# Acceptance Tests

**Test Name**:   
**Requirement(s) Tested**:   
**Outline**:   
**Pre-requisites**:   
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
|  |  |  |
|  |  |  |

**Test Name**: PD\_1  
**Requirement(s) Tested**: 1, 25  
**Outline**: Load a Protocol plugin  
**Pre-requisites**: Service is running, a protocol plugin is available for loading, CLI open  
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
| 1 | Enter command to load protocol plugin | Service accepts command without issue |

**Test Name**: PD\_2  
**Requirement(s) Tested**: 2, 25  
**Outline**: Load a communication plugin  
**Pre-requisites**: Service is running, a communication plugin is available for loading, CLI open  
  
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
| 1 | Enter command to load communication plugin | Service accepts command without issue |

**Test Name**: PD\_3  
**Requirement(s) Tested**: 3, 25, 29, 30, 31  
**Outline**: Program reads the maximum threads from the testcase file  
**Pre-requisites**: Service is running, a testcase is ready to read, CLI open  
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
| 1 | Enter command to load a test | Service accepts command without issue |
| 2 | Read log | Log shows that maximum threads has been set |

**Test Name**: PD\_4  
**Requirement(s) Tested**: 3, 4, 25, 29, 30, 31  
**Outline**: Program reads the maximum threads from the testcase, applies hard limit when too high  
**Pre-requisites**: Service is running, a testcase is ready to read, CLI open  
  
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
| 1 | Enter command to load a test | Service accepts command without issue |
| 2 | Read log | Log shows that maximum threads has been set (lower than asked for but *at* the limit) |

**Test Name**: PD\_5  
**Requirement(s) Tested**: 5, 6, 7, 8, 14, 15, 18, 19, 25, 27, 28  
**Outline**: While a test is running, measure performance changes when making other queries against the system  
**Pre-requisites**: Service is running, a testcase is running, CLI is open, log is being read  
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
| 1 | Enter different commands to interact with the system | The system accepts the command |
| 2 | Do this several times in a row | The system happily performs each task |
| 3 | When the command completes, check logs | Logs should show no discernible difference in running time or efficiency |

**Test Name**: PD\_6  
**Requirement(s) Tested**: 9, 25  
**Outline**: When making changes regarding the threading facility, few files need changing, the ones that exist are relatively easy to edit  
**Pre-requisites**: Have source code, change needs to be made  
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
| 1 | Look at area that needs changing | Only a single class needs editing |

**Test Name**: PD\_7  
**Requirement(s) Tested**: 10, 11, 25  
**Outline**: The service needs to be somewhat self-recoverable, and should be handled by sysvinit/system.  
**Pre-requisites**: Device off, software installed on device and setup as a service/daemon  
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
| 1 | Turn system on |  |
| 2 | Check running services | ProtDev should be listed |

**Test Name**: PD\_8  
**Requirement(s) Tested**: 11, 20, 21, 22  
**Outline**: The service should be started exclusively as a service/daemon  
**Pre-requisites**: Service is not running  
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
| 1 | Attempt to run a ProtDev executable directly | Program should error out requiring it to be run as a service |

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
| 1 | Attempt to run ProtDev via the systems init infrastructure | The program should happily start running and the init systems logs should show a running service |

**Test Name**: PD\_9  
**Requirement(s) Tested**: 12, 25, 29, 32, 33, 34, 35, 36  
**Outline**: Initiate a test by passing in a testcase, it sends data to the target  
**Pre-requisites**: Service is running, protocol available, testcase written and correct, CLI open  
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
| 1 | Send test case to program | Program accepts it without issue |
| 2 | Check logs | A log file for the test has been created, showing statistics and options provided along with any unexpected issues from the service |
| 3 | Check target | Target should show interaction with the service |

**Test Name**: PD\_10  
**Requirement(s) Tested**: 13, 14, 25  
**Outline**: Stop a test while it’s running  
**Pre-requisites**: Service is running, a test is running  
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
| 1 | Enter a command to view running tests | A list is shown displaying tests IDs and their description |
| 2 | Enter a command referencing the test ID to stop it | Program accepts the command and begins to shut down the test cleanly |

**Test Name**: PD\_11  
**Requirement(s) Tested**: 15, 25  
**Outline**: As the program is running, query the program for its current statistics  
**Pre-requisites**:   
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
| 1 | Enter a command querying the current statistics for the programs running | The program outputs current stats, including any running threads and other such business |

**Test Name**: PD\_12  
**Requirement(s) Tested**: 16, 26, 45  
**Outline**: The API allows a UX designer access to the system for data or control purposes  
**Pre-requisites**: API is written and public  
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
| 1 | Check documentation for API | API is present and made public for easy access |

**Test Name**: PD\_13  
**Requirement(s) Tested**: 17, 25  
**Outline**: Developer loads a plugin, program performs self-checks to verify the plugin is functioning correctly  
**Pre-requisites**: Service is running, plugin is written and ready to load  
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
| 1 | Enter a command to solo load a plugin | Program accepts command without issue and loads the plugin |
| 2 | Wait | Program outputs that the plugin is working/valid |

**Test Name**: PD\_14  
**Requirement(s) Tested**: 23  
**Outline**: When the program is first launched performs a series of steps that validate its environment and rectify any issues that may be present  
**Pre-requisites**: Service hasn’t been started on this device before/clean installation  
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
| 1 | Run either setup script or start the program | Program announces it’s the first time it’s being setup |
| 2 | Wait | Program will go through a checklist of items and setup its environment |
| 3 | Wait | Service will be running and will have set up its own directories with its required libs, logs and other such locations |

**Test Name**: PD\_15  
**Requirement(s) Tested**: 25, 55  
**Outline**: While the program is installed perform an uninstallation  
**Pre-requisites**: Service is installed and configured on the system  
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
| 1 | Enter command to remove the service | The program begins exiting gracefully |
| 2 | Wait | The uninstallation process will verify what the user wishes to do with user edited directories (plugins, logs, configs) |
| 3 | Enter choices | The program will carry out the choices (remove or keep and tar) |
| 4 | Wait | The program will no longer have a significant presence on the device |

**Test Name**: PD\_16  
**Requirement(s) Tested**: 24, 25  
**Outline**: While the program is running/performing tests, perform a hard yet As-Graceful-As-Possible (AGAP) exit.  
**Pre-requisites**: Program is running  
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
| 1 | Send the Programs kill command on the CLI | The program will begin attempting to kill all its child threads as quickly as possible while attempting to maintain some level of graceful exit |
| 2 | Verify resources returned and program killed | The program should no longer be present, and any resources returned to the system |

**Test Name**: PD\_17  
**Requirement(s) Tested**:   
**Outline**:   
**Pre-requisites**:   
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
|  |  |  |
|  |  |  |

**Test Name**: PD\_18  
**Requirement(s) Tested**:   
**Outline**:   
**Pre-requisites**:   
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
|  |  |  |
|  |  |  |

**Test Name**: PD\_19  
**Requirement(s) Tested**:   
**Outline**:   
**Pre-requisites**:   
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
|  |  |  |
|  |  |  |

**Test Name**: PD\_20  
**Requirement(s) Tested**:   
**Outline**:   
**Pre-requisites**:   
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
|  |  |  |
|  |  |  |

**Test Name**: PD\_21  
**Requirement(s) Tested**:   
**Outline**:   
**Pre-requisites**:   
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
|  |  |  |
|  |  |  |

**Test Name**: PD\_22  
**Requirement(s) Tested**:   
**Outline**:   
**Pre-requisites**:   
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
|  |  |  |
|  |  |  |

**Test Name**: PD\_23  
**Requirement(s) Tested**:   
**Outline**:   
**Pre-requisites**:   
**Method**:

|  |  |  |
| --- | --- | --- |
| **STEP** | Action | Expected Observation |
|  |  |  |
|  |  |  |