

ST-Ericsson Android Bluetooth FTP

Software Architecture version 0.1



DISCLAIMER

© Copyright ST-Ericsson, 2010. All Rights Reserved.

The contents of this document are subject to change without prior notice. ST-Ericsson makes no representation or warranty of any nature whatsoever (neither expressed nor implied) with respect to the matters addressed in this document, including but not limited to warranties of merchantability or fitness for a particular purpose, interpretability or interoperability or, against infringement of third party intellectual property rights, and in no event shall ST-Ericsson be liable to any party for any direct, indirect, incidental and or consequential damages and or loss whatsoever (including but not limited to monetary losses or loss of data), that might arise from the use of this document or the information in it.

ST-Ericsson and the ST-Ericsson logo are trademarks of the ST-Ericsson group of companies or used under a license from STMicroelectronics NV or Telefonaktiebolaget LM Ericsson.

All other names are the property of their respective owners.

For more information on ST-Ericsson, visit www.stericsson.com

Introduction to Android Bluetooth File Transfer Profile

- Designed by ST-Ericsson based on SIG FTP specs 1.1
- In line with existing OPP/PBAP profile in Android. In terms of architecture and code organization.
- Reuses existing OPP source code and enhances FTP specific functionality.
- Extensively tested and stable.
- Data path fine tuned for maximum throughput over obex in Android.
- Code profiling done using advanced tools such as Coverity Prevent.

Features highlights of ST-Ericsson Android Bluetooth File Transfer Profile

- FTP server role supported.
- Uses Secure Rfcomm API's for pin authentication for ftp service connection.
- FTP server accepts and responds to browse folder request.
- FTP server accepts the browse folder and subfolder requests.
- A file or folder can be pushed to the FTP server.
- Multiple files or folders can also be pushed to the ftp server.
- A file with any extension or folder can be pulled from FTP server.
- New folders can also be create on FTP server depending on permissions.
- Respective error codes generated and sent from ftp server in case of a failed transaction.
- Option to configure root directory on server.

Integration with Android Framework

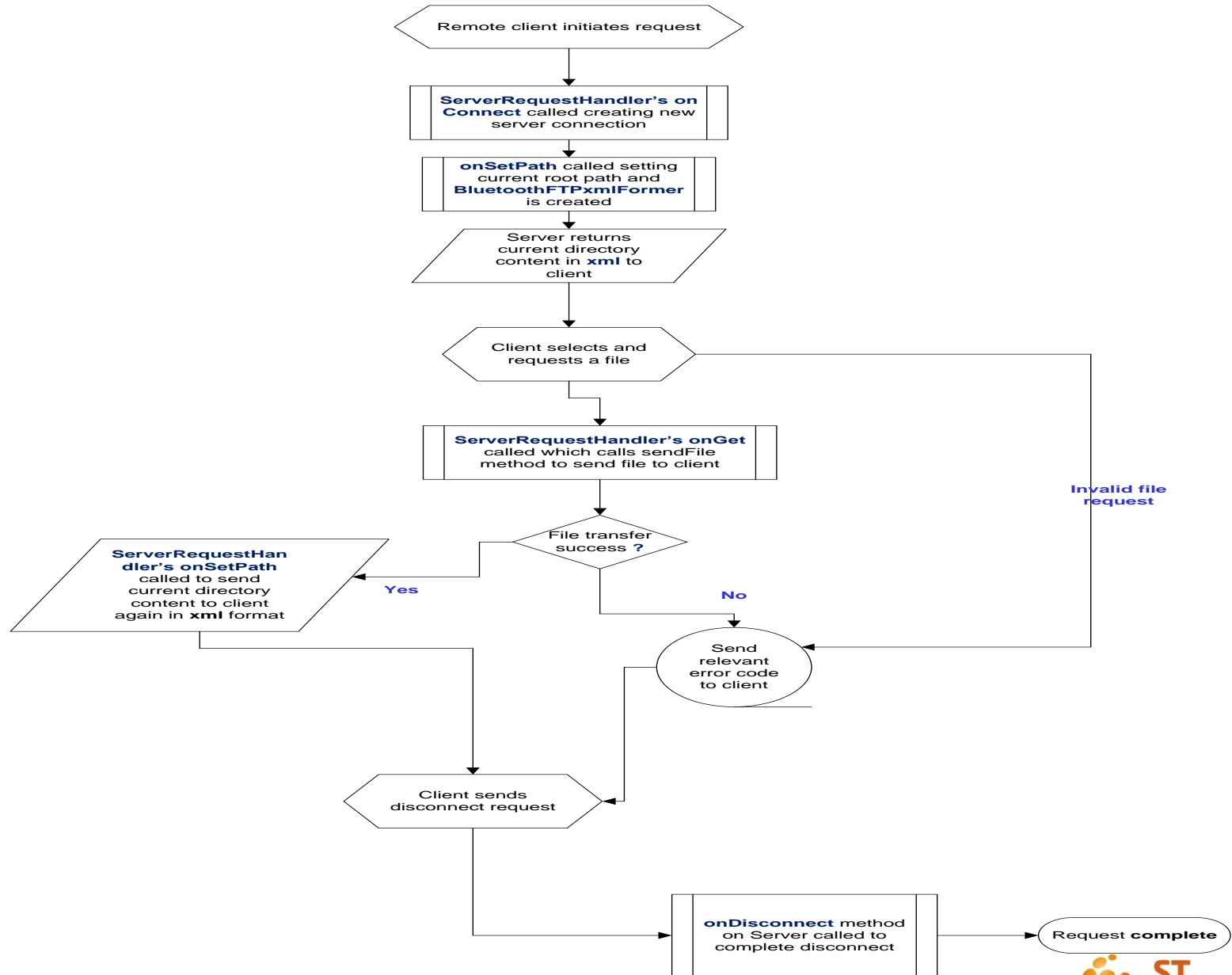
Changes

- Below are the files which have to be modified for integrating FTP with framework.
 - `$<mydroid>/frameworks/base/core/java/android/server/BluetoothService.java`
 - BluetoothService.java - This class is responsible to register Bluetooth services to the sdp tool when device's Bluetooth switches ON. Here new lines are added to register FTP service. <in mHandler definition>
 - `$<mydroid>/frameworks/base/core/java/android/bluetooth/BluetoothUuid.java`
 - BluetoothUuid.java This class defines the UUIDs used to represent Bluetooth profiles.
 - Here we add UUID for FTP and defines it as reserved UUID. Add UUID for FTP.
 - The definition of RESERVED_UUIDS also needs to be changed.
 - AndroidManifest.xml in location `$<mydroid>/packages/apps/Bluetooth`.
 - Create root.config in `/data/data/com.android.bluetooth/files`. If files folder not present create it. This file will have the configurable path for the root directory of ftp server. viz /usr is a valid path.

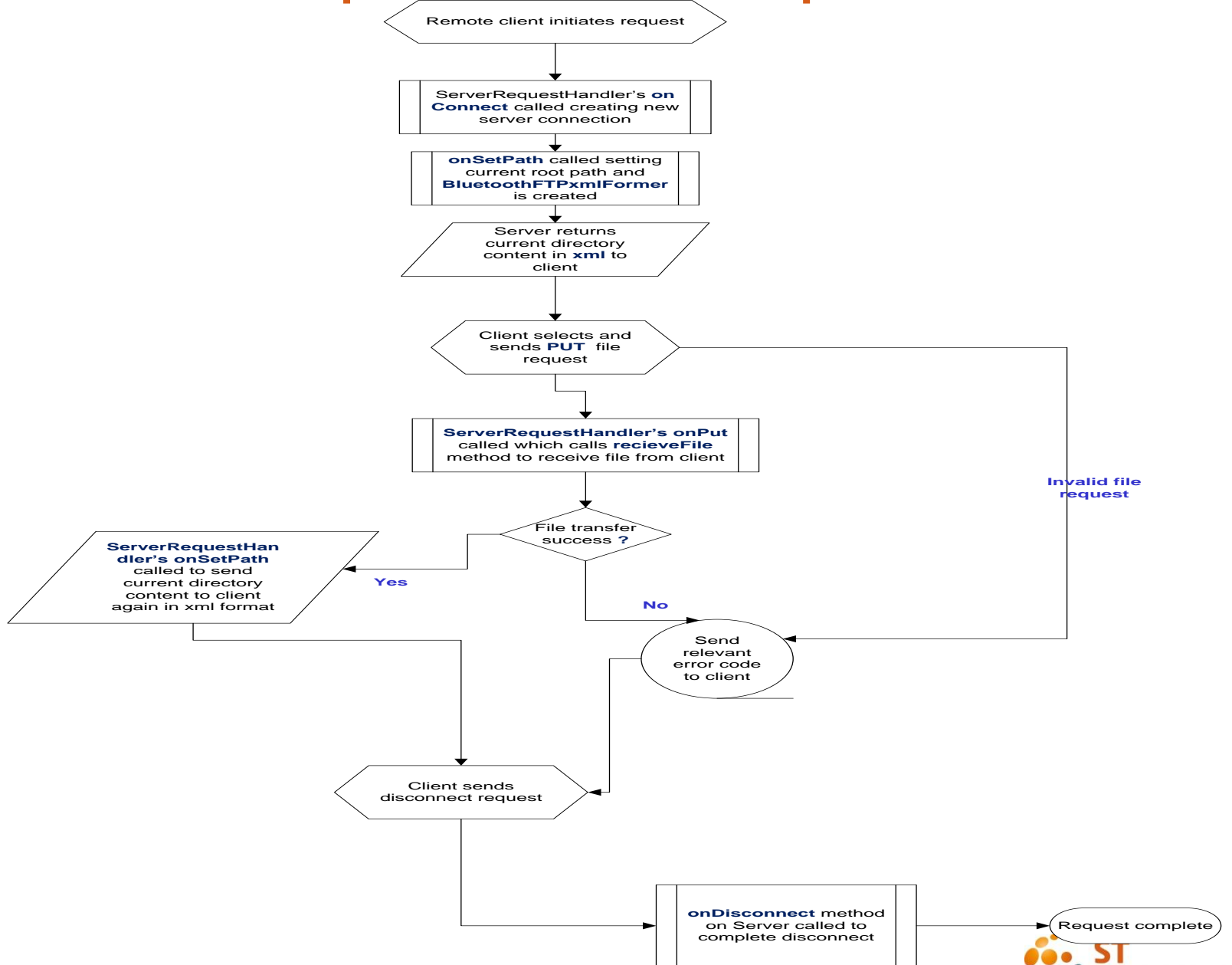
Limitations of ST-Ericsson Android Bluetooth File Transfer Profile

- File of zero size or without extension cannot be pushed to the FTP server
- Implementation does not delete the incomplete folder/file transfers
- Directory path which is mentioned in the root.config (root configuration file in /data/data/com.android.bluetooth/files) may contain system files/folders. Hence it may be deleted by the remote ftp client and cause abnormal behavior of the system.
- Before connecting first time to FTP server, root.config must be added else usr root directory will be exposed to the remote client by default.
- If a folder has a file with zero size or no extension file, its push transmission will be aborted and hence folder push request will be terminated with respective response code. FTP server role supported.

Use case: GET request from remote ftp client



Use case: PUT request from remote ftp client



THANK YOU

