
Resource Manager (RM)

Release Notes

Applies to Product Release: 02.01.00.05
Publication Date: June 20, 2014

Document License

This work is licensed under the Creative Commons Attribution-NoDerivs 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nd/3.0/> or send a letter to Creative Commons, 171 Second Street, Suite 300, San Francisco, California, 94105, USA.

Contributors to this document

Copyright (C) 2012-2014 Texas Instruments Incorporated - <http://www.ti.com/>



Texas Instruments, Incorporated
20450 Century Boulevard
Germantown, MD 20874 USA

Contents

Overview.....	1
Dependencies.....	1
New/Updated Features and Quality.....	1
Resolved Incident Reports (IR).....	3
Known Issues/Limitations.....	4
Licensing.....	4
Delivery Package.....	4
Installation Instructions.....	4
Directory structure.....	4
Customer Documentation List.....	5

RM version 02.01.00.05

Overview

This document provides the release information for the latest Resource Manager (RM) which can be used by applications that want to manage device resources.

RM includes:

- Pre-compiled library for DSP (Big and Little) Endian of RM.
- Source code.
- API reference guide

Dependencies

RM is dependent on following external components delivered in PDK package:

- None

New/Updated Features and Quality

This is an **engineering release**, tested by the development team.

Release 2.1.0.5

- Restricted access to netcp-rx/tx channels for K2L and K2E platforms. **Replace with IR when it is opened.**

Release 2.1.0.4

- Removed duplicated rmServer source code in the test/ directory. The rmServer runs on ARM but, otherwise, is device independent. The rmServer source code has been relocated to the test/armv7/linux/ directory.
- Fixes for issues flagged by Coverity
- See Table 1 for resolved IRs

Release 2.1.0.3

- Sync with 2.0.0.9

Release 2.1.0.2

- Sync with 2.0.0.8

Release 2.1.0.1

- Sync with 2.0.0.6
- K2E and K2L support

Release 2.0.0.9

- Added two new transport APIs:
 - Rm_receiveGetPktServiceSrcName – Returns the RM instance name that originated the service request encapsulated within the RM packet.
 - Rm_receiveGetPktSrcName – Returns the RM instance name that originated the RM packet.
- Modified the Linux rmServer application so that it can be run as a daemon.

Release 2.0.0.8

- Added new resources for user stats counters in PA_LLD: 32bUsrStats and 64bUsrStats

Release 2.0.0.7

- Added shared object library support

Release 2.0.0.6

- See Table 1 for resolved IRs

Release 2.0.0.5

- Add Global resource list and Policy to support Qmss_QueueType_INTC_SET[234]_QUEUE to add all INTC(CIC) accessible queues.
- Service responses now contain the number of times the requesting instance has allocated the resource.
- Update policies and global resource lists to latest linux DTS.
- See Table 1 for resolved IRs

Release 2.0.0.4

- Added example that has Client running on DSP that communicates with a Server running in Linux user-space. Communication between ARM and DSP is over IPC MessageQ. Instructions for running the test application can be found in the RM User Guide: http://processors.wiki.ti.com/index.php/MCSDK_UG_Chapter_Developing_System_Mgmt#Running_RM_Test_Projects
- See Table 1 for resolved IRs

Release 2.0.0.3

- Added high priority accumulation and starvation counter queues to QM2.
- Extensive bug fixes and cleanup of RM source and test code

- Added ability to request a resource's reference count
- Added ability for Client Delegates to locally manage resources
- Added Server process for Linux
- Added Linux process to process test project that exchanges messages between a Client process and the latter Server process
- See list of Resolved IRs

Release 2.0.0.2

- Added Shared Server and Shared Client instance types that use shared memory to communicate internally. No transport code is needed for these instance types to communicate. Intended for use in DSP only, time-critical applications where, non-blocking APIs are required.
 - Includes example showing how to setup and use Shared Server/Client
- Added RM instance cleanup APIs
- Extensive bug fixes and cleanup of RM source

Release 2.0.0.1

- Initial release

Resolved Incident Reports (IR)

Table 1 provides information on IR resolutions incorporated into this release.

Table 1 Resolved IRs for this Release

IR Parent/ Child Number	Severity Level	IR Description
SDOCM00107840	S2 – Major	rmServer file name should be independent of the platform name like rmServer.out and rmServer_so.out
SDOCM00103974	S2 - Major	QMSS Memory Region Insertion: Need the ability to specify if this should be in INTERNAL or EXTERNAL Linking RAM
SDOCM00106374	S2 - Major	K2L: qmss_device.c needs to be enhanced to cover all the QPEND queues exposed by hardware
SDOCM00107944	S2 - Major	PDK 03.01.00.01 -> CPPI LLD 02.01.00.05 -> Device -> K2E -> CPPI_DEVICE.C needs update to support XGE
SDOCM00108233	S2 - Major	Usermode LLD cpptest makes the board to hang and only power cycle recovers the EVM

IR Parent/ Child Number	Severity Level	IR Description
SDOCM00108235	S2 - Major	RM Server failing when running k2h cppiTest.out

Known Issues/Limitations

IR Parent/ Child Number	Severity Level	IR Description

Licensing

Please refer to the SDK licensing document for the details.

Delivery Package

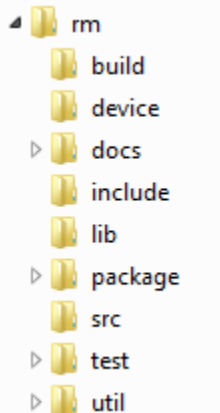
There is no separate delivery package. RM is being delivered as part of PDK.

Installation Instructions

RM is currently bundled as part of Platform Development Kit (PDK). Refer installation instruction to the release notes provided for PDK.

Directory structure

The following is the directory structure after the RM package has been installed:



The following table explains each individual directory:

Directory Name	Description
ti/drv/rm	The top level directory contains the following:-

	<ol style="list-style-type: none"> 1. <u>Environment configuration batch file</u> The file “setupenv.bat” is used to configure the build environment for RM. 2. <u>XDC Build and Package files</u> These files (config.bld, package.xdc etc) are the XDC build files which are used to create the RM package. 3. <u>Exported Driver header files</u> Header files which are provided by RM and should be used by the application developers for customization and usage.
ti/drv/rm/build	The directory contains internal XDC build related files which are used to create the RM package.
ti/drv/rm/device	The directory contains the device specific RM global resource list and policy files.
ti/drv/rm/docs	The directory contains the RM documentation.
ti/drv/rm/include	The “include” directory has private RM header files. These files should not be used by application developers.
ti/drv/rm/lib	The “lib” folder has pre-built Big and Little Endian libraries for RM along with their <u>code/data size information</u> .
ti/drv/rm/package	Internal RM package files.
ti/drv/rm/src	Source code for the RM.
ti/drv/rm/test	The “test” directory in RM has unit test cases which are used by the development team to test RM.
ti/drv/rm/util	The “util” directory has BSD-licensed, open source components used by RM.

Customer Documentation List

Table 2 lists the documents that are accessible through the /docs folder on the product installation CD or in the delivery package.

Table 2 Product Documentation included with this Release

Document #	Document Title	File Name
1	API documentation (generated by Doxygen)	docs/rmDocs.chm