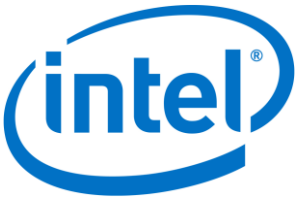

Intel® Rack Scale Design Software RMM

User Guide
RMM Reference Stack Build and Debug

December 2016

Revision 008

Document Number: 332875-008

Contents

Revision History

Revision	Description	Date
1.3.	Update to running package to RMM 1.2.6.17	December 7, 2016
1.2	Update running package to RMM 1.2.6.9	September 6, 2016
1.1	Update running package to RMM 1.2.6.6	August, 3, 2016
1.0	Update running package to RMM 1.2.6.5	June 21, 2016
0.9	Update running package to RMM 1.2.6.3	April 1, 2016
0.8	Update running requirement , aligned with RMM 1.2.5.3	March 3, 2016
0.7	Align with RMM release 1.2.5.1	February 29, 2016
0.6	Add uuid library information, aligned with RMM release 1.2.3.1	December 16, 2015
0.5	Add third party libraries information.	September 2, 2015
0.4	Add CM password encryption.	August 13, 2015
0.3	Fix minor issue and align with RMM 1.1.3 release.	June 26, 2015
0.2	Update code checkout part	June 15, 2015
0.1	Initial version	May 13, 2015

§

1 Introduction

1.1 Scope

This document provides the recommended installation and debug procedures for Intel Rack Scale Design Rack Management Module (RMM) Software version 1.2.6.x

1.2 Intended audience

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

1.3 Referenced documents

Doc ID	Title	Location
332868	Intel® Rack Scale Design GAMI API Specification	*http://intel.com/intelRSD*
332869	Intel® Rack Scale Design Pod Manager REST API Specification	*http://intel.com/intelRSD*
332870	Intel® Rack Scale Design Pod Manager Release Notes	*http://intel.com/intelRSD*
332871	Intel® Rack Scale Design Pod Manager User Guide	*http://intel.com/intelRSD*
332873	Intel® Rack Scale Design PSME REST API Specification	*http://intel.com/intelRSD*
332872	Intel® Rack Scale Design PSME Release Notes	*http://intel.com/intelRSD*
332874	Intel® Rack Scale Design PSME User Guide	*http://intel.com/intelRSD*
332877	Intel® Rack Scale Design RMM REST API Specification	*http://intel.com/intelRSD*
332876	Intel® Rack Scale Design RMM Release Notes	*http://intel.com/intelRSD*
332875	Intel® Rack Scale Design RMM User Guide	*http://intel.com/intelRSD*
332878	Intel® Rack Scale Design Storage Services API Specification	*http://intel.com/intelRSD*
332936	Intel® Rack Scale Design BIOS/BMC Tech Guide	*http://intel.com/intelRSD*
332937	Intel® Rack Scale Design Architectural Requirements Specification	*http://intel.com/intelRSD*
334611	Intel® Rack Scale Design Getting Started Guide	*http://intel.com/intelRSD*
n/a	Scalable Platforms Management API	*http://dmtf.org/standards/redfish*

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
OVF	Open Virtualization Format
POD	A physical collection of multiple racks
PODM	POD Manager

Term	Definition
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® Rack Scale Design RMM version 1.2.6 source code

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

2.1.2 Configure server internet access

Intel Rack Scale Design 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 Intel Rack Scale Design RMM 1.2.6 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 has been linked Intel Rack Scale Design RMM 1.2.6 source code.

Module	Version	Source Link (suggestion only)	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 the library curl is using to do its job. 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 (suggestion only)	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.ossp.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® Rack Scale Design Github Site located at *https://github.com/01org/IntelRSD*.

Please refer to the Intel® Rack Scale Design Package Content section of the Intel® Rack Scale Design Customer Release Notes for the Reference Number of the latest posted version of the RMM source code packages.

The user should implement the cusomized 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® Rack Scale Design RMM 1.2.6 software

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

1. Decompress the RMM package.
2. Append ADD_SUBDIRECTORY(asset_module) in file RMM_{version}/src/module/CMakeLists.txt if missing.

Start the code compiling process:

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

Access the binaries:

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

2.3 Installation procedure

Untar the tarball of the binaries:

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

Install the packages:

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

Starting the RMM service:

```
sudo service rmm start
```

2.4 Uninstallation procedure

Stopping the RMM service:

```
sudo service rmm stop
```

Removing 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

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

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

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

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

And copy the generated keyfile to directory /etc/rmm

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

2.6 Debugging

2.6.1 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
```

2.6.2 Memdb

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

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

2.6.3 Minicom

Use minicom to connect to CM console for debugging:

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

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

2.6.4 Ipmitool

Use ipmitool to ensure CM software stack is working correctly.

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

§