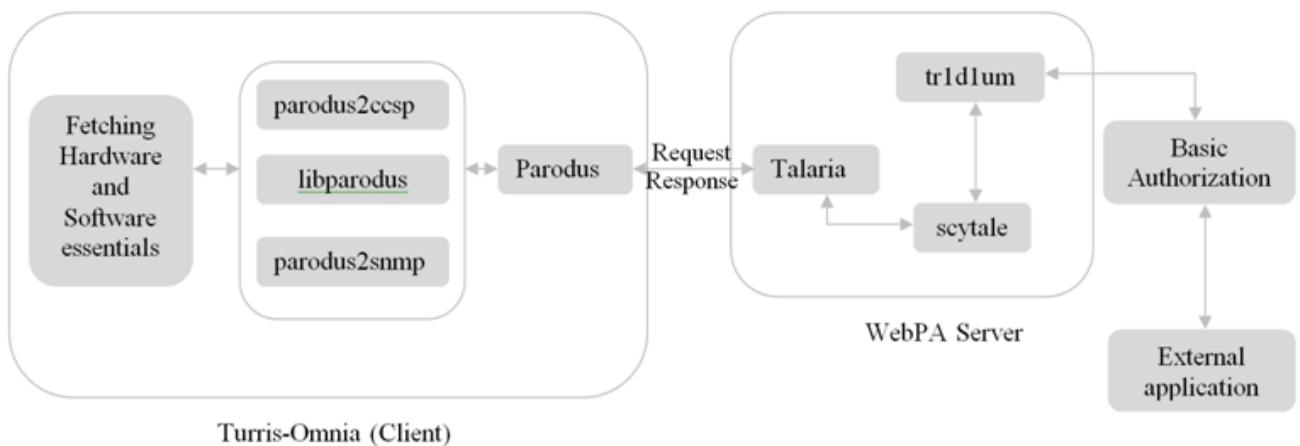


# RDK Broadband - WebPA Support in Turris Omnia - User Manual - 2020 - M3

- [Introduction](#)
- [Environment Setup](#)
  - [Set-up Considerations](#)
  - [User Access Considerations](#)
  - [Accessing the System](#)
  - [System Work Flow](#)
  - [Exiting the System](#)
- [Executing System](#)
- [Limitations](#)
- [Troubleshooting](#)
  - [Error Messages](#)
  - [Special Considerations](#)

## Introduction

- Webpa is a combination of a server and client that provide a simple interface to access the TR-181 objects on an RDK device.
- It is a secure web protocol messaging system for bi-directional communication.



## Environment Setup

### • Set-up Considerations

On the WebPA Server and Client side ensure the following services.

1. **Talaria:** Talaria Service connects with the client and retrieves the data according to the user request.
2. **Scytale:** This service acts as a communication bridge between Talaria and Tr1d1um which is highly helpful in transmitting the data.
3. **Tr1d1um:** This receives the requests from the external resources, encrypts it and send it Scytale.

#### On client-side

4. **Parodus:** Parodus is the light weight client that reaches out to the xmidt cloud to establish the connection from CPE devices.

Additional Services:

5. **Petasos :** Petasos helps reduce the load on the Talaria machines by calculating which specific Talaria a device should connect to & redirecting the incoming request.
6. **Caduceus:** Caduceus provides the pub-sub message delivery (notification) mechanism for xmidt

### • User Access Considerations

1. The mentioned services need to be actively running with the basic key authorization.
2. **Auth token:** The auth token will be used when configuring different webPA components as well while performing GET/SET requests to the CPE from a 3rd party application.

3. Make ServerURL changes in the script file of client service, /lib/rdk/parodus\_start.sh

4. Run the below commands for effective communication

```
rm -rf /tmp/parodusCmd.cmd
```

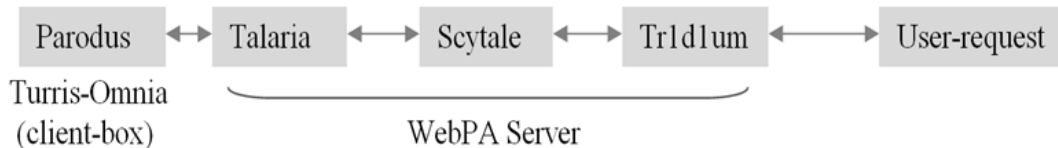
```
systemctl restart parodus
```

## • Accessing the System

Once the Turris-Omnia set-up is active with client services, communication between turris and WebPA server can be established.

## • System Work Flow

The following are the major components/services involved in the fetching the data,



## • Exiting the System

The validation artifacts that are essential to ensure the functionality,

- Authorization Key
- WebPA Server details(IP, Port number)
- CPE Device mac address
- Data Parameters(Specific details that need to be retrieved)

## Executing System

The data from the client device can be fetched through the curl commands,

- Curl command to retrieve the data(get parameter):

```
curl -H 'Authorization:Basic dXNlcjp3ZWJwYQo=' -i 'http://192.168.2.75:9003/api/v2/device/mac:d858d700a5d6/config?names=Device.Users.User.3.X_CISCO_COM_Password'
```

Response Message:

```
{"parameters":[{"name":"Device.Users.User.3.X_CISCO_COM_Password","value":"b1be9cacbfaf0d9d1b633915e8ed0259753057a0a10853a414947d6c27d074c1","dataType":0,"parameterCount":1,"message":"Success"}],"statusCode":200}
```

- Curl command for the set operation:

```
curl -X PATCH http://192.168.2.75:9003/api/v2/device/mac:d858d700a5d6/config -d '{"parameters":[{"dataType":0,"name":"Device.Users.User.1.X_CISCO_COM_Password","value":"Testing123"}]}' -H 'Authorization:Basic dXNlcjp3ZWJwYQo='
```

Response message:

```
{"parameters":[{"name":"Device.Users.User.1.X_CISCO_COM_Password","message":"Success"}],"statusCode":200}
```

Console Output Screenshot:

## Generic Data Parameters: Get Only

- i. Device.DeviceInfo.Manufacturer
- ii. Device.DeviceInfo.ManufacturerOUI
- iii. Device.DeviceInfo.ModelName
- iv. Device.DeviceInfo.SerialNumber
- v. Device.DeviceInfo.HardwareVersion
- vi. Device.DeviceInfo.SoftwareVersion
- vii. Device.DeviceInfo.UpTime
- viii. Device.DeviceInfo.MemoryStatus.Total
- ix. Device.DeviceInfo.MemoryStatus.Free
- x. Device.DeviceInfo.ProcessStatus.CPUUsage
- xi. Device.Hosts.X\_CISCO\_COM\_ConnectedDeviceNumber

## Set/Get Parameters:

- i. Device.WiFi.Radio.10000.Channel
- ii. Device.Users.User.3.X\_CISCO\_COM\_Password

## Limitations

Validated only the above get and set parameters mentioned.

## Troubleshooting

- Error Messages

Following are the error message that user may taken into considerations:

1. "message": "Invalid parameter value"}], "statusCode": 520

For Invalid parameter value, check for correct parameter name and the unwanted space in the command

2. "message": "Error unsupported namespace", "statusCode": 520

For Unsupported namespace, check for the respective services that are essential to fetch the data. For example, WiFi related information can be accessed only if ccspwifiaagent service is active.

3. "message": "Service Unavailable", "statusCode": 531

For this error, ensure the network connection and the server and client-side services are up.

In client device, ex: RPI, check parodus is running actively, also ensure below points

- webpa\_pre\_setup.sh missing in device
- after adding br0 its working

## • Special Considerations

Since different services are involved in the communication, port-number specification should be taken into account

1. In Client-side, along with ServerURL Port number of Talaria should be specified.
2. From user-end, while requesting for information Tr1d1um's Port number should be given.