docs.oracle.com/cd/E23095_01/Platform.93/ATGProgGuide/html/s0204propertiesfileformat01.html.

For a list of all the configurable parameters, please see the appendix.

Windows

Installation Steps

1. Download the files.
2. Install the 32-bit `.msi` or 64-bit `.msi` based on the endpoint's Windows version.
3. If prompted for allowing the app to make changes to the devices, select "Yes".
4. Move the configuration files to this directory `C:/Program Files/Indexing Service/conf`.
5. Verify the install is complete by checking that a service named "Indexing Service" has appeared in the Services list (Start - run - `services.msc`) and is running. The machine should appear in the Endpoints page of the Interrogate web application.

If the machine does not appear in Interrogate contact a Heureka representative for support steps.

Uninstallation Steps

1. Uninstall using the same `.msi` file that was used to install.
2. Delete the Endpoint from the Interrogate web application.

Log Files

Log files for the Endpoint service are present in `C:\Program Files\Indexing Service\log`.

Mac

Installation Steps

1. Open the terminal: Applications > Utilities > Terminal.

2. Run the provided installer file

```
sudo installer -verboseR -pkg endpoint-installer-<version>.pkg
↳-target /.
```

3. Move the configuration files into the location used by the Endpoint service `/usr/local/opt/heureka-agent/conf/`.

   - Via terminal (replace the sources with the real paths on the machine):

```
sudo mv ~/path/to/agent.config /usr/local/opt/heureka-agent/conf/
↳agent.config

sudo mv ~/path/to/updater.config /usr/local/opt/heureka-agent/conf/
↳updater.config
```

   - Or by launching Finder from inside this directory open `/usr/local/opt/heureka-agent/conf`, then dragging-and-dropping `agent.config` and `updater.config`.

4. Verify the install is complete by checking that the machine appears in the Endpoints page of the Interrogate web application.

Uninstallation Steps

1. Run the following commands from a terminal:

```
sudo launchctl stop com.heurekasoftware.agent

sudo launchctl unload /Library/LaunchDaemons/com.heurekasoftware.
↳agent.plist

sudo rm -rf /usr/local/opt/heureka-agent

sudo rm -rf /Library/LaunchDaemons/com.heurekasoftware.agent.plist

sudo rm -rf /var/log/com.heurekasoftware.agent*
```

2. Delete the Endpoint from the Interrogate web application.

Log Files

Log files for the Endpoint service are present in `/var/log/com.heurekasoftware.agent/`.

Linux

Update the system

It is highly recommended to update to the latest packages before installation. This is distribution and package-manager specific but here are some general commands that should work for most installs:

Updating a debian-based installation:

```
sudo apt-get update

sudo apt-get upgrade
```

Updating an rpm-based distribution:

```
sudo yum update
```

Installation Steps

1. Open a terminal.
2. Install the package.
   - For rhel rpm-based installs

     ```
     sudo rpm -Uv --nodeps endpoint-installer-rhel-<version>.rpm
     ```

   - For fedora rpm-based installs

     ```
     sudo rpm -Uv --nodeps endpoint-installer-fedora-<version>.rpm
     ```

   - For debian-based installs

     ```
     sudo dpkg -i endpoint-installer-<version>.deb
     ```

3. Put the configuration files here `/var/lib/indexing-service/conf`
   - For example (replace the paths with the real paths on the machine):

     ```
     sudo mv ~/path/to/agent.config /var/lib/indexing-service/conf/
     ↳agent.config

     sudo mv ~/path/to/updater.config /var/lib/indexing-service/conf/
     ↳updater.config
     ```

4. The hosts init system must be configured to auto-restart the service.
   - For `systemd` follow the instructions found in "Systemd Configuration".
   - For Upstart follow the instructions found in "Upstart Configuration".
   - For SysV `init` follow the instructions found in "SysV init Configuration".

- For other init systems follow their documentation on setting up an auto-restarting service.

Systemd Configuration

To configure the service to start automatically when the system boots up:

```
sudo /bin/systemctl daemon-reload

sudo /bin/systemctl enable indexing-service.service
```

Start the service.

```
sudo systemctl start indexing-service
```

Upstart Configuration

Start the service:

```
sudo initctl start indexing-service
```

SysV init Configuration

For deb-based installs use the update-rc.d command to configure the service to run when the system boots up:

```
sudo update-rc.d indexing-service defaults 95 10
```

For rpm-based installs use the chkconfig command to configure the service to run when the system boots up:

```
sudo chkconfig --add indexing-service
```

Add a `respawn` line for the service at the bottom of `/etc/inittab`. For example:

```
is:2345:respawn:/etc/init.d/indexing-service start
```

Reboot the server for the `/etc/inittab` changes to take effect.

Uninstallation Steps

1. Uninstall the service, this may take up to 90 seconds for the service to stop.
   - For rpm-based distributions

     ```
     sudo rpm -e indexing-service
     ```

   - For debian-based distributions

     ```
     sudo dpkg --remove indexing-service
     ```

2. Delete the Endpoint from the Interrogate web application.

Log Files

Logs for `systemd` systems are viewable with `journalctl`.

Logs for non-`systemd` systems are in `/var/log/indexing-service.log*`.

Note: For `systemd` based systems the endpoint installer makes some configuration changes to `journald`. View the file at `/usr/lib/systemd/journald.conf.d/10-indexing-service.conf` for details. If these settings are undesirable consult the `journald` documentation found at https://www.freedesktop.org/software/systemd/man/journald.conf.html to change them, though this could result in a loss of logs that may be required for supporting the indexing service.

`journalctl` examples

Follow the log:

```
journalctl -f -u indexing-service
```

View the past 1 hour of logs:

```
journalctl --since '1 hour ago' -u indexing-service
```

For more information on `journalctl` review the documentation online or view the man pages `man journalctl`.

High Performance Configuration

For large indexes, such as indexing large network shares, there are some configuration options which can greatly enhance performance. We provide a preconfigured virtual machine called the High-Performance Indexing Appliance which already has PostgreSQL and the endpoint software installed with all the performance features enabled. If you wish to configure your own agent, the following is a guide to assist you.

Metadata Database Performance

By default, the endpoint software used a Java based relational database called HyperSQL to store file metadata. Although this database is very performant, it is not as performant as standalone database. For increased performance, the endpoint software also supports using PostgreSQL. If you don't already have a PostgreSQL database available to use, you can install it locally on the endpoint. We recommend you create a new database in PostgreSQL specifically for use with the endpoint. Please visit https://www.postgresql.org for details on how to download, install, and configure PostgreSQL.

To configure the endpoint to use a PostgreSQL server, you need to set three properties in the `agent.config` file. The following properties assume you have a PostgreSQL server installed locally, with a database named "`heureka`", user named "`endpoint`", and a password of "`12345`" listening on port `5432`.

```
endpoint.jdbc.user=endpoint
endpoint.jdbc.password=12345
endpoint.jdbc.url=jdbc:postgresql://localhost:5432/heureka
```

If you are switching from a HyperSQL database that already has metadata to PostgreSQL or vice versa, you will have to completely remove the existing index by deleting the index directory and allowing the endpoint to recreate the entire index to ensure the index is in sync with the metadata.

Multi-Threaded Search

By default, the endpoint software uses a single thread to search the index. For large indexes, enabling multiple threads to search the index increases performance. The following `agent.config` property configures the endpoint software to use four threads to search. After any configuration change, please restart the indexing service to allow it to take effect.

```
endpoint.index.search.threads=4
```

Multi-Threaded Indexing

By default, the endpoint software uses two threads when crawling the filesystem to create or update the index. The following `agent.config` property configures the endpoint software to use four threads to create an index:

```
endpoint.indexer.threads=4
```

The number of threads to use for indexing or searching should be a function of the number of available cores on a machine. If you have an endpoint with 8 physical cores, setting threads to `8` would ensure all cores to be used. If the cores support hyperthreading, you may see a further performance increase setting the threads to `16`.

Sharded Index

As an index gets larger, it tends to degrade in search performance. One way to help with this is instead of having one large index, break the index up into smaller indexes called shards. The following `agent.config` property configures the endpoint software to use 2 shards:

```
endpoint.index.shards=2
```

Shards can be increased over time but realize that increasing the shard count on an existing index will not break up the existing index. For example, if the original index has 100,000 files in it and the shard count is set to two, you will have one shard with 100,000 files in a second shard with no entries. As new files are indexed, they will be placed in the second shard until the shards are about even. For best performance, set a shard count and remove the index the shards can be built up with an even distribution.

Also note that lowering the shard count does not remove shards. If you have two shards and want to move to one, you have to delete the index and let it rebuild.

Optimal Shard Count

We recommend one at least shard per 2 TB of data indexed. For example, if you have a network share that is 5 TB, then three shards should suffice. However, there isn't anything stopping you from creating more shards if you wish to account for future growth but realize that there is an overhead to managing shards and collating search results from shards. Creating a 100 shard index for 1 TB of files will probably degrade performance.

Memory Configuration

By default, the endpoint software is configured to use a maximum of 1 GB of memory. To configure the endpoint software maximum memory usage, you have to modify the file that specifies the indexing service options. This file is located:

- On Linux: `/var/lib/indexing-service/bin/indexing-service`
- On Windows: `C:\Program Files\Indexing Service\winsw\indexing-service.xml`
- On macOS: `/Library/LaunchDaemons/com.heurekasoftware.agent.plist`

Open the file for your operating and look for the line that contains the `-Xmx` option. For example:

```
-Xmx1024m
```

Change the number of MB to match the maximum amount of memory you wish to use. You can also specify memory using GB. For example, the following specifies that a maximum of 4 GB will be used:

```
-Xmx4g
```

CPU Configuration

By default, the endpoint software is configured to only allow each thread to consume 50% of the CPU they are executing on. The following `agent.config` property configures the endpoint software to allow threads to use 100% of the CPU:

```
endpoint.system.cpu.max=1.0
```

H5 Integration

H5 is a software product that specializes in classifying file content. For our customers who use H5, they typically use Heureka to do the initial indexing and classification, submit interesting files to H5 for further classification, export the resultant tags from H5, and then import them into Heureka. To facilitate a more streamlined process, Heureka has collaborated with H5 to provide a means for Heureka endpoints to use H5 classifiers directly. This means that H5 tagging can be natural result of Heureka's indexing and classification.

Installation

The first step to installing the H5 Classification Engine on the endpoint is to work with H5 support to ask them to export your classifiers into a bundled application. Ask them to export them into version 4 format for your target operating system. A zip file will be provided to you (for example, `H5CE.zip`) that contains:

- The H5Classifier program and supporting files for your operating system
- A directory containing all your classifiers

One you have this zip file you can perform the installation.

1. Extract the `H5CE.zip` file in the Agent Home directory

   The files will be extracted into the H5CE subdirectory. The Agent Home directory can be found at:
   - On Linux: `/var/lib/indexing-service`
   - On Windows: `C:/Program Files/Indexing Service`
   - On macOS: `/usr/local/opt/heureka-agent`

2. If on Linux or Mac, make the H5Classifier program executable

   ```
   chmod 755 H5CE/H5Classifier
   ```

3. Enable the library

   Add the following to `conf/agent.local.config` (or the `conf/agent.config` is there is no local file):

   ```
   endpoint.h5.classification.enabled=true
   ```

Once these steps are complete, restart the indexing service and Heureka will pick up and use H5 classifiers for all files that are classified from that point forward. Please note that the endpoint will not go back and reclassify files that have already been classified. If you need all files reclassified, then you will have to delete the index and allow the endpoint to rebuild it.

Appendix

Agent Properties

The following are configuration properties which can be set in the `agent.config` and `agent.local.config` files. Be very careful when modifying the values in this file because it can drastically change the functionality and performance of the endpoint software.

Important: The `agent.config` file is loaded first and those property values are overridden by properties found in the `agent.local.config` file. When setting these properties, it is best to make the change in `agent.local.config` to ensure to doesn't get overridden.

Property	Description
`endpoint.api.client.key`	The client key that is required to authenticate with the api. This will be provided for you.
`endpoint.api.connect.timeout`	The api call connect timeout in milliseconds. This defaults to `10000` (10 seconds).
`endpoint.api.read.timeout`	The api call read timeout in milliseconds. This defaults to `30000` (30 seconds).
`endpoint.api.url`	The endpoint api endpoint to use in the format of `https://<address>:<port>`.
`endpoint.api.write.timeout`	The api call write timeout in milliseconds. This defaults to `10000` (10 seconds).
`endpoint.enabled`	Set to `true` if the agent should start.
`endpoint.h5.classification.enabled`	Set to true to enable H5 Classification (requires external program).
`endpoint.id`	The unique id that this endpoint was given upon registration. This is automatically generated and should not be modified.
`endpoint.index.deleteRetentionDays`	The number of days a deleted file should be retained in the index.
`endpoint.index.deleteRetentionEpoch`	The deleted date to use for deleted files with no delete date.
`endpoint.index.deleteRetentionSchedule`	The cron schedule for the job that prunes deleted files from the index.

`endpoint.index.search.threads`	The number of threads to use when searching the index. The default is `1`.
`endpoint.index.shards`	The number of shards to use when indexing files. The default is `1`.
`endpoint.indexer.crawlPaths`	A comma-separated list of paths to crawl.
`endpoint.indexer.exclusions`	A comma-separated list of globs to exclude.
`endpoint.indexer.followLinks`	Set to `true` to allow indexer to follow links. **Warning**: There is no loop detection logic in the endpoint software. Enabling this feature may result in an index that never completes. Please enable with great caution.
`endpoint.indexer.schedule`	The Quartz style cron schedule that determines indexing time. The default schedule is "`0 0 21 * * ?`" which runs at 8:00 PM UTC. Please visit https://www.freeformatter.com/cron-expression-generator-quartz.html to help generate a valid cron schedule.
`endpoint.indexer.threads`	The number of threads to use when indexing. Defaults to `2`.
`endpoint.jdbc.password`	The JDBC password to use for the local metadata database.
`endpoint.jdbc.url`	The JDBC URL to use for the local metadata database. In format of `jdbc:postgresql://<address>:<port>/<database name>`. Currently, only PostgreSQL is supported as an alternative to the default HSQLDB.
`endpoint.jdbc.user`	The JDBC user to use for the local metadata database.

`endpoint.log.level`	The log level to use for the endpoint logs. Logging levels are `ERROR`, `WARN`, `INFO`, `DEBUG`, and `TRACE`. The default log level is `INFO`.
`endpoint.name`	Manually sets the agents host name.
`endpoint.registration.key`	The registration key to use on first run. This will be provided for you.
`endpoint.system.cpu.max`	The amount of cpu for each thread to use where `1.0` means 100% of the CPU and `0.5` means 50% of the CPU. Defaults to `0.5`.
`endpoint.upload.size.limit`	Upload size limit in bytes. The default is 104857600 (100 MB). A value of -1 means no limit.

Updater Properties

The following are configuration properties which can be set in the `updater.config` file. Be very careful when modifying the values in this file because it can drastically change the functionality and performance of the endpoint software.

Property	Description
`heureka.api.keystone.url`	The keystone api endpoint to use in the format of `https://<address>:<port>`.
`heureka.api.org.key`	The client key that is required to authenticate with the api. This will be provided for you.
`heureka.api.revision.service.key`	The client key that is required to authenticate with the api. This will be provided for you.
`heureka.api.revision.url`	The revision api endpoint to use in the format of `https://<address>:<port>`.
`updater.log.level`	The log level to use for the update operations. Logging levels are `ERROR`, `WARN`, `INFO`, `DEBUG`, and `TRACE`. The default log level is `INFO`.

Heureka

Heureka is a technical leader in endpoint search, identify and classification software. Our goal is to bring order to unstructured data by identifying risk while helping you realize the value of unstructured data across all endpoints.

Heureka, Inc
PH. 800.310.0981
info@heurekasoftware.com