ACKNOWLEDGEMENT

By utilizing this website and/or documentation, I hereby acknowledge as follows:

Effective October 1, 2012, QUALCOMM Incorporated completed a corporate reorganization in which the assets of certain of its businesses and groups, as well as the stock of certain of its direct and indirect subsidiaries, were contributed to Qualcomm Technologies, Inc. (QTI), a wholly-owned subsidiary of QUALCOMM Incorporated that was created for purposes of the reorganization.

Qualcomm Technology Licensing (QTL), the Company's patent licensing business, continues to be operated by QUALCOMM Incorporated, which continues to own the vast majority of the Company's patent portfolio. Substantially all of the Company's products and services businesses, including QCT, as well as substantially all of the Company's engineering, research and development functions, are now operated by QTI and its direct and indirect subsidiaries[1]. Neither QTI nor any of its subsidiaries has any right, power or authority to grant any licenses or other rights under or to any patents owned by QUALCOMM Incorporated.

No use of this website and/or documentation, including but not limited to the downloading of any software, programs, manuals or other materials of any kind or nature whatsoever, and no purchase or use of any products or services, grants any licenses or other rights, of any kind or nature whatsoever, under or to any patents owned by QUALCOMM Incorporated or any of its subsidiaries. A separate patent license or other similar patent-related agreement from QUALCOMM Incorporated is needed to make, have made, use, sell, import and dispose of any products or services that would infringe any patent owned by QUALCOMM Incorporated in the absence of the grant by QUALCOMM Incorporated of a patent license or other applicable rights under such patent.

Any copyright notice referencing QUALCOMM Incorporated, Qualcomm Incorporated, QUALCOMM Inc., Qualcomm Inc., Qualcomm or similar designation, and which is associated with any of the products or services businesses or the engineering, research or development groups which are now operated by QTI and its direct and indirect subsidiaries, should properly reference, and shall be read to reference, QTI.

---

[1] The products and services businesses, and the engineering, research and development groups, which are now operated by QTI and its subsidiaries include, but are not limited to, QCT, Qualcomm Mobile & Computing (QMC), Qualcomm Atheros (QCA), Qualcomm Internet Services (QIS), Qualcomm Government Technologies (QGOV), Corporate Research & Development, Qualcomm Corporate Engineering Services (QCES), Office of the Chief Technology Officer (OCTO), Office of the Chief Scientist (OCS), Corporate Technical Advisory Group, Global Market Development (GMD), Global Business Operations (GBO), Qualcomm Ventures, Qualcomm Life (QLife), Quest, Qualcomm Labs (QLabs), Snaptracs/QCS, Firethorn, Qualcomm MEMS Technologies (QMT), Pixtronix, Qualcomm Innovation Center (QuIC), Qualcomm iSkoot, Qualcomm Poole and Xiam.

MDM9x15/MDM9x25 VSS QMI Service

Application Note

80-N5576-90 B

August 2, 2012

Submit technical questions at:
https://support.cdmatech.com/

Qualcomm Confidential and Proprietary

QUALCOMM Incorporated
5775 Morehouse Drive
San Diego, CA 92121-1714
U.S.A.

Contents

Figures

Tables

MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

Revision history

Revision	Date	Description
A	Jun 2012	Initial release
B	Aug 2012	Added MDM9x25 to applicable targets throughout document

1 Introduction

NOTE: Numerous changes were made to this document. It should be read in its entirety.

1.1 Purpose

This document explains how to to implement a VSS QMI service in MDM9x15/MDM9x25 targets.

1.2 Scope

This document is intended for customers interested in creating VSS QMI service in their MDM9x15/MDM9x25-based devices.

1.3 Conventions

Function declarations, function names, type declarations, and code samples appear in a different font, e.g., #include.

Shading indicates content that has been added or changed in this revision of the document.

1.4 References

Reference documents, which may include Qualcomm documents, standards, and resources, are listed in Table 1-1. Reference documents that are no longer applicable are deleted from this table; therefore, reference numbers may not be sequential.

Table 1-1 Reference documents and standards

Ref.	Document	
Qualcomm		
Q1	*Application Note: Software Glossary for Customers*	CL93-V3077-1
Q2	*Presentation: AMSS 9615 QMI Communications Between Processors*	80-N5576-26
Q3	*Application Note: QMI Vendor-Specific Services with IDL/QCSI/QCCI*	80-N5706-1

1.5 Technical assistance

For assistance or clarification on information in this guide, submit a case to Qualcomm CDMA Technologies at https://support.cdmatech.com/.

If you do not have access to the CDMATech Support Service website, register for access or send email to support.cdmatech@qualcomm.com.

1.6 Acronyms

For definitions of terms and abbreviations, see [Q1].

MAY CONTAIN U.S. AND INTERNATIONAL EXPORT CONTROLLED INFORMATION

2 Background

The IDL, QCCI, and QCSI frameworks are available in MDM9x15/MDM9x25. See [Q2] to create VSS QMI service in MDM9x15/MDM9x25. In this target, QMI service can reside in both Hexagon™ and A5. This document depicts different user scenarios.

3 QMI Communication Inside MDM9x15/MDM9x25

QMI service can run in Hexagon and the QMI client can run in A5 (see Figure 3-1). Also, QMI service can run in A5 and the QMI client can run in Hexagon (see Figure 3-2). QMI communication between A5 and Hexagon is over the IPC router.
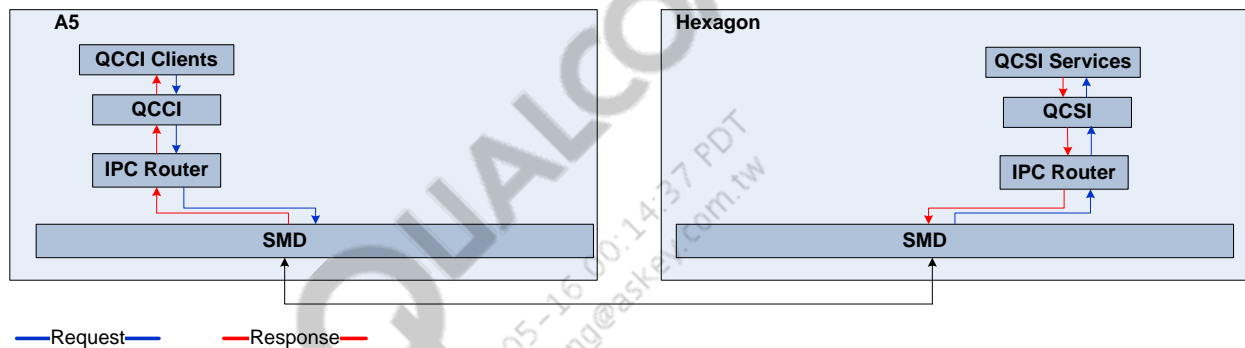

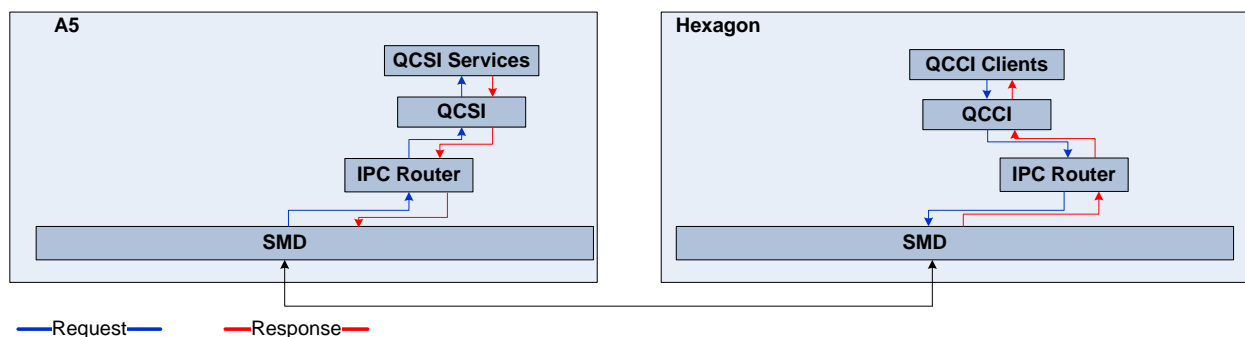**Figure 3-1 QMI service in Hexagon, QMI client in A5**

Figure 3-2 QMI service in A5, QMI client in Hexagon

4 QMI Communication Between External AP and MDM9x15/MDM9x25

It is possible to have the QMI client in an external AP and QMI service in MDM9x15/MDM9x25 A5. Also, the QMI client can be in an external AP and QMI service can be in MDM9x15/MDM9x25 Hexagon.

4.1 QCSI service running in MDM9x15/MDM9x25 Hexagon

Qualcomm does not support an IPC router in an external AP, and all QMI messages are sent/received over Qmux from/to an external AP. To support a QCSI-based QMI messsage, QSAP translates the Qmux-based QMI message into an IPC-routable QMI message. Figure 4-1 shows the QMI communication between an external AP and MDM9x15/MDM9x25 when service is running in Hexagon.
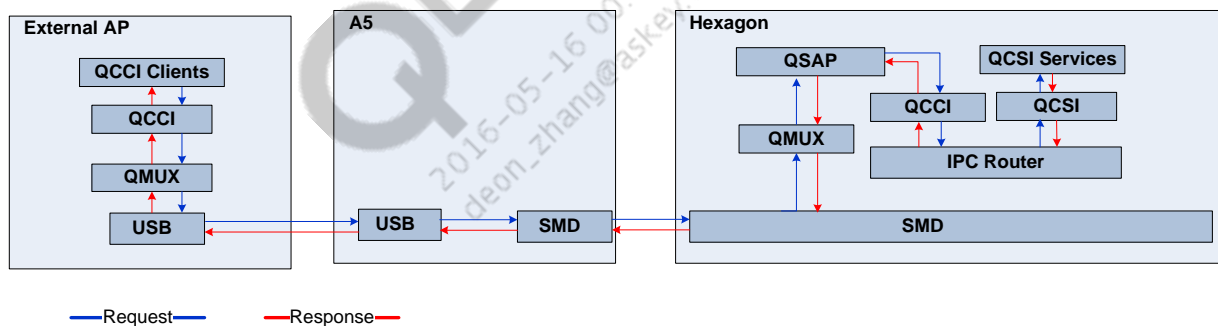

Figure 4-1 QMI communication between external AP and MDM9x15/MDM9x25 – QSCI service

4.2 Legacy QMI service running in MDM9x15/MDM9x25 Hexagon

Legacy QMI service is still available in MDM9x15/MDM9x25. Figure 4-2 shows the QMI communication between an external AP and MDM9x15/MDM9x25 when legacy service is running in Hexagon.
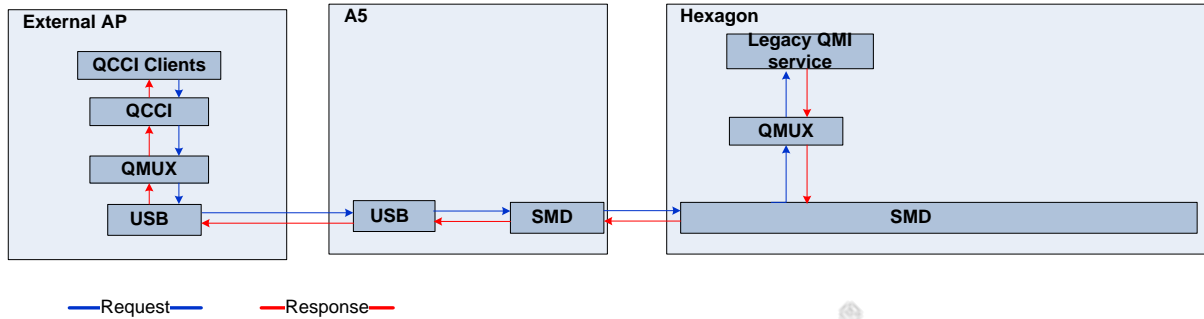
—Request—   —Response—

Figure 4-2 QMI communication between external AP and MDM9x15/MDM9x25 – Legacy QMI service

4.3 Service running in MDM9x15/MDM9x25 A5

When the service is running in A5, the QMI message is sent first to Qmux in Hexagon. Qmux then sends the QMI message to QSAP.

QSAP translates a Qmux-based QMI message into an IPC-routable QMI message via the QCCI API. The message would be routed to A5 over the IPC router. Figure 4-3 shows the QMI communication between an external AP and MDM9x15/MDM9x25 when service is running in A5.
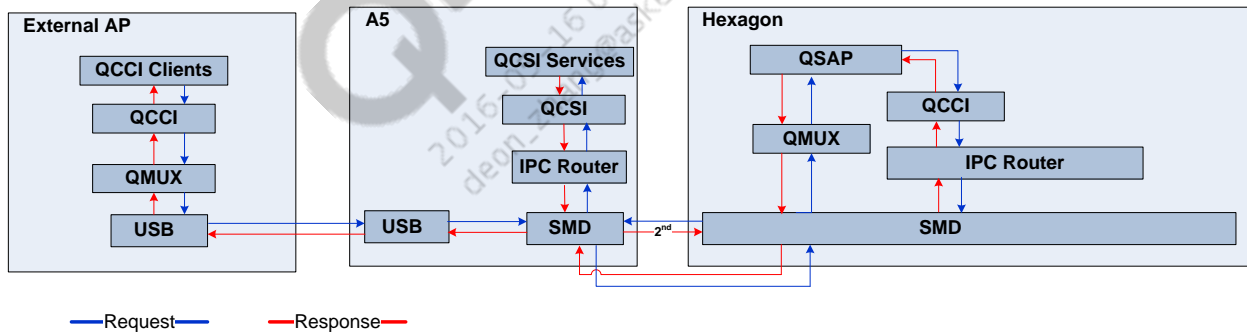

—Request—   —Response—

Figure 4-3 QMI communication between external AP and MDM9x15/MDM9x25 – Service in A5

5 Limitation and Workaround

Qualcomm does not support having QMI service running in an external AP and having a client running in either A5 or Hexagon, e.g., when a file system is in the external AP and you want to get information from the external AP to MDM9x15/MDM9x25 (Figure 5-1).
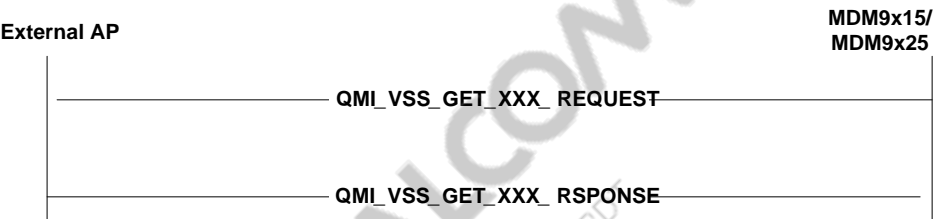

Figure 5-1 Limitation

The following procedure is available to overcome this limitation in MDM9x15/MDM9x25:

1. Register the event mask.

2. When the modem needs information from the AP side, it sends an indication to the external AP.

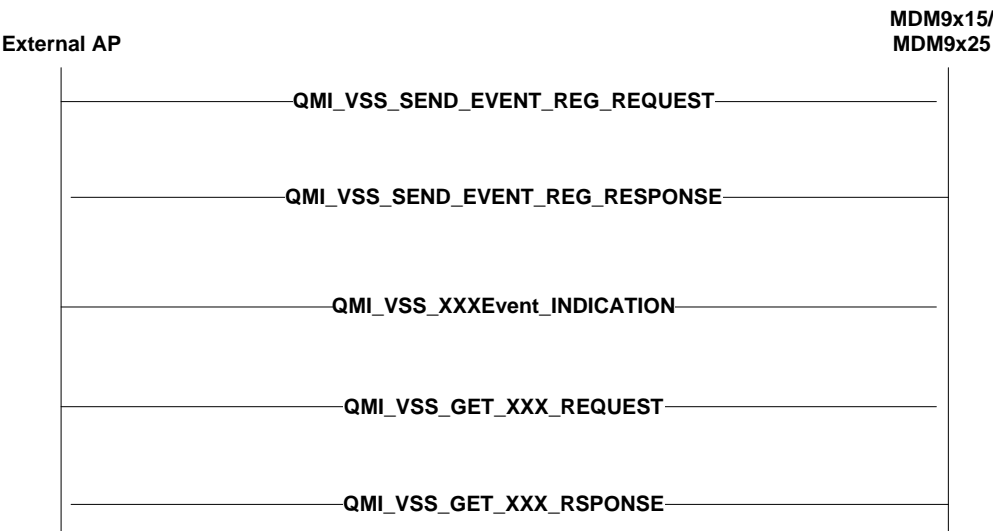3. The AP then sends the information to the modem.

Figure 5-2 Workaround

6 Service ID Assignment

Every vendor-specific service must have a service ID. Service ID assignment should occur inside the enum qmux_service_e_type (in ds_qmi_svc_ext.h). Vendor-specific service IDs must be placed between QMUX_SERVICE_VENDOR_MIN and QMUX_SERVICE_VENDOR_MAX in the enum (see the bold type in the sample code below).

Vendor-specific services must be assigned service IDs consecutively and must begin with QMUX_SERVICE_VENDOR_MIN. QMUX_SERVICE_VENDOR_MAX should always be QMUX_SERVICE_VENDOR_MIN + the number of vendor-specific services. For example, if there are no vendor-specific services, then QMUX_SERVICE_VENDOR_MAX has a value of QMUX_SERVICE_VENDOR_MIN.

Sample code in ds_qmi_svc_ext.h

```
typedef enum
{
QMUX_SERVICE_MIN = 0,
/* Qualcomm Internal Service IDs */

QMUX_SERVICE_VENDOR_MIN = 224,
VENDOR_SPECIFIC_SERVICE1 = QMUX_SERVICE_VENDOR_MIN,
VENDOR_SPECIFIC_SERVICE2 = QMUX_SERVICE_VENDOR_MIN +1,
QMUX_SERVICE_VENDOR_MAX //(= QMUX_SERVICE_VENDOR_MIN +2)
} qmux_service_e_type;
```

7 QSAP Registration

To enable QSAP to handle the VSS QMI message, QSAP registration is needed. The following is an example in the ping application. During service initialization, the newly created VSS service must register itself with QSAP.

```
#include "qmi_sap.h"
void *qmi_ping_register_service(qmi_csi_os_params *os_params)
{
  qmi_idl_service_object_type ping_service_object =
test_get_service_object_v01();
  qmi_csi_error rc = QMI_CSI_INTERNAL_ERR;
  qmi_sap_client_handle qsap_handle;
  if(!ping_service_object || !os_params)
  {
    return NULL;
  }

  rc = qmi_csi_register(ping_service_object, ping_connect_cb,
      ping_disconnect_cb, ping_handle_req_cb, &service_cookie, os_params,
      &service_cookie.service_handle);
  rc |= qmi_sap_register(ping_service_object,NULL,&qsap_handle);
  if(rc != QMI_CSI_NO_ERR)
    return NULL;
  return service_cookie.service_handle;
```

Note that os_params should not be NULL when qmi_sap_register is called if VSS is running on A5.