

Testing Broadband Forum TR-181 Data Models for TR-069 and USP

Tucker Owens
QA Cafe Software
Development and
Standards Expert



What we'll cover today

- Overview of the three standards: TR-069, USP, TR-181
- How to effectively test your entire data model
- How to test deployment or customer specific data models
- How CDRouter can help you understand your device



**Before we start talking
about data models**

Let's talk TR-069 and USP

- Remote management protocols
- Allow for remote configuration/monitoring/diagnostics of deployed devices

TR-069

- CWMP (CPE WAN Management Protocol)
- Managed device sends HTTP POSTs to ACS (Auto-Configuration Server)
- ACS replies with RPCs
 - Get/Set Parameters
 - Add/Delete Objects
- Managed device then sends another POST with the outcome of the RPC

TR-069 Downsides

- It's not easy to get a managed device's attention
 - Connection Requests over HTTP are sometimes impossible
 - XMPP - out of band protocol
 - STUN - "Classic STUN" deprecated in favor of XMPP
- Only supports a single ACS
- Wordy inefficient communication

USP

- Multiple controllers with varying permissions
- All communication is done over the same transport protocol
 - Always on connection
 - No more connection requests
- 3 transport protocols are supported
 - WebSockets
 - STOMP
 - MQTT
- 2 possible layers of encryption
- Efficient messages encoded using Google's protobuf

**So... funny weather we've
been having?**

There's not much to say to these
devices without a data model

What is a data model?

- An agreed upon collection of
 - Objects
 - Parameters
 - Events and Commands (USP only)
- `Device.DeviceInfo.FirmwareImage.{i}`.
 - Name
 - Version
 - Status

TR-181

- First published 2010 by the Broadband Forum
- Issue 2 was published in May
 - Issue 1 was published in February
 - We're not going to talk about issue 1
- The data model most modern TR-69 or USP devices are using
 - In the case of USP it's the only option

Why should I implement an existing data model?

- The documentation is already written
- Unified way of interacting with a device
- Abstracts device specifics

How do you decide which parameters to implement?

- Profiles
- Subsets of the whole data model targeting specific functionality

PacketCaptureDiagnostics:1 Profile

This table defines the *PacketCaptureDiagnostics:1* profile for the *Device:2* data model. The minimum REQUIRED version for this profile is *Device:2.13*.

Name	Requirement
Device.PacketCaptureDiagnostics.	P
DiagnosticsState	W
Interface	W
Duration	W
FileTarget	W
Device.PacketCaptureDiagnostics.PacketCaptureResult.{i}.	P
FileLocation	R

***Okay, I've implemented
TR-181***

Or at least the profiles I want

Now what?

Testing

CDRouter can make data model testing easy

- Profile Testing
- Functional Testing

Profile Testing

- Verify that all the expected parameters and objects exist on the device
 - Get | GetParameterValues
 - GetSupportedDM | GetParameterNames
- Have the expected data type
 - SOAP element data type
 - Reported GetSupportedDM types
- Have the expected read/write access
 - Can values be written to writable parameters?
- Writable objects can be created and deleted

CDRouter TR-69 Profile Tests

- 7 test types
- GetParameterNames from the top level of the profile
- GetParameterNames walking each level
- GetParameterNames verifying Write access
- GetParameterValues
- SetParameterValues
- Object Add and Delete
- GetParameterValues on all GPN full paths

CDRouter USP Profile Tests

- 5 test types
- GetSupportedDM verifying presence of parameters
- GetSupportedDM verifying Write access
- Get
- Set
- Add and Delete objects

TR-181 Profile Tests

TR-69:

- 1158 Profile Tests
- <Data Model>_<Profile>_<test>

Device2_VLANTermination_gpn_1	Verify Device:2 VLANTermination Profile using GetParameterNames from top level
Device2_VLANTermination_gpn_walk_2	Verify Device:2 VLANTermination Profile using GetParameterNames walk at each level
Device2_VLANTermination_gpn_req_3	Verify Device:2 VLANTermination Profile parameters with 'Write' requirement have Writable flag
Device2_VLANTermination_gpv_4	Verify Device:2 VLANTermination Profile using GetParameterValues RPC
Device2_VLANTermination_spv_5	Verify Device:2 VLANTermination Profile using SetParameterValues RPC
Device2_VLANTermination_ado_6	Verify Device:2 VLANTermination Profile using AddObject and DeleteObject on all creatable objects
Device2_VLANTermination_gpn_and_gpv_7	Verify Device:2 VLANTermination Profile using GetParameterValues for all GetParameterNames full paths

+87 RDK Profile Tests

USP:

- 787 Profile Tests
- USP_<Data Model>_<Profile>_<test>

USP_Device2_VLANTermination_get_1	Verify USP_Device2 VLANTermination Profile using Get from top level
USP_Device2_VLANTermination_set_2	Verify USP_Device2 VLANTermination Profile using Set message
USP_Device2_VLANTermination_ado_3	Verify USP_Device2 VLANTermination Profile using Add and Delete on all creatable objects
USP_Device2_VLANTermination_gsdm_4	Verify USP_Device2 VLANTermination Profile using GetSupportedDM from top level
USP_Device2_VLANTermination_gsdm_req_5	Verify USP_Device2 VLANTermination Parameters have at least as much access as defined in the profile

+61 RDK Profile Tests

**Sometimes profiles
need to be tweaked**

Customizing Profiles

- Modifying the expected write access
- Modifying the expected parameter syntax
- Making a parameter optional
- Skipping a parameter

Profile Modifications

TR-69

- cwmpModifyParameters

```
testvar cwmpModifyParameters {  
    string R Device.Time.NTPServer1 {size {maxLength 64}}  
}
```

USP

- uspModifyParameters

```
testvar uspModifyParameters {  
    string R Device.Time.NTPServer1 {size {maxLength 64}}  
}
```

Skipping Parameters in Profiles

TR-69

- cwmpSkipParameters

```
testvar cwmpSkipParameters {  
    Device.Time.NTPServer1  
}
```

USP

- uspSkipParameters

```
testvar uspSkipParameters {  
    Device.Time.NTPServer1  
}
```


Custom Profile Tests

- Partial implementations of many profiles
- Vendor defined parameters
- Many customizations to parameter types/syntax
- Run the CDRouter profile tests on a custom data model

Custom Data Model Format

- BBF TR-106 Format
- CDRouter treats the entire data model as a single profile
- `<document>`
 - `<model>`
 - `<object>`
 - `<parameter>`
 - `<syntax>`
 - `<parameter>`
 - `<syntax>`
 - `<object>`
 - ...

Configuring CDRouter to use a Custom Data Model

```
testvar_group cwp_profile_1 {  
    testvar cwpProfileName          KITT:2  
    testvar cwpProfilePath          /usr/cdrouter-data/custom/cwp/KnightIndustries-prototype.xml  
}
```

```
testvar_group usp_profile_1 {  
    testvar uspProfileName          KITT:2  
    testvar uspProfilePath          /usr/cdrouter-data/custom/usp/KnightIndustries-prototype.xml  
}
```

Add Custom Profile Tests to a Package

CWMP



[cwmp_profiles.tcl \(7\)](#) - CWMP profile testing for user defined CWMP profiles



[usp_datamodels.tcl \(5\)](#) - USP data model testing for user defined USP data models

USP

**I don't have an XML data
model for my device**

CDRouter Auto Profile

- Uses RPCs to profile the DUT
- Generates a XML data model based on the reported parameters
- Should be used as a starting point
 - May be missing parameters if they weren't reported by the DUT
 - Does not contain syntax information

CDRouter Auto Profile

testvar uspProfilePath

auto

testvar cwpmpProfilePath

auto

Test Navigation

Jump to

Enter test name

← Prev Failure

Next Failure →



Result History

Export as CSV

Printable Report

Delete this Result...

	testlist.log
	usp-extracted-data-model.xml
	usp-scenario-test.usp.debug
	usp_conformance_8_5-wan-eth6.cap
	usp_conformance_8_5-wan-m.cap

Functional Testing

- TR-69
 - `tp181.tcl` - TP-181 test plan
 - `tr69_wireless.tcl` - wireless functionality
 - `tr69_diagnostics.tcl` - diagnostics functionality
 - `tr69_annex_n.tcl` - bulk data functionality
- USP
 - `usp_conformance.tcl` - TP-469 test plan
 - `usp_annex_a.tcl` - bulk data functionality

CDRouter Scenarios

- Simplified test case language
- Easy to manipulate parameters, create and configure objects
- Bootstrap parameters
 - Allows for configuration as a package starts
- USP/CWMP Scenarios
 - Full test cases written in the scenario language
- Full documentation in the User Guides

Resources

- CDRouter User Guides:
 - support.qacafe.com/cdrouter/user-guide/cdrouter-tr-069-user-guide
 - support.qacafe.com/cdrouter/user-guide/cdrouter-usp-user-guide
- USP spec: <https://usp.technology>
- USP data models: usp-data-models.broadband-forum.org
- CWMP data models: cwmp-data-models.broadband-forum.org
- CDRouter training series:
<https://www.qacafe.com/cdrouter-training>

Thank You