Software Release Report- DIComm Client

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Document History** | | | | |
| **Version** | **Date** | **Author** | **Section** | **Changes** |
| 1.0 | 22-09-2015 | Deepthi Shivakumar | All | Release version 1.0.2 |
|  |  |  |  |  |
|  |  |  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Administrative Information | | | |
| Project Name | Horizontal (DIComm Client) | Project Identification | NA |
| Project Leader | Maarten Nielen/Satheesh K. S | Date | 22-September-2015 |
| Quality Leader | Sudhir | Form Filled by | Deepthi Shivakumar |
| Name  Address  Phone  Fax | All Project in CDP | | |

|  |  |
| --- | --- |
| **Authorisation** | |
| Quality Leader Signature | Project Leader Signature |
| Sudhir | Maarten Nielen/Satheesh K. S |

**For Code:**

**Product : DIComm Client**

**Release Version :** **1.0.2**

**Release Date: 27-September-2015**

Integration document: Refer DICommClient\_V1\_Integration\_Android.docx

Customer Version ID (Optional): NA

# **Release Description:**

# DIComm client is a horizontal component which allows mobile applications to communicate with connected appliances by implementing DICOMM protocol.

# This currently supports Wi-Fi propositions.

Version number of this release is 1.0.2

# **Source code Links:**

<http://ingbtcpic2lx253.blr.pin.philips.com:8080/scm/git/hor-dicommclient-release-android>

**Branch: master**

# **Source/Binary Files Description:**

# Not applicable

# **System Requirements / Configuration:**

Android: Min sdk version is “9”.

# **Supported features:**

* Mobile applications can discover any Philips connected appliances.
* DIComm Client allows creating specific appliance from the devices discovered.
* DIComm client provides APIs to perform EWS.
* Apps can communicate to an appliance either locally or remotely via CPP.
* Apps can enable/disable subscription events from appliance either locally or remotely.
* Apps can send requests or subscribe events from multiple appliances.
* By default DIComm client has support for Wifi port, Device port, Firmware port, Pairing port, Wifi UI port and Schedule port.
* DIComm client provides flexible APIS for adding/removing appliance specific ports.
* DIComm client takes care of encryption/decryption of data when connected locally if the appliance supports encryption and if it is requested by app.
* DIComm takes care of switching between local and remote communication if appropriate strategy is set by applications when appliance is created.
* DIComm client provides wrapper classes and methods to communicate with cpp for key provisioning, sign on, data download, app update, push notifications etc.

**Known Issues:**

* DIComm client supports only Wifi propositions.
* Pairing is adopted by Air purifier code base and not tested by any other appliance specific requirements.
* Does not support data upload to cpp.

**Key changes in 1.0.2:**

* Fix for EWS failure on lollipop.

**Notes:**

* DIComm client makes use of external ICPClient library for communicating with CPP developed by another Philips team. When the ICPClient APIs gets changed, DIComm client APIs are subjected to change.
* DIComm client is tested in Air and MultiCooker app context.
* EWS changes are required on app side too on lollipop, please contact us for more details if required.

**FAQ:**

1. **DicommClientLib is not downloading from Maven/Artifactory.**

We see that proxy is required to download any dependencies from jcenter, gradle etc but to download binary from maven we may not need proxy.

Please set “No Proxy” in android studio and delete the project local gradle.properties file if it exists.

Also please comment out all the lines in global gradle.properties in your home directory.

Once it downloads, please enable the proxy if required.