Release Notes for hub SDK of managed integrations for AWS IoT Device Management

1. Version number

v2.2.0

2. Changes between v2.2.0 and v2.1.0

2.1 Notification class support for Z-Wave door lock
2.2 Support for basic info cluster
2.3 Mqtt keep alive made configurable
2.4 Fix for delete Z-Wave device issue caused by missing hyphen in DSK
2.5 Fix for Mqtt proxy component memory issue

3. Supported devices

Refer to the “Supported Zigbee and Z-Wave device types” section of the Managed integrations for AWS IoT Device Management Developer Guide on AWS docs - https://docs.aws.amazon.com/iot-mi/latest/devguide/supported-devices.html

4. Supported functionality

4.1 Hub onboarding
- Device certificate provisioning of hub using Fleet provisioning by claim method
- Functionality to store certificates and associated keys stored in hub’s Secure Storage

4.2 Hub offboarding
- De-register or offboard hub from AWS IoT managed integrations

4.3 Device onboarding
- Simple Setup / Smart Setup / ZTS method of onboarding Z-Wave and Zigbee devices
- User Guided Setup (UGS) method of onboarding Z-Wave and Zigbee devices

4.4 Support for Z-Wave and Zigbee device lifecycle management, including:
- Adding a new device and associating it with the hub
- Deleting a device currently associated with the hub
- Controlling a device
- Reporting control command responses and device event notifications to managed integrations

4.5 Control hub as a device
- Generate C function stubs for controlling attributes/features on the hub. The C function stubs are generated off Matter clusters (OnOff, diagnostics, etc.) which represent the attributes on the hub. These C functions can then be implemented to execute those commands

4.6 Offline control of Zigbee and Z-Wave devices via local application running on hub
- IPC interface that client applications running on the hub can use to send commands to, read attributes, and receive notifications from Z-Wave and Zigbee devices associated with the hub

4.7 Runtime device log configuration, and upload to managed integrations
- Logger daemon on hub that can be configured via managed integrations API (log level, log file size, etc.), as well as be configured to upload hub SDK logs to the managed integrations

4.8 Custom protocol plugin
- The Custom protocol plugin support feature allows customers to bring their own proprietary IoT protocols into the AWS managed integrations ecosystem. Through well-defined SDK interfaces, customers can onboard devices, define capabilities, and handle real-time control flows—all while maintaining full compatibility with AWS managed integrations cloud and hub SDK components.

5. Components

5.1 iotmi_hub_onboarding
5.2 iotmi_mqtt_proxy
5.3 iotmi_lpw_provisioner
5.4 iotmi_device_agent
5.5 iotmi_cdmb
5.6 iotmi_log_daemon
5.7 SDK libraries:
- libiotshd_mqtt_proxy_lib.so
- libiotshd_log_c.so

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

- libSecureStorageCertHandler.so (soft link)
- libSecureStorageCertHandler.so.1 (soft link)
- libSecureStorageCertHandler.so.1.0
- libsqlite.so.3.50.3

5.8 start_hub_sdk.sh (SDK startup script)

6. Supported Toolchains and System Requirements

6.1 Toolchain: aarch64-unknown-linux-gnu
Linux Kernel version: 5.3.18 and above
CPU Architecture: aarch64
Compiler: gcc-10.5.0
glibc: glibc-2.33
binutils: 2.37 and above

6.2 Toolchain: armv7-unknown-linux-gnueabihf
Linux Kernel version: 4.19.294
CPU Architecture: ARM V7 32-bit
Compiler: gcc-10.3.0
glibc: glibc-2.32
binutils:2.36.1

6.3 Toolchain: armv8-unknown-linux-gnueabihf
Linux Kernel version: 4.14.255 and above
CPU Architecture: ARM V8 32-bit
Compiler: gcc-12.3.0
glibc: glibc-2.38
binutils:2.4

7. Documentation

Refer to the “Hub SDK” section of the Managed integrations for AWS IoT Device Management Developer Guide on AWS docs - https://docs.aws.amazon.com/iotmanagedintegrations/latest/devguide/what-is-managedintegrations.html

8. License Information

Refer to the accompanying LICENSE.txt and the THIRD_PARTY_LICENSES files

9. Software Footprint

- Compiled using the linux-x86-64 aarch64-unknown-linux-gnu toolchain
- Default compiler optimization level of -O0 (optimize for compile time and debug-ability)
- RAM as measured by Resident Set Size (RSS) with 2 commands per second (1 Zwave, 1 Zigbee)

9.1 SDK

Component	Flash (Release build) (MB)	RAM (Release build) (MB)
SDK		
iotmi_hub_onboarding	9.9	13.40
iotmi_mqtt_proxy	7.5	16.63
iotmi_lpw_provisioner	8.1	22.51
iotmi_device_agent	13	27.55
iotmi_cdmb	1.1	12.55

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.

iotmi_log_daemon	.435	9.85
libiotshd_mqtt_proxy_lib.so	.514	-
libSecureStorageCertHandler.so.1.0	.066	-
libiotshd_log_c.so	.376	-
SDK Total	**40.991**	**102.49**
Third Party Dependencies		
openssl	-	-
json-c	-	-
sqlite3	-	-
cJSON	-	-
nng	-	-
curl	-	-
aws-sdk-cpp	-	-
aws-iot-device-sdk-cpp-v2	-	-
Third Party Total	**59**	-
SDK + Third Party Total	**99.991**	**102.49**

9.2 Middleware

Component	Flash (Release Build) (MB)	RAM (Release Build) (MB)
Middleware		
zwave_svc	.136	16.1
ace_zigbee_service	.3	8.12
ace_eventmgr / ace common	.032	4.75
ace zwave service shared libs	3.5	-
ace zigbee service shared libs	1.8	-
ace_eventmgr / ace common shared libs	2.1	-
MW Total:	**7.87**	**28.97**

9.3 SDK + Middleware total

Flash (Release build) (MB)	RAM (Release build) (MB)
107.86	**131.46**

Copyright © 2025 Amazon Web Services, Inc. and/or its affiliates. All rights reserved.