



TR69 Vendor Specific RPC APP NOTE For BCM93xxx Linux

Broadcom Corporation
5300 California Avenue
Irvine, California, USA 92677
Phone: 949-926-5000
Fax: 949-926-5203

Revision History

Revision	Date	Change Description
0.1	2/14/2014	Preliminary Version

Table of Contents

Introduction	1
Vendor Specific RPC Example	1
Interfacing with TR69 Protocol Stack.....	2
SOAP BODY FORMAT	3
APIs.....	5
Building Image with Vendor Specific RPC	6

Introduction

Starting in 4.16L02 release, TR69 client supports Vendor Specific RPC. Vendors can now add their vendor specific RPC to Broadcom Corporation's TR69 client. This document describes procedures to use this feature.

Vendor Specific RPC Example

The following is an example of a Broadcom RPC, X_BROADCOM_COM_GetDataModelInfo sent from OpenAcS.

```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:cwmp="urn:dslforum-org:cwmp-1-0"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<SOAP-ENV:Header>
  <cwmp:ID
SOAPENV:mustUnderstand="1">ID:intrnl.unset.id.X_BROADCOM_COM_GetDataModelInfo139
2351616425.17697028</cwmp:ID>
  <cwmp:NoMoreRequests>0</cwmp:NoMoreRequests>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<cwmp:X_BROADCOM_COM_GetDataModelInfo xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
<CommandKey>YenCommand</CommandKey>
</cwmp:X_BROADCOM_COM_GetDataModelInfo>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

TR69 client respond to this RPC as followed:

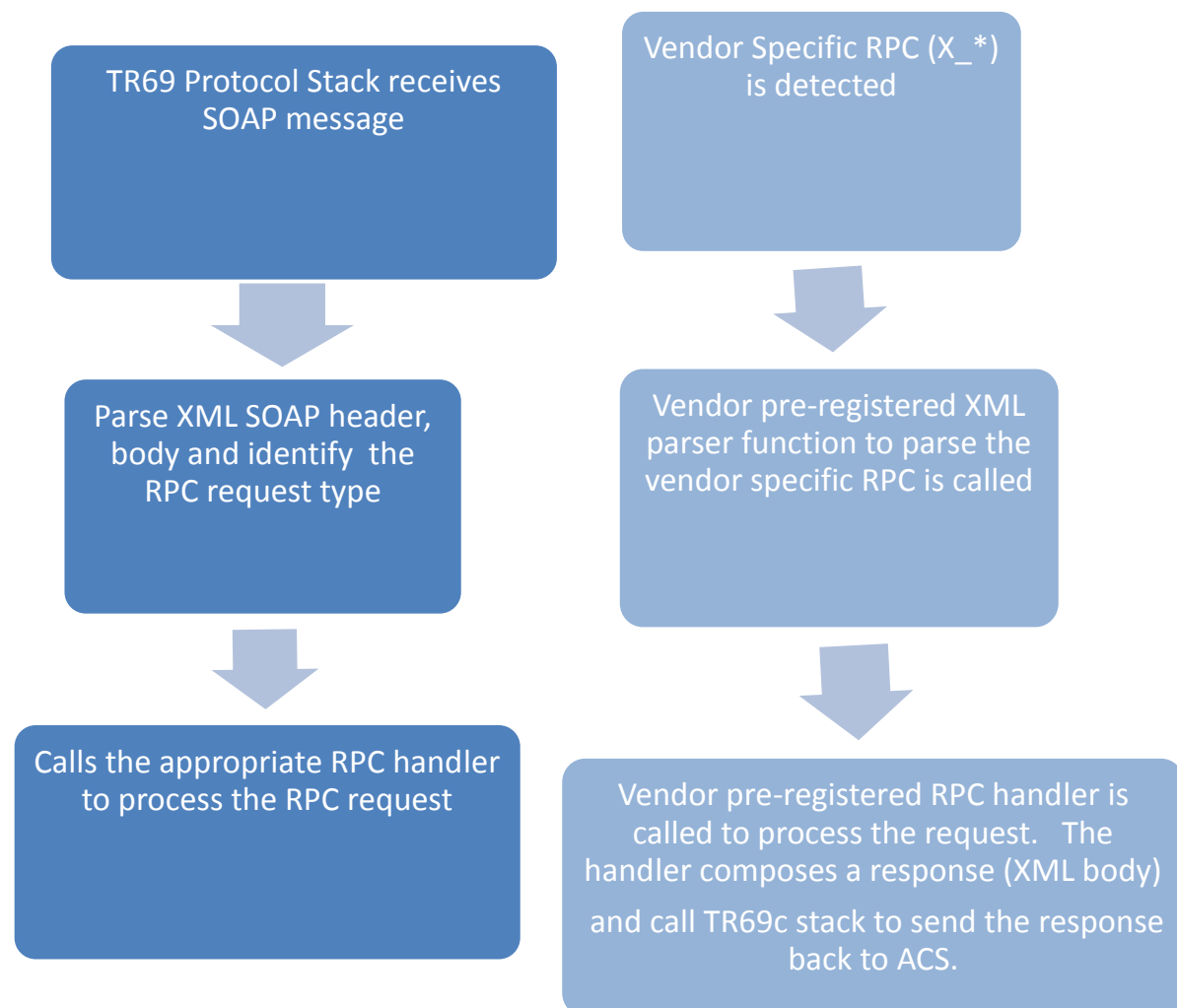
```
<SOAP-ENV:Envelope
  xmlns:SOAP-ENV="http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:SOAP-ENC="http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:cwmp="urn:dslforum-org:cwmp-1-0">
<SOAP-ENV:Header>
```

TR69 Vendor Specific RPC APP NOTE

```
<cwmp:ID SOAP-
ENV:mustUnderstand="1">ID:intrnl.unset.id.X_BROADCOM_COM_GetDataModelInfo13923516
16425.17697028</cwmp:ID>
</SOAP-ENV:Header>
<SOAP-ENV:Body>
  <cwmp:X_BROADCOM_COM_GetDataModelInfo>
    <DataModelSupported>TR98</DataModelSupported>
    <DataModelUsed>TR98</DataModelUsed>
  </cwmp:X_BROADCOM_COM_GetDataModelInfo>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

Interfacing with TR69 Protocol Stack

This chart shows how TR69 client receives and process a SOAP message. On the right is what happens in the case of when Vendor Specific RPC is received.



TR69 Vendor Specific RPC APP NOTE

When TR69 protocol stack receives a soap envelop with X_* in the body, it calls the vendor XML body parser which would parse the body and store all the RPC's parameter(s) in a structure called vendorRPCAction. TR69 protocol stack then calls the vendor's RPC action handler which would fulfill the RPC request. Vendor RPC handler also needs to compose a XML body response and request the protocol stack to send it. TR69 protocol stack encapsulates the response body in the appropriate header and sends the complete response to the ACS.

The next sections list the API available for vendor specific code to call to perform the above sequence of events.

SOAP BODY FORMAT

The header files /SOAPPARSER/RPCState_public.h, /tr69c/inc/tr69cdefs.h and /private/inc/nanoxml.h are the header files that contain data structure pertaining to XML and RPC functions. These are the structures developer of vendor specific RPC should be aware of. Broadcom reference software has a simple SOAP message parser, the body and header are parsed according to the following data structure.

/private/include/nanoxml.h contains definition of XML format that parser expects.

A token type in XML:

```
typedef enum {
    TOKEN_INVALID,
    TAGBEGIN,
    TAGEND,
    TAGDATA,
    ATTRIBUTE,
    ATTRIBUTEVALUE
}TOKEN_TYPE;
<GetParameterValues>
* <ParameterNames arrayType="xsd:string[x]">
*   <xsd:string>name1</xsd:string>
*   .....
* </ParameterNames>
* </GetParameterValues>
```

A Node Descriptor:

```
typedef struct XmlNodeDesc {
    NameSpace *nameSpace;
    char *tagName;
    XML_SET_FUNC setXmlFunc;
```

TR69 Vendor Specific RPC APP NOTE

```
void    *leafNode;
} XmlNodeDesc;
iSOAP "SOAP-ENV:" "http://schemas.xmlsoap.org/soap/envelope/"
iCWMP "cwmpp:"      "cwmpp_url_str_v1_2"
iXSI  "xsi:"        "http://www.w3.org/2001/XMLSchema-instance"
iXSD  "xsd:"        http://www.w3.org/2001/XMLSchema
typedef XML_STATUS (*XML_SET_FUNC)(const char *name, TOKEN_TYPE ttype, const char
*value);
```

leafNode points to parameter of the Method; getRPCMethods has no leaf node
{iCWMP, "GetRPCMethods", fGetRPCMethods, NULL},

leafNode of getParamterNames node has a leaf node of GetParmaterNamesDesc to handle the Name path parameter
{iCWMP, "GetParameterNames", fGetParameterNames, getParameterNamesDesc},

Vendor needs to define a table, mainly XmlNodeDesc array which defines the body of the Vendor RPC. This table has the name of each parameter, a function to parse each parameter, and a leaf node of more parameters if any. For example, for a GetParameterNames RPC, this is the SOAP body and its table of XmlNodeDesc to parse the parameters of this RPC.

```
<SOAP-ENV:Body>
    <cwmpp:GetParameterNames>
        <ParameterPath>InternetGatewayDevice.</ParameterPath>
        <NextLevel>0</NextLevel>
    </cwmpp:GetParameterNames>
</SOAP-ENV:Body>
XMLFUNC(fParameterPath);
XMLFUNC( fNextLevel);
static XmlNodeDesc getParameterNamesDesc[] = {
    {NULL, "ParameterPath", fParameterPath, NULL}, /* data value */
    {NULL, "NextLevel", fNextLevel, NULL},          /* data value */
    {NULL, NULL, NULL, NULL}
};
```

As the body is being parsed, the parameter values need to be stored in a structure for RPC handler function to access. A new structure has been defined for this purpose. Vendor needs to add data field pertaining to the vendor's RPC to this data structure. VendorRPCAction structure is allocated when TR69 protocol stack detects a X_* RPC in the incoming SOAP envelop. It is freed by the protocol stack when it finishes processing the RPC.

TR69 Vendor Specific RPC APP NOTE

```
typedef struct VendorRPCAction {
    char *rpcName; /* the actually name of the method, BROADCOM_COM_restoreToDefault*/
    int  arrayItemCnt; /* cnt of items in parameter list--not used now*/
    union {
        VendorRestoreToDefault Req vendorRestoreToDefaultReq;
        /* more items here later for each vendor RPC method*/
    } ud;
} VendorRPCAction;
```

APIs

Vendors need to implement code to parse the SOAP message of the vendor specific RPC. Since this RPC is vendor specific, vendors also need to implement the RPC handler for this RPC, and also compose a RPC response as well. The following functions (binary release) can be called to make the TR69 protocol stack aware of such vendor RPC. Files xmlVendorRpcTables.c and vendorRpc.c contain implementation of X_BROADCOM_COM_GetDataModelInfo RPC, and can be used as a reference.

```
/** Register Vendor RPC to TR69 protocol stack.
 *
 * @param char* rpcNameStr (IN) name of RPC (i.e. X_BROADCOM_COM_GetDataModelInfo)
 * @param XmlNodeDesc* rpcNodeDesc (IN) The node descriptor containing function to parse
 *                                     the XML format of the RPC
 * @param DO_VENDOR_RPC_FUNC doRpcAction (IN) Function pointer of RPC handler
 *
 */
registerVendorRPC(char *rpcNameStr, XmlNodeDesc *rpcNodeDesc,
                 DO_VENDOR_RPC_FUNC doRpcAction)
```

```
/** Function for vendor code to call after it finishes composing a response's SOAP Body to be
 *     sent to the ACS. Indentation should be done to make the body string more readable.
 *     All the '<' and '"' characters will be converted by this routine before sending to ACS.
 * @param char* response (IN) a string of SOAP body response
 *
 */
sendVendorResponseToAcs(char *response)
```

```
/** Function for vendor code to call to add an inform event to the next inform message
 *     sent to the ACS.
 * @param UINT8 even (IN) event is defined in tr69c/inc/tr69cdefs.h
```


TR69 Vendor Specific RPC APP NOTE

*

*/

UINT32 addInformEventToList(UINT8 event);

Building Image with Vendor Specific RPC

By default, vendor specific RPC is not enabled in the build profile. To build this feature, the following step need to be followed:

- 1.Enable Vendor Specific under “Management Protocols and User Interface Selection”

```
Management Protocols and User Interface Selection
the menu. <Enter> selects submenus --->. Highlighted letters are hotkey
ularizes features. Press <Esc><Esc> to exit, <?> for Help. Legend: [*]
e capable

[*] SERIAL Port Driver
[*] consoled
[ ] Support Menu Driven Interface
[*] Support Command Line Interface
[*] RDPA CLI Support
[*] Telnet
[*] SSHD (Secure Shell Server)
[ ] SSHD Key Generation on target
(16) Maximum Sessions for Telnet and SSH(BSD Pseudo Terminals)
(basic) httpd
[ ] Display HTML page in pop-up window
[*] Quick Setup
WebUI language selection --->
(TR69_SSL) TR69 Management protocol
[*] BCM_PROPRIETARY
[*] RPC_UPLOAD
[*] RPC_QUEUED_TRANSFERS
[*] TR143
[*] VENDOR SPECIFIC RPC
< > SNMP Management protocol
[ ] TR64 Application
<dynamic> UPNP
[ ] Hashed Passwords
```

- 2.Save the build profile (i.e. 963138GW)

- 3.make PROFILE=963138GW clean

TR69 Vendor Specific RPC APP NOTE

4.make PROFILE=963138GW