Intel Cloud Integrity Technology

**Folders**

# Background

An extensible application needs to provide its plugins a way to store and retrieve configuration, data, and other extensions in order to avoid duplication of effort and minimize maintenance costs. This filesystem blueprint describes the layout and default locations of application folders.

# Architecture

To provide a very general facility for plugins to extend Mt Wilson in the future, the folder layout is very minimal.

The application root folder is also called its home directory.

The binaries directory contains executable files for the local platform, for example the script to start and stop the application may be placed there.

The configuration directory contains files that are necessary for initializing the application every time it starts. It contains the logging configuration file, the application configuration file (possibly encrypted), and other files as necessary. The main configuration files should be restricted to application infrastructure needs. Feature-specific configuration files should be stored under a subdirectory per installed feature.

The features directory contains one subdirectory per installed feature. A feature can be attestation, whitelist, asset tags, reports, etc.

The repository directory contains one subdirectory per installed feature. Data files which may grow or change during the routine operation of the software should be stored here.

# Implementation

## Linux

In 1.0 the application has the following default layout:

/etc/intel/cloudsecurity

/opt/intel/cloudsecurity/attestation-service

/opt/intel/cloudsecurity/management-service

/opt/intel/cloudsecurity/whitelist-service

In 3.0 the application has the following default layout:

/opt/mtwilson

/opt/mtwilson/bin

/opt/mtwilson/configuration

/opt/mtwilson/features

/opt/mtwilson/repository

/var/log/mtwilson/mtwilson.log

/var/run/mtwilson.pid

The application must be installed in /opt/{provider} in accordance with Linux Filesystem Hierarchy Specification for add-on software packages. Intel has a registered provider name “intel” so /opt/{provider} becomes /opt/intel under which we can install any of our applications without conflicting with other software providers.

In the past our default behavior was to use Linux standard locations such as /etc for configuration, /var/log for logs, and /var/run for the PID file. However, these locations are not writable by a non-root user without prior setup. Therefore, we need to use locations under /opt/intel/mtwilson when installing as non-root. To avoid confusion, the same new default locations under /opt/intel/mtwilson should also be used when installing as root, however the root user would have the option of using the standard locations by setting corresponding environment variables. When the installer runs as root, it can automatically create and configure these user-provided locations. When the installer runs as non-root and the non-default locations are specified, it must rely on these locations to already exist and if they don’t it must abort immediately.

Therefore in 4.0 Mt Wilson has the following default layout:

/opt/intel/mtwilson

/opt/intel/mtwilson/bin

/opt/intel/mtwilson/configuration

/opt/intel/mtwilson/features

/opt/intel/mtwilson/logs

/opt/intel/mtwilson/repository

/opt/intel/mtwilson/repository/launcher/mtwilson.pid

In version 3.0 the default install directory is /opt/mtwilson instead of /opt/intel/mtwilson. The rest of this document still applies with that alternate path.

The “bin” folder contains the launcher executable which may be a binary or script file.

The “configuration” folder contains the local application configuration.

The “features” folder contains a subfolder for every installed feature. This comprises the main application code.

The “logs” folder contains all Mt Wilson log files.

The “repository” folder contains a subfolder for every installed feature that needs to store local data. The subfolder “init” belongs to the “mtwilson.sh” init script which stores the “mtwilson.pid” file when starting the service.

The administrator can add the /opt/intel/mtwilson/bin folder to their PATH variable (optional).

To find the software from the distributed Linux locations the administrator can create symlinks:

ln -s /opt/intel/mtwilson/bin/mtwilson.sh /usr/bin/mtwilson

ln -s /opt/intel/mtwilson/configuration /etc/opt/intel/mtwilson

ln -s /opt/intel/mtwilson/logs /var/log/opt/intel/mtwilson

ln -s /opt/intel/mtwilson/repository /var/opt/intel/mtwilson

It’s also possible to use traditional Linux locations without defining any variables to override Mt Wilson default locations by moving the folders to the distributed locations and creating symbolic links from the Mt Wilson default location (for example /opt/intel/mtwilson/repository) to the preferred location (for example /var/opt/intel/mtwilson).

mv /opt/intel/mtwilson/bin/mtwilson.sh /usr/bin/mtwilson && ln -s /usr/bin/mtwilson /opt/intel/mtwilson/bin/mtwilson.sh

mv /opt/intel/mtwilson/configuration /etc/opt/intel/mtwilson && ln -s /etc/opt/intel/mtwilson /opt/intel/mtwilson/configuration

mv /opt/intel/mtwilson/repository /var/opt/intel/mtwilson && ln -s /var/opt/intel/mtwilson /opt/intel/mtwilson/repository

If the application has been installed in a location other than /opt/intel/mtwilson, the environment variable MTWILSON\_HOME must be set to inform the launcher where to find the application folders.

For automatic startup, a start script may be created in /etc/init.d/mtwilson and there are variations on this file name and location between different Linux distributions.

## Windows

The application has the following layout:

C:\mtwilson

C:\mtwilson\configuration

C:\mtwilson\features

C:\mtwilson\repository

The main Mtwilson folder contains the launcher executable which may be a binary (mtwilson.exe) or script file (mtwilson.bat). The administrator may add the “mtwilson” folder to the PATH variable or create the symbolic link.

# Configuration

It’s possible to customize the locations of specific folders without creating symbolic links by setting Java system properties or environment variables. If a system property is set, it takes precedence over an environment variable. The default platform-specific values are used only if there is no system property or environment variable to override them.

The following table summarizes the options:

Table 1 Customizable file system locations

|  |  |  |
| --- | --- | --- |
| Folder | System Property | Environment Variable |
| Home | mtwilson.home | MTWILSON\_HOME |
| Configuration | mtwilson.configuration | MTWILSON\_CONFIGURATION |
| Features | mtwilson.features | MTWILSON\_FEATURES |
| Repository | mtwilson.repository | MTWILSON\_REPOSITORY |
| Log file | mtwilson.logfile | MTWILSON\_LOGFILE |
| Process ID file | mtwilson.pidfile | MTWILSON\_PIDFILE |

In general, any folder that is used by the application should be customizable by setting either a system property like “mtwilson.fs.{name}” or an environment variable like “MTWILSON\_FS\_{name}”. This is automatic for designated subfolders of the application. This does not apply to feature subfolders because there could be many identically-named subfolders across features, for example {feature}/java.

If an application derived from Mt Wilson needs to use separate property names or environment variable names, the application can include an application.properties file under src/main/resources/com/intel/mtwilson. The application.properties file can have the following properties:

mtwilson.application.id=mtwilson

mtwilson.application.name=Mt Wilson

mtwilson.configuration.file=mtwilson.properties

mtwilson.environment.prefix=MTWILSON\_

Setting id=”kms” would cause the launcher to look for system properties like kms.home, kms.fs.configuration, etc. using a “kms” prefix instead of a “mtwilson” prefix, and the default value for kms.home on Linux would be /opt/kms and on Windows would be C:\kms. This allows other applications such as Trust Director, KMS, and Trust Agent to reuse the mtwilson-launcher.

There should be at most one application.properties file on the classpath. The Mt Wilson server does not require because the defaults are already for Mt Wilson. Other applications such as the Trust Agent should include application.properties in a jar file which is only included when they are running as the main application. That is, it should not be included in any libraries or features which may be added as dependencies in other projects.

# Clusters

In a cluster deployment, the local configuration is typically just enough to connect to the cluster and the remainder of the configuration is obtained from the cluster database except for items that must be local such as TLS key and certificate

In a cluster deployment, some features may use the cluster database as their repository.