

Application Test Plan

V1.0.0

Important notices, IPR statement, disclaimer and Copyright

This chapter contains important information about PRPL and this document (hereinafter 'This PRPL Document').

ABOUT PRPL

The prpl Foundation (PRPL) is a not-for-profit organization which publishes documents including, but not limited to, Requirements, Specifications, Recommendations, API application programming interfaces, and Test Plans.

THIS MAY NOT BE THE LATEST VERSION OF THIS PRPL DOCUMENT

This PRPL Document is the output of the Working Groups of the PRPL and its members as of the date of publication. Readers of This PRPL Document should be aware that it can be revised, edited or have its status changed according to the PRPL working procedures.

THERE IS NO WARRANTY PROVIDED WITH THIS PRPL DOCUMENT

The services, the content and the information in This PRPL Document are provided on an "as is" basis. PRPL, to the fullest extent permitted by law, disclaims all warranties, whether express, implied, statutory or otherwise, including but not limited to the implied warranties of merchantability, non-infringement of third-parties rights and fitness for a particular purpose. PRPL, its affiliates and licensors make no representations or warranties about the accuracy, completeness, security or timeliness of the content or information provided in the PRPL Document. No information obtained via the PRPL Document shall create any warranty not expressly stated by PRPL in these terms and conditions.

EXCLUSION OF LIABILITY

Any person holding a copyright in This PRPL Document, or any portion thereof, disclaims to the fullest extent permitted by law (a) any liability (including direct, indirect, special, or consequential damages under any legal theory) arising from or related to the use of or reliance upon This PRPL Document; and (b) any obligation to update or correct this technical report.

THIS PRPL DOCUMENT IS NOT BINDING ON PRPL NOR ITS MEMBER COMPANIES

This PRPL Document, though formally approved by the PRPL member companies, is not binding in any way upon the PRPL members.

Patents essential or potentially essential to the implementation of features described in This PRPL

INTELLECTUAL PROPERTY RIGHTS

Document may have been declared in conformance to the PRPL IP Policy (available at the PRPL website: www.prplFoundation.org).

COPYRIGHT PROVISIONS

This PRPL Document is copyrighted by PRPL, and all rights are reserved. The contents of This PRPL Document are protected by the copyrights of PRPL or the copyrights of third-parties that are used by agreement. Trademarks and copyrights mentioned in This PRPL Document are the property of their respective owners. The content of This PRPL Document may only be reproduced, distributed, modified, framed, cached, adapted or linked to, or made available in any form by any means, or incorporated into or used in any information storage and retrieval system, **with the prior written permission of PRPL or the applicable third-party copyright owner**. Such written permission is **not** however required under the conditions specified in the following two sections.

INCORPORATING PRPL DOCUMENTS IN WHOLE OR PART WITHIN DOCUMENTS RELATED TO COMMERCIAL TENDERS

Any or all section(s) of PRPL Documents may be incorporated into Commercial Tenders (RFP, RFT, RFQ, ITT, etc.) by PRPL and non-PRPL members under the following conditions:

(a) The PRPL Requirements numbers, where applicable, must not be changed from those within the PRPL Documents;

(b) A prominent acknowledgement of the PRPL must be provided within the Commercial document identifying any and all PRPL Documents referenced and giving the web address of the PRPL;

(c) The Commercial Tender must identify which of its section(s) include material taken from PRPL Documents and must identify each PRPL Document used, and the relevant PRPL Section Numbers; and,

(d) The Commercial Tender must refer to the copyright provisions of PRPL Documents and must state that the sections taken from PRPL Documents are subject to copyright by PRPL and/or applicable third parties.

COPYING THIS PRPL DOCUMENT IN ITS ENTIRETY

This PRPL Document may be electronically copied, reproduced, distributed, linked to, or made available by other means, or incorporated into or used in any information storage and retrieval system, but **only in its original, unaltered PDF format**, and with its original PRPL title and file name unaltered. It may not be modified without the advanced written permission of the PRPL.

Executive Summary

This Application test plan outlines the comprehensive validation strategy for an Application Under Test's (AUT) deployment on prplWare through the Life Cycle Management (LCM) framework. The testing approach focuses on verifying that third-party applications install, start, and operate using the standard interfaces and resources prplWare provides, rather than bypassing or modifying the platform beneath them.

By installing the AUT on a gateway running a certified prplWare release and observing its behavior through the TR-181 data model, broker logs, and system resource measurements, the test suite validates that the application integrates cleanly with the platform it runs on. This systematic approach confirms that each application uses only the access, dependencies, and resources it declares, while proactively identifying applications that could compromise gateway stability or security once deployed in the field.

Key objectives of this test plan include:

- Verification of USP-UDS interface use, authentication, and credentials
- Validation of declared data model access and registered data model paths
- Verification of installation and operation on certified prplWare releases
- Assessment of declared Linux kernel module and library dependencies
- Measurement of RAM, flash, and CPU usage against declared values

This test plan is one component of the prpl Application Certification Program. Where the prplWare, High-Level API, and Low-Level API test plans certify the platform, this plan certifies the applications that run on it. Its goal is to give operators confidence that a certified application can be deployed across prplWare-based gateways without platform-specific modification.

This testing initiative will provide crucial validation that third-party applications behave predictably on prplWare, supporting an open and interoperable application ecosystem for the prpl community.

Revision History

Version	Date	Notes
1.0.0	October 8, 2026	● Released for certification. ● Review comments addressed.
0.0.2	October 2, 2026	● Hardware Resources tests added. ● Required information added to Results.
0.0.1	September 13, 2026	● Create Document.

Test Organization

This document organizes tests by group based on related test methodology or goals. Each group begins with a brief set of comments pertaining to all tests within that group. This is followed by a series of description blocks; each block describes a single test. The format of the description block is as follows:

Test Label	The Test Label and Title comprise the first line of the test block. The Test Label is composed of the short test suite name, the group number, and the test number within the group, separated by periods.
Purpose	The Purpose is a short statement describing what the test attempts to achieve. It is usually phrased as a simple assertion of the feature or capability to be tested.
Reference	The References section lists cross-references to the specifications and documentation that might be helpful in understanding and evaluating the test and results.
Functionality Tags	The functionality tag indicates whether the test is mandatory or optional as defined in the prpl Certification Program Guide. If optional, the tag also names the optional feature.
Procedure	The procedure indicates the steps, in order, taken to perform the test. Actions are steps taken by the test tool. Expected Behavior describes what the AUT must do to pass the test.

Table of Contents

Table of Contents

Important notices, IPR statement, disclaimer and Copyright *2*
Executive Summary *4*
Revision History *5*
Test Organization *6*
Table of Contents *7*
Results *8*
Common Test Setup (CTS) *9*
Topology *10*
Optional Features *11*
Group 1: Data Models *12*
Test App.1.1: IPC Mechanism **13**
Test App.1.2: USP Authentication **14**
Test App.1.3: USP Credentials **15**
Test App.1.4: Data Model Dependencies **16**
Test App.1.5: Data Model Trust Paths **17**
Group 2: Dependencies *18*
Test App.2.1: prplWare Version **19**
Test App.2.2: Linux Kernel **20**
Test App.2.3: Libraries **21**
Group 3: Hardware Resources *22*
Test App.3.1: RAM Memory **23**
Test App.3.2: Flash Memory **24**
Test App.3.3: CPU Usage ..... Error! Bookmark not defined.
References *25*

Results

Each test in the Application Test Plan must have a pass and fail metric, per Section 5.2 of the prpl Certification Program Guide.

Additional information that is required and documented as part of the testing:

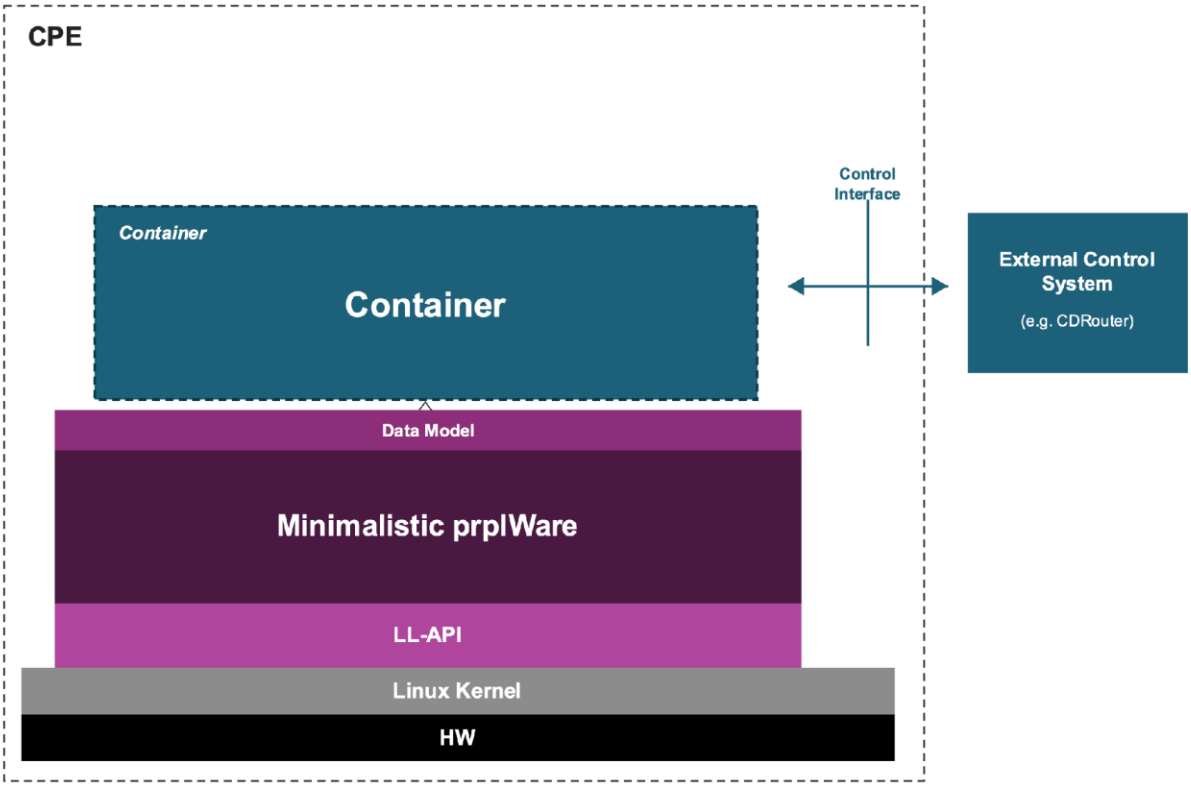
- GW prplWare Version (MUST be a certified version of prplWare)
- The ExecEnv commands to provision the ACL roles used in the InstallDU()
    - For example, to make the Full Access role available, use: SoftwareModules.ExecEnv.1.ModifyAvailableRoles(AvailableRoles = "Full Access", AvailableUserRoles = "")
- The InstallDU() command used to install the application.
    - Network Configuration
    - Storage Configuration
- User Roles for Capabilities.
- Mode (Choose 1)
    - Privileged (root)
    - Privileged (non-root)
    - Unprivileged
- Data Model Paths
- Linux Kernel Modules
- Library Dependencies
- Hardware Resources (Values Required)
    - RAM Memory requirement
    - Flash Memory requirement
- GW Architecture (Choose 1)
    - x86
    - arm64

Common Test Setup (CTS)

The following items are provisioned before running any tests.

1. A Gateway (GW) has an official build of prplWare installed. The following information is recorded as part of the certification and the results.
    a. The architecture of the GW (e.g., x86, arm64)
    b. The version of prplWare.
2. The Application Under Test (AUT) must provide the full InstallDU() command used for testing. When a test mentions InstallDU(), this is the command that will be used. Any additional arguments or changes will be noted in each test. In this document, <AUT> denotes the instance number of the AUT's DeploymentUnit or ExecutionUnit object.
    a. The application must declare which mode is needed: Privileged (root), Privileged (non-root), or Unprivileged.
3. The AUT must provide the

Topology


CPE
Container
Container
Control Interface
External Control System
(e.g. CDRouter)
Data Model
Minimalistic prplWare
LL-API
Linux Kernel
HW

Optional Features

This is the list of optional features and the associated test cases.

- Data Model
    - Applications that modify or read the data model MUST execute the Group 1 tests.

Group 1: Data Models

Scope

The following tests verify that applications properly interact with the data model.

Overview

The tests in this group verify that the AUT uses the USP-UDS connection, authentication, and credentials, and that it accesses and registers only the data model paths it declares.

Test App.1.1: IPC Mechanism

Purpose: Verify that the application utilizes a USP-UDS interface when interacting with the data model.

Reference: BR-CR-A00010

Functionality Tags: Data Model

Test Setup: Common Test Setup is used at the beginning of each part.

Procedure:

Step	Action	Expected Behavior
1.	Use the InstallDU() command to install the AUT on the GW.	The OperateResp message must be returned with Success. The InstallDU() command MUST NOT mount the ubus socket.
2.	Wait 60 seconds.	
3.	Get the Device.SoftwareModules.DeploymentUnit.<AUT>.Status parameter.	The Status must be Installed.
4.	Get the Device.SoftwareModules.ExecutionUnit.<AUT>.Status parameter.	The Status must be Active.

Test App.1.2: USP Authentication

Purpose: Verify that the application utilizes the USP-UDS authenticated Unix socket.

Reference: BR-CR-A00011

Functionality Tags: Data Model

Test Setup: Common Test Setup is used at the beginning of each part.

Procedure:

Step	Action	Expected Behavior
1.	Use the InstallDU() command to install the AUT on the GW or InstallDU(X_PRPLWARE-COM_AutoMountIPC = USP_UDS_Authenticated) if the GW is running prplWare 5.0.0 or later.	The OperateResp message must be returned with Success. The InstallDU() command MUST NOT mount the untrusted UDS socket.
2.	Wait 60 seconds.	
3.	Get the Device.SoftwareModules.DeploymentUnit.<AUT>.Status parameter.	The Status must be Installed.
4.	Get the Device.SoftwareModules.ExecutionUnit.<AUT>.Status parameter.	The Status must be Active.

Test App.1.3: USP Credentials

Purpose: Verify that the application properly utilizes credentials when connecting on the USP-UDS socket.

Reference: BR-CR-A00012

Functionality Tags: Data Model

Test Setup: Common Test Setup is used at the beginning of each part.

Procedure:

Step	Action	Expected Behavior
1.	Use the InstallDU() command to install the AUT on the GW.	The OperateResp message must be returned with Success.
2.	Wait 60 seconds.	
3.	Get the Device.SoftwareModules.DeploymentUnit.<AUT>.Status parameter.	The Status must be Installed. Verify that the OB-USPA logs show no authentication failures during the install.
4.	Get the Device.SoftwareModules.ExecutionUnit.<AUT>.Status parameter.	The Status must be Active.
5.	Get the Device.UnixDomainSockets.Authentication.{i}.EndpointID parameter.	Confirm that this value matches the EndpointID of the application.

Test App.1.4: Data Model Dependencies

Purpose: Verify the data model dependencies, including the data model objects, parameters, commands, and events, used by the application.

Reference: BR-CR-A00020

Functionality Tags: Data Model

Test Setup: Common Test Setup is used at the beginning of each part. The required data model parameters exist in the prplWare version being tested.

Procedure:

Step	Action	Expected Behavior
1.	Configure ACLs with data models specified by the AUT.	ACL configuration is accepted without error.
2.	Use the InstallDU() command to install the AUT on the GW.	The OperateResp message must be returned with Success.
3.	Wait 60 seconds.	
4.	Get the Device.UnixDomainSockets.Authentication.{i}.EndpointID.	Return <AUT-Endpoint-ID>
5.	Verify the Endpoint is in the USP Service Table with Device.USPServices.USPService.[Endpoint==<AUT-Endpoint-ID>	This should return the instance number with the Endpoint-ID.
6.	Get the Device.SoftwareModules.DeploymentUnit.<AUT>.Status parameter.	The Status must be Installed.
7.	Get the Device.SoftwareModules.ExecutionUnit.<AUT>.Status parameter.	The Status must be Active. Verify in the broker logs that the AUT does not try to access any data models that weren't configured in Step 1.

Note: Some AUTs MAY try to query data models that are not required or applicable to prplWare (e.g., RDK extensions). These queries should be documented and explained as such.

Test App.1.5: Data Model Trust Paths

Purpose: Verify that the data model paths registered by the application match the list declared at install and are removed on uninstall.

Reference: BR-CR-A00021

Functionality Tags: Data Model

Test Setup: Common Test Setup is used at the beginning of each part.

Procedure:

Part A: Install

Step	Action	Expected Behavior
1.	Perform a GetSupportedDM on the GW.	The GW sends a GetSupportedDMResp.
2.	Use the InstallDU() command with the list of data model paths to install the AUT on the GW.	The OperateResp message must be returned with Success.
3.	Wait 60 seconds.	
4.	Get the Device.SoftwareModules.DeploymentUnit.<AUT>.Status parameter.	The Status must be Installed.
5.	Get the Device.SoftwareModules.ExecutionUnit.<AUT>.Status parameter.	The Status must be Active.

Part B: Uninstall

Step	Action	Expected Behavior
6.	Use the Uninstall() command for the AUT on the GW.	The OperateResp message must be returned with Success.
7.	Wait 30 seconds.	
8.	Get the Device.SoftwareModules.DeploymentUnit.<AUT> object.	The AUT object must be removed.
9.	Perform a GetSupportedDM on the GW.	The GW sends a GetSupportedDMResp. The response matches Step 1.

Group 2: Dependencies

Scope

The following tests verify application dependencies for proper function.

Overview

The tests in this group verify the dependencies the AUT requires to function.

Test App.2.1: prplWare Version

Purpose: Verify that the application installs and starts properly on the declared prplWare version.

Reference: BR-CR-A00050

Functionality Tags: Mandatory

Test Setup: Common Test Setup is used at the beginning of each part. A certified version of prplWare is installed on the GW.

Procedure:

Step	Action	Expected Behavior
1.	Record the prplWare version running on the GW.	The version must match the prplWare version recorded in the Results.
2.	Use the InstallDU() command to install the AUT on the GW.	The OperateResp message must be returned with Success.
3.	Wait 60 seconds.	
4.	Get the Device.SoftwareModules.DeploymentUnit.<AUT>.Status parameter.	The Status must be Installed.
5.	Get the Device.SoftwareModules.ExecutionUnit.<AUT>.Status parameter.	The Status must be Active.

Test App.2.2: Linux Kernel

Purpose: Verify that the application installs and starts with declared Linux kernel modules not included in standard prplWare.

Reference: BR-CR-A00030

Functionality Tags: Mandatory

Test Setup: Common Test Setup is used at the beginning of each part.

Procedure:

Step	Action	Expected Behavior
1.	Install any declared Linux kernel modules on the GW.	Confirm the kernel modules are loaded.
2.	Use the InstallDU() command to install the AUT on the GW.	The OperateResp message must be returned with Success.
3.	Wait 60 seconds.	
4.	Get the Device.SoftwareModules.DeploymentUnit.<AUT>.Status parameter.	The Status must be Installed.
5.	Get the Device.SoftwareModules.ExecutionUnit.<AUT>.Status parameter.	The Status must be Active.

Test App.2.3: Libraries

Purpose: Verify that the application installs and starts with declared library dependencies not included in standard prplWare.

Reference: BR-CR-A00040

Functionality Tags: Mandatory

Test Setup: Common Test Setup is used at the beginning of each part.

Procedure:

Step	Action	Expected Behavior
1.	Install any declared library dependencies on the GW.	Confirm the libraries are installed.
2.	Use the InstallDU() command to install the AUT on the GW.	The OperateResp message must be returned with Success.
3.	Wait 60 seconds.	
4.	Get the Device.SoftwareModules.DeploymentUnit.<AUT>.Status parameter.	The Status must be Installed.
5.	Get the Device.SoftwareModules.ExecutionUnit.<AUT>.Status parameter.	The Status must be Active.
6.	Get the Device.SoftwareModules.ExecutionUnit.<AUT>.HostObject.{i} object.	Confirm the libraries listed match the declared libraries.

Note: AUTs MAY mount different utilities and libraries depending on the GW target architecture (x86, arm64).

Group 3: Hardware Resources

Scope

The following tests verify application hardware resources.

Overview

The tests in this group verify the hardware resources the AUT requires to install and start. Memory, flash, and CPU usage are measured after installation and compared against the application's stated requirements. These measurements reflect the application at idle, not under load, and should be interpreted accordingly.

Test App.3.1: RAM Memory

Purpose: Verify the RAM memory used by the application after install does not exceed the declared amount.

Reference: BR-CR-A00042

Functionality Tags: Mandatory

Test Setup: Common Test Setup is used at the beginning of each part. The AUT must declare its RAM requirement.

Procedure:

Step	Action	Expected Behavior
1.	Use the InstallDU() command to install the AUT on the GW.	The OperateResp message must be returned with Success.
2.	Wait 60 seconds.	
3.	Get the Device.SoftwareModules.DeploymentUnit.<AUT>.Status parameter.	The Status must be Installed.
4.	Get the Device.SoftwareModules.ExecutionUnit.<AUT>.Status parameter.	The Status must be Active.
5.	Get the Device.SoftwareModules.ExecutionUnit.<AUT>.MemoryInUse parameter.	The memory value of the AUT must not exceed the declared RAM requirement.

Test App.3.2: Flash Memory

Purpose: Verify the flash memory used by the application after install does not exceed the declared amount.

Reference: BR-CR-A00042

Functionality Tags: Mandatory

Test Setup: Common Test Setup is used at the beginning of each part. The AUT must declare its flash storage requirement.

Procedure:

Step	Action	Expected Behavior
1.	Execute df /lcm on the GW.	Record the results of the command.
2.	Use the InstallDU() command to install the AUT on the GW.	The OperateResp message must be returned with Success.
3.	Wait 60 seconds.	
4.	Get the Device.SoftwareModules.DeploymentUnit.<AUT>.Status parameter.	The Status must be Installed.
5.	Get the Device.SoftwareModules.ExecutionUnit.<AUT>.Status parameter.	The Status must be Active.
6.	Execute df /lcm on the GW.	The increase in used space since Step 1 MUST NOT exceed the declared flash storage requirement.

References

- Business Requirements For Certification, Section 5.4
  - https://prplfoundationcloud.atlassian.net/wiki/x/toDe