



Intel[®] Rack Scale Design Software RMM

User Guide

RMM Reference Stack Build and Debug Software Version 2.1.3

May 2017

Revision 002



No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

Intel disclaims all express and implied warranties, including without limitation, the implied warranties of merchantability, fitness for a particular purpose, and noninfringement, as well as any warranty arising from course of performance, course of dealing, or usage in trade.

This document contains information on products, services, and/or processes in development. All information provided here is subject to change without notice. Contact your Intel representative to obtain the latest forecast, schedule, specifications, and roadmaps.

The products and services described may contain defects or errors known as errata which may cause deviations from published specifications. Current characterized errata are available on request.

Copies of documents that have an order number and are referenced in this document may be obtained by calling 1-800-548-4725 or by visiting <http://www.intel.com/design/literature.htm>.

Intel and the Intel logo are trademarks of Intel Corporation in the United States and other countries.

*Other names and brands may be claimed as the property of others.

Copyright © 2017 Intel Corporation. All rights reserved.



Contents

1	Introduction	5
1.1	Scope.....	5
1.2	Intended audience	5
1.3	Referenced documents.....	5
1.4	Definition of terms	5
1.5	Notes and symbol convention	6
2	RMM Software Build and Installation	7
2.1	Prerequisites	7
2.1.1	Intel® RSD RMM version 2.1.x source code.....	7
2.1.2	Configure server internet access	7
2.1.3	OS package requirements	7
2.1.4	Linked third party libraries.....	7
2.2	Building procedure.....	8
2.2.1	Source Code.....	8
2.2.2	Build Intel® RSD RMM 2.1.x software	8
2.2.3	Adding a signature to .deb files.....	8
2.3	Installation procedure for binary	9
2.4	Uninstallation procedure	10
2.5	CM password encryption.....	10
2.6	System dependencies	10
2.6.1	tftp service	10
2.7	Debugging.....	11
2.7.1	RMM service	11
2.7.2	Memdb	11
2.7.3	Minicom	11
2.7.4	Ipmitool.....	11



Revision History

Revision	Description	Date
002	Maintenance release.	May 12, 2017
001	Initial release.	February 9, 2017

§



1 Introduction

1.1 Scope

This document provides the recommended installation and debug procedures for Intel® Rack Scale Design (Intel® RSD) Rack Management Module (RMM) Software version 2.1.x.

1.2 Intended audience

- Server rack management software vendors who are investigating Intel® RSD RMM API functionalities, such as discovery, composition, and management, in a single- or multi-vendor environment.
- Firmware vendors who are exploring Intel® RSD RMM API as a tool to offer easy firmware adoption/migration.

1.3 Referenced documents

Doc ID	Title	Location
335451	Intel® Rack Scale Design Generic Assets Management Interface API Specification	Intel.com/intelrsd_resources
335452	Intel® Rack Scale Design BIOS & BMC Technical Guide	Intel.com/intelrsd_resources
335501	Intel® Rack Scale Design Architecture Specification	Intel.com/intelrsd_resources
335454	Intel® Rack Scale Design Software Reference Kit Getting Started Guide	Intel.com/intelrsd_resources
335455	Intel® Rack Scale Design Pod Manager API Specification	Intel.com/intelrsd_resources
335456	Intel® Rack Scale Design Pod Manager Release Notes	Intel.com/intelrsd_resources
335457	Intel® Rack Scale Design Pod Manager User Guide	Intel.com/intelrsd_resources
335458	Intel® Rack Scale Design PSME REST API Specification	Intel.com/intelrsd_resources
335459	Intel® Rack Scale Design PSME Release Notes	Intel.com/intelrsd_resources
335460	Intel® Rack Scale Design PSME User Guide	Intel.com/intelrsd_resources
335461	Intel® Rack Scale Design Storage Services API Specification	Intel.com/intelrsd_resources
335462	Intel® Rack Scale Design Rack Management Module (RMM) API Specification	Intel.com/intelrsd_resources
335463	Intel® Rack Scale Design RMM Release Notes	Intel.com/intelrsd_resources
335464	Intel® Rack Scale Design Software RMM User Guide	Intel.com/intelrsd_resources
n/a	Scalable Platforms Management API	http://dmtf.org/standards/edfish

1.4 Definition of terms

Term	Definition
ACL	Access Control List
BMC	Baseboard Management Controller
CIMI	Cloud Infrastructure Management Interface
HTTP	Hypertext Transfer Protocol
JSON	JavaScript Object Notation
Lid	Localization ID
OCCI	Open Cloud Computing Interface
OData	Open Data Protocol
OS	Operating System



Term	Definition
OVF	Open Virtualization Format
Pod	A physical collection of multiple racks
PODM	Pod Manager
RCPM	Rack Control Plane Manager ¹
REST	Representational State Transfer
RMM	Rack Management Module
SDV	Software Development Vehicle
SVN	Subversion Repository Number
TMC	Tray Management Controller
URI	Uniform Resource Identifier
UUID	Universally Unique Identifier
VM	Virtual Machine
XML	Extensible Markup Language

1.5 Notes and symbol convention

Symbol and note convention are similar to typographical conventions used in CIMI specification.

Notation used in JSON serialization description:

- Values in italics indicate data types instead of literal values.
- Characters are appended to items to indicate cardinality:
 - "?" (0 or 1)
 - "*" (0 or more)
 - "+" (1 or more)
- Vertical bars, "|", denote choice. For example, "a|b" means a choice between "a" and "b".
- Parentheses, "(" and ")", are used to indicate the scope of the operators "?", "*", "+" and "|".
- Ellipses (i.e., "...") indicate points of extensibility. Note that the lack of an ellipses does not mean no extensibility point exists, rather it is just not explicitly called out.



¹ RCPM is an equivalent of PODM run at the rack level and managing resources within this rack only.



2 RMM Software Build and Installation

2.1 Prerequisites

2.1.1 Intel® RSD RMM version 2.1.x source code

Refer to the Software Package Contents section of the Intel® RSD RMM Release Notes for the latest posted version of the RMM source code.

2.1.2 Configure server internet access

Intel® RSD RMM installation and debugging procedures may require access to the worldwide web. It is at the user's discretion to setup proper networking, firewall, and proxy configurations.

2.1.3 OS package requirements

The following Linux* OS packages may be required for the Intel® RSD RMM 2.1.x compilation.

Module	Version	Description
Ubuntu*	>12.04	Operating System requirement.
build-essential	N/A	Build tool chain.
Cmake	2.8.12	Make tool.
Automake	1.4.1	Automate part of the compilation process.
Autoconf	2.69	A tool to producing configuration file for compile.

2.1.4 Linked third party libraries

The following opensource libraries have been linked to Intel® RSD RMM 2.1.x source code.

Module	Version	Source Link (suggested)	Description
jansson	2.5	http://www.digip.org/jansson/	Jansson is a C library for encoding, decoding and manipulating JSON data.
openssl*	1.0.1u	http://www.openssl.org	The OpenSSL Project is a collaborative effort to develop a robust, commercial-grade, full-featured, and Open Source toolkit implementing the Secure Sockets Layer (SSL v2/v3) and Transport Layer Security (TLS v1) protocols as well as a full-strength general purpose cryptography library. This product includes software developed by the OpenSSL Project for use in the OpenSSL Toolkit (http://www.openssl.org/) This product includes cryptographic software written by Eric Young (eay@cryptsoft.com). This product includes software written by Tim Hudson (tjh@cryptsoft.com).
libcurl	7.40.0	http://curl.haxx.se	Curl is a command line tool for transferring data specified with URL syntax. libcurl is a C library for transferring data specified with URL syntax. It is readily available to be used by your software.
Flat UI	2.2.1	https://github.com/designmodo/Flat-UI	Flat style theme for web UI.



Module	Version	Source Link (suggested)	Description
zlib	1.2.8	http://www.zlib.net	zlib is designed to be a free, general-purpose, legally unencumbered—that is, not covered by any patents—lossless data-compression library for use on virtually any computer hardware and operating system.
net-snmp	5.7.2.1	http://www.net-snmp.org/about/license.html	Net-SNMP is a suite of applications used to implement SNMP v1, SNMP v2c and SNMP v3 using both IPv4 and IPv6.
Uuid	1.6.2	ftp://ftp.ossdp.org/pkg/lib/uuid/uuid-1.6.2.tar.gz	The libuuid library is used to generate unique identifiers for objects that may be accessible beyond the local system.

2.2 Building procedure

2.2.1 Source Code

The RMM source code can be downloaded from the Intel® RSD Github Site located at <https://github.com/01org/IntelRSD>.

Refer to the Intel® RSD Design Package Content section of the Intel® RSD Customer Release Notes for the Reference Number of the latest posted version of the RMM source code packages.

The user should implement the customized HAL driver code by following the GAMI specification. There is a stub version, for reference only, in the `asset_module` folder to help the user implement their own HAL driver.

2.2.2 Build Intel® RSD RMM 2.1.x software

Combine the RMM reference stack code and RMM HAL driver code together:

1. Decompress the RMM package.
2. Copy or download the third party package to its proper location in `RMM_{version}/src/deps/{third_party_library_name}/` if missing.
Every third party library package should be named as it is in the first line of relevant `CMakeLists.txt`.
Additionally, copy `bootstrap-3.0.3-dist.zip` and `jquery-2.0.0.min.js` to `RMM_{version}/src/deps/flatui/`.
3. Append `ADD_SUBDIRECTORY(asset_module)` in file `RMM_{version}/src/module/CMakeLists.txt` if missing.
4. Start the code compiling process:

```
cd RMM_{version}/utils
./rmm_release.sh
```

5. Access the binaries:

```
RMM_{version}/build/release
```

2.2.3 Adding a signature to .deb files.

To add a signature to `.deb` packages you need to have a GPG key on your system. To confirm that a GPG key is on your system, run the following command:

```
gpg --list-key
```

You may want to create a new GPG key pair using the following command (this may take several minutes):

```
gpg --gen-key
```




To sign .deb packages, use the debsigs tool. If you do not have it on your system, run the following command to install it:

```
sudo apt install debsigs
```

To sign the RMM base package, use the following command:

```
debsigs --sign=origin -k <key ID> rmm-base.deb
```

Repeat the above command for each .deb package.

Follow this guide to exchange the GPG key you used to sign the packages with the recipient:

```
https://www.gnupg.org/gph/en/manual/x56.html
```

2.3 Installation procedure for binary

This procedure is for the user who has a pre-built binary.

First, untar the tarball of binaries:

```
tar -xvf rmm-{version}.tar.gz
```

If the .deb packages have signatures, you may want to verify them before installation. Follow the steps below to do that:

1. Install debsig-verify package if not installed:

```
sudo apt install debsig-verify
```

2. To verify the base RMM package, run the following:

```
sudo debsig-verify rmm-base.deb
```

If the above command fails, you may need to first import the public part of the PGP key pair that was used during package signing:

3. Import the GPG public key from a file:

```
gpg --import <gpg-public-key-file>
```

4. Get the fingerprint of the key:

```
gpg --fingerprint
```

Use the last 8 bytes and remove spaces, for example, if the fingerprint from the above command looks like this:

```
Key fingerprint = AEFD 42F3 1F2E CCA3 19AE 992C EBDE EB78 5B35 B559
```

The fingerprint needed for the next steps is EBDEEB785B35B559.

5. Create a keyring directory:

```
sudo mkdir -p /usr/share/debsig/keyrings/<FINGERPRINT>
```

6. Import the public key to the keyring:

```
sudo gpg --no-default-keyring --keyring  
/usr/share/debsig/keyrings/<FINGERPRINT>/rmm.gpg --import <gpg-public-key>
```

7. Create a directory for the policy document:

```
sudo mkdir -p /etc/debsig/policies/<FINGERPRINT>
```

8. Create an XML policy document. Use the following example:

```
<?xml version="1.0"?>  
<!DOCTYPE Policy SYSTEM "http://www.debian.org/debsig/1.0/policy.dtd">  
<Policy xmlns="http://www.debian.org/debsig/1.0/">
```



```
<Origin Name="rmm" id="EBDEEB785B35B559" Description="RMM package"/>
<Selection>
  <Required Type="origin" File="rmm.gpg" id="EBDEEB785B35B559"/>
</Selection>
<Verification MinOptional="0">
  <Required Type="origin" File="rmm.gpg" id="EBDEEB785B35B559"/>
</Verification>
</Policy>
```

9. Save the file in the policies directory you created under the name rmm.pol. Replace the IDs with the fingerprint of your public key and repeat the step for package signature verification for each RMM package.

10. Install the packages:

```
cd rmm-{version}
sudo dpkg -i *.deb
```

11. Start the RMM service:

```
sudo service rmm start
```

2.4 Uninstallation procedure

1. Stop the RMM service:

```
sudo service rmm stop
```

2. Remove the RMM packages:

```
sudo dpkg --purge rmm-all
sudo dpkg --purge rmm-consolecontrol
sudo dpkg --purge rmm-api
sudo dpkg --purge rmm-base
```

2.5 CM password encryption

1. To encrypt the CM password or username, use the encrypt tool with the format below:

```
$ encrypt <password> <key_file>
```

2. Copy the output string to the RMM configuration file in the "UserName" or "Password" field.

```
/etc/rmm/rmm.cfg
```

3. Copy the generated keyfile to directory /etc/rmm.

- If the key file does not exist, the tool will generate a new one.
- The tool can be built from rmm source code under the directory utils/encrypt_text.
- The key must be 8 characters long.
- The default key file name is "keyfile".

2.6 System dependencies

Installed RMM software requires additional to run some actions properly.

2.6.1 tftp service

For MBP update requests **tftp** service must be up and running. It must accept GET requests.



Configuration must be as follow:

- User: tftp
- Port: 69
- Directory: /srv/tftp.

Server directory must be writable for root user.

2.7 Debugging

2.7.1 RMM service

Check the RMM service status with the following command:

```
sudo service rmm status
```

Access the service log:

```
sudo cat /var/log/{component_name}/logfile
```

2.7.2 Memdb

Dump memdb and access the dumped content with the following commands:

```
sudo dumpmemdb  
sudo cat /var/log/Memdbd/dump.out
```

2.7.3 Minicom

Use minicom to connect to the CM console for debugging:

```
sudo minicom -D /dev/ttyCm2Console
```

Use the "--help" command to list all supported commands (e.g. the CM IP address).

2.7.4 Ipmitool

Use ipmitool to ensure the CM software stack is working correctly.

```
ipmitool -H {cm_ip} -U admin -P admin raw 0x06 0x01
```

